https://theses.gla.ac.uk/

Theses Digitisation:

https://www.gla.ac.uk/myglasgow/research/enlighten/theses/digitisation/

This is a digitised version of the original print thesis.

Copyright and moral rights for this work are retained by the author

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge

This work cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

Enlighten: Theses
https://theses.gla.ac.uk/
research-enlighten@glasgow.ac.uk

APPLICATION OF MICROCOMPUTERS TO ANAESTHESIA AND INTENSIVE CARE

VOLUME I of II VOLUMES

GAVIN NC KENNY, BSc, MBChB, FFARCS
University Department of Anaesthesia
Glasgow Royal Infirmary

Thesis submitted to the University of Glasgow
for the degree of Doctor of Medicine

July, 1982

ProQuest Number: 10647038

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

ProQuest 10647038

Published by ProQuest LLC (2017). Copyright of the Dissertation is held by the Author.

All rights reserved.
This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

Thesis
6561
Copy 2

GLASGOW
UNIVERSITY
LIBRARY

CONTENTS - VOLUME I

CONTENTS - VOLUME II

LIST OF TABLES AND FIGURES

TABLES

FIGURES

ACKNOWLEDGMENTS

I would like to acknowledge the support and advice of all those who have been associated with this project. They include the anaesthetists and medical students who have participated in the CAL and CASA studies and who provided considerable help in the form of suggestions to improve the teaching programs. Dr Carol Schmulian was a co-worker in the CAL and CASA studies and I am grateful to her for the co-operation and enthusiasm she showed. The Department of Health and Social Security provided a grant to fund Dr Schmulian and equipment used for the CASA study.

The nursing and medical staff of the Respiratory Intensive Care Unit have also shown considerable enthusiasm and tolerance during the time the intensive care programs were under development. The support of Mrs Shirley Watt has been especially of value. The Miss MJM Smith Trust provided the Apple II computer used for development of programs in intensive care.

Other members of the University and NHS Departments of Anaesthesia have provided encouragement and advice. Dr W Fitch and Dr JC Howie have kindly assisted with proof reading. In addition, they have both been enthusiastic users of the computer system in intensive care and operating theatre and have made valuable suggestions for improvements in the programs.

Dr Sheila Madsen has been involved in development of the FILE suite of programs during the six months she has spent in our department to study microcomputer applications in medicine, and has provided useful suggestions for these programs.

Mr Paul Davis has been a stimulating associate in development of computer programs for use in the intensive care unit. Mr K Carter and the technicians of the University Department of Anaesthesia, Glasgow Royal Infirmary, have constructed the interfaces used for teaching and connection to equipment in the intensive care unit.

A special acknowledgement must be made of the assistance provided by Strathand Ltd. The directors of this company have provided time and expertise in many technical aspects and have been of considerable help throughout the project.

A project can only succeed if those in authority have the belief that potentially valuable developments could result from an idea and provide the encouragement, both moral and financial, to enable the idea to be tested. Professor Donald Campbell has demonstrated his belief in various applications of microcomputers to anaesthesia and intensive care by continuous support. His own enthusiasm and interest in the project emphasise that application of new microtechnology is not the province of the young, but of those who have the vision to see beyond the immediate horizon.

DECLARATION

Information and results presented in this thesis have been used in the following publications :-

Campbell D, Kenny GNC, Schmulian C, Davis PD. Computer assisted self-assessment in anaesthesia: a preliminary study. Anaesthesia 1980; 35: 998-1002.

Davis PD, Kenny GNC. Microcomputer control of a video cassette lecture. Med Ed 1980; 14: 196-198.

Davis PD, Kenny GNC, Campbell D. On-line analysis of respiratory function with a microcomputer. In: Paul JP, ed. Computing in medicine, London: Macmillan, (in press).

Howie JC, Kenny GNC. A microcomputer based technique for the measurement of respiratory depression in postoperative patients. Br J Anaesth 1982; 1-2P.

Kenny GNC. Programmable calculator: a program for use in the intensive care unit. Br J Anaesth 1979; 51: 793-796.

Kenny GNC. Computer assisted learning in medical education. Proc 10th and 11th Annual Meetings of the National Association of Clinical Tutors. London: Update Publications, 1981; 39-41.

Kenny GNC. Which microcomputer? Br J Hosp Med 1982; 27: 74-78.

Kenny GNC, Campbell D. Intrapulmonary shunt. In: Spence AA, ed. Clinics in Critical Care Medicine: Respiratory Monitoring in Intensive Care. Edinburgh: Churchill Livingston, (in press).

Kenny GNC, Campbell D. Microcomputer assisted cardiovascular monitoring. In: Paul JP, ed. Computing in Medicine. London: Macmillan, (in press).

Kenny GNC, Davis PD. The use of a micro-computer in anaesthetic teaching. Anaesthesia 1979; 34: 583-585.

Kenny GNC, Davis PD. Computing techniques in critical care medicine. In: Ledingham IMcA, Hanning CD, eds. Recent advances in critical care medicine. Edinburgh: Churchill Livingstone, (in press).

Kenny GNC, Davis PD, Campbell D. Monitoring respiratory function with a microcomputer. Br J Anaesth 1979; 51: 996P.

Kenny GNC, Schmulian C. Computer assisted learning in the teaching of anaesthesia. Anaesthesia 1979; 34: 159-162.

Kenny GNC, Schmulian C. Micro computers in self-assessment. In: Payne JP, ed. Proc. Computers in critical care and pulmonary medicine. (in press).

Madsen S, Kenny GNC. Data collection and analysis with a microcomputer. In: Payne JP, ed. Proc. Computers in critical care and pulmonary medicine. (in press).

Schmulian C, Kenny GNC. Use of microcomputers in medical education. In: Paul JP, ed. Computing in Medicine. London: Macmillan, (in press).

Schmulian C, Kenny GNC, Campbell D. Use of microcomputers for self-assessment and continuing education in anaesthesia. Br Med J 1982; 284: 403-405.

SUMMARY

The application of microcomputers to anaesthesia and intensive care has been examined. Three different areas of experience are presented to illustrate the potential use of such systems.

1. COMPUTER-ASSISTED LEARNING/
COMPUTER-ASSISTED SELF-ASSESSMENT

The use of computers for teaching has been evaluated and found to be successful compared with the more traditional form of teaching by tutorial. The reason would appear to be that the provision of immediate feedback as the students progress through the CAL session, re-inforces the learning process.

Computer-assisted self-assessment was also evaluated as a possible aspect of medical audit. The method proved to be well accepted by those who used it and appears to provide an estimate of theoretical knowledge.

The programs used in these evaluations were simple in structure but have been sent by request to several hospitals in the UK and other parts of the world for teaching and self-assessment in anaesthesia.

2. MICROCOMPUTERS IN INTENSIVE CARE
AND OPERATING THEATRE

On-line cardiorespiratory data collection has been undertaken from seriously ill patients in the intensive care unit and operating theatres. This has provided the nursing staff with more time for direct patient care and the medical staff with the derived information which is required to deal with complex clinical problems.

Data analysis is described and the different forms of data display are illustrated by the presentation of three case

reports.

3. MICROCOMPUTERS IN EXPERIMENTATION

The final example of the use of microcomputers describes the control and automatic data collection system designed to measure respiratory response to increasing concentrations of carbon dioxide. The experimental procedure requires such precise timing that it could not be undertaken using manual methods.

All the examples presented use the same inexpensive microcomputer with only alterations in the software and interfaces required to completely change its function. Other examples of the applications of microcomputers are briefly described.

The implication of the use of this new form of technology is that medical and nursing staff must become more aware of the potentials and limitations of microcomputers to direct the use of these powerful tools for the maximum benefit of their patients.

CHAPTER 1

INTRODUCTION

INTRODUCTION

DEVELOPMENT OF COMPUTERS

Blaise Pascal, whose name is associated with many aspects of science and mathematics, devised and built the first mechanical adding machine in 1642. This was called the Pascaline and had the ability to add and subtract at a much faster rate than could be achieved by clerks but could not undertake successfully the processes of multiplication and division. The multiplier wheel was developed by Gottfried Leibniz in 1692 and this invention provided the basis of electromechanical calculators which were only superceded by completely electronic calculators in the 1960's.

Charles Babbage is credited with inventing the concept of the programmable computer. He was awarded the first gold medal of the Royal Astronomical Society in 1822 for his paper 'Observations on the Application of Machinery to the Computation of Mathematical Tables'. The system was called the Difference Engine and was to be a complicated series of cogs and wheels but the lack of engineering precision prevented completion of the final machine.

The Difference Engine was designed to undertake the single task of solving polynomial equations but Babbage decided to embark on an even more ambitious project - the Analytical Engine, which could be employed for a wide variety of tasks. Each task was to be controlled by a unique pattern of internal actions, which was essentially the formula for a programmable computer. The work was never completed but the concept was sound and formed the basis for the design of present computers.

In 1936, Konrad Zuse in Germany and Howard Aiken in the United States of America almost simultaneously developed the idea of using binary units rather than decimal for calculation. Both used electromechanical relays as the switching devices and functioned usefully during the Second World War.

The war provided a major stimulus to computer development and a British team produced the Colossus series of machines which were the world's first electronic digital computers. They were based on high speed electronic valves rather than electromechanical relays and therefore performed calculations much more rapidly. The information to be processed was entered at a speed of 5 000 characters per second using photoelectric paper tape readers but the task undertaken by the system was fixed. These machines were only built to decipher codes and could not be used for other functions without considerable difficulty.

The Electronic Numerical Integrator And Calculator (ENIAC) was built in the USA to produce ballistic tables for guns and missiles. It employed a decimal rather than a binary mode of calculation and also used electronic valves as the switching devices. The program which the computer ran could be changed but alterations were complex and involved rewiring parts of the machine.

Von Neumann joined the team who had built ENIAC and suggested replacing the decimal system with binary. He also introduced the concept of the stored program which allowed the computer to load and change programs when required. Programs could therefore be interlinked and call each other. They could also be modified by the computer as they were executed.

The introduction of the transistor in 1956 was a major advance in the development of present computers. The advantages over electronic valves were size, speed, power requirements and cost. They led to the introduction of increased minaturisation which developed dramatically because of the American space project. The need to produce small lightweight computers for use in space resulted in thousands of components being incorporated on to tiny pieces of silicon.

In 1971, the Intel corporation introduced the microprocessor. These were mass produced and inexpensive with a powerful computing facility (88) which eventually led to the introduction of the microprocessor-based computer. This has fulfilled the aims of Charles Babbage - a machine which could perform an almost limitless variety of tasks and which would be widely available.

ANALOGUE AND DIGITAL COMPUTERS

The computers used in medical applications can be allocated to two distinct groups, analogue and digital. Analogue computers employ specially designed circuits to perform operations such as addition and multiplication and have been used in anaesthesia and intensive care for functions such as the measurement of cardiac output. Analogue computers accept varying voltage levels as inputs so that a signal such as an arterial waveform or a thermodilution curve can be entered directly and analysed by the system.

Digital computers are based on the binary principle where a switch can be either on or off. Vast numbers of switching devices enable such computers to perform mathematical operations and to undertake logical decisions. Unlike analogue computers, digital systems perform no intrinsically useful function and they require a series of instructions in the form of a computer program to direct their operation.

The advantage of a digital system is that it confers great flexibility and the same computer can perform a wide range of different functions simply by changing the program supplied to it. A further benefit is the ease with which data can be stored for subsequent evaluation so that information can be collected, stored and analysed using the same computer system and with mimimal effort once suitable programs have been prepared. Digital circuitry has high noise immunity compared with analogue circuits with better accuracy and is also simpler to modify since their functions can be altered by changing the program (87).

Before the development of the microprocessor, computer applications involved the use of large centralised computers. These were expensive to install and maintain and required the services of a large team of specialist programmers and computer experts. The introduction of microprocessor-based computers has decreased greatly the cost of suitable hardware and allowed them to be used on a much wider basis. The design of simpler programming languages has facilitated the production of useful programs by allowing easier communication between the user and the machine.

A further disadvantage of the mainframe system is that it contains all the processing power and memory in one unit and, when a malfunction occurs, the entire system may cease to operate. Microcomputers, however, can be used as discreet units placed where they are required and linked together when necessary. Therefore, there is less danger of total breakdown of the system if one component fails.

MICROPROCESSORS

The central feature of the microcomputer is the microprocessor unit which performs all the logic and arithmetic functions for the system. Microprocessors can be allocated to two main groups, namely 8-bit and 16-bit devices. A bit is a binary digit that can either be set on or off. Thus a single bit can express the decimal numbers 0 and 1, while two bits can represent the numbers 0-3. With 8 bits (one byte), the numbers 0-255 can be expressed, and with 16 bits, any decimal number between 0-65536 can be represented.

Eight-bit microprocessors handle binary data in blocks of 8 bits and calculations involving large amounts of complex high precision arithmetic take longer to perform than with 16-bit microprocessors. In general, therefore, 16-bit microprocessors have more processing power than 8-bit devices. Most of the microcomputers available are fitted with the less powerful 8-bit microprocessors. At present, the most commonly used 16-bit microprocessors are the LSI 11/02 and the LSI 11/23 (Digital Equipment). Recently, 16-bit devices have been introduced by the major microprocessor manufacturers such as the Intel 8086, the Zilog Z8000 and the Motorola 68000.

However, complete microcomputers fitted with these processors and with adequate computer languages are not yet widely available. The Intel 8088 is a hybrid between the two forms of microprocessors in that its internal architecture is 16-bit, but the external data lines are 8-bit and a number of systems are now produced with this processor.

Communication with the microprocessor is usually with a typewriter-like keyboard but can also be with devices such as a light pen or directly from some form of instrumentation. The results of the processing are usually displayed on a visual display unit (VDU) or printer.

COMPUTER MEMORY

The series of instructions which form the computer program and any data required or produced during execution of the program are stored in random access memory (RAM). This can be considered as a vast array of cells in which information can be stored and from which it can be retrieved. When the power to the computer is removed, the information contained in this form of memory is lost entirely. There is usually another form of memory in a microcomputer which cannot be altered by the user in the normal course of events and is known as Read Only Memory (ROM). It acts as a series of start-up instructions which place the system in a usable form when the power is switched on.

OPERATING SYSTEMS

Also stored in ROM, or automatically loaded into RAM when the computer is switched on, is the computer operating system which can be considered as a program which links together the various facilities including the input and output devices such as the VDU or mass storage equipment. The operating system provides the user with easy access to the microprocessor, the computer memory and the peripheral devices connected to the system.

When a program is run, the processor may not perform any function for much of the time and some operating systems allow several users access to the same processor and memory. This may provide more efficient use of the equipment but the possible disadvantage is that each person may have to wait an unacceptably long time before his requirement is serviced.

COMPUTER LANGUAGES

Microprocessors can be programmed directly using a series of commands called machine code but this is difficult to use. Computer languages are special programs which have been written to allow the user to pass instructions to the computer in a form which is easier for him to understand.

Probably the most widely used and simplest language available for microcomputers is BASIC (Beginners All purpose Symbolic Instructional Code). A typical program line written in BASIC is :-

```
10 IF A B THEN GOTO 100
```

The meaning of this statement at line 10 of the program is obvious from reading the line itself. The equivalent list of instructions in machine code is :-

```
0010  AD  00  C4
0013  CD  01  C4
0016  B0  64  00
```

An error made in the machine code routine is obviously much more difficult to detect than in the equivalent line of BASIC. Most microcomputers have a BASIC interpreter which translates each line of a BASIC program into the equivalent machine code instructions as the program is executed. This has the advantage that it is simpler and faster to develop programs where the program text is similar to English and easily understood.

However, the process of translation considerably increases the time taken for the program to run. This may be unimportant, but where data must be obtained on-line at a rapid rate, or numerous complex calculations have to be performed, the time needed for translation into machine code can lead to loss of data or to an unacceptable time to execute the program.

An alternative to the interpreter is a BASIC compiler. This translates the BASIC program into the equivalent machine code before the program is run so that no interpretation is necessary during execution of the program. A considerable increase in speed can be achieved if the complier is well written but production of the final result can take longer since the program has to be compiled and any extra routines linked in before it can be run and tested.

Where speed is essential and a compiler is not available or is still excessively slow, assembly language can be used. This produces compact and efficient machine code and, in practice, is the method used for writing machine code routines. It employs a system of mnemonics to represent instructions and the equivalent assembler commands to those displayed in BASIC and in machine code above would be :-

```
1  LDA   $C400
2  CMP   $C401
3  BCS   $0064
```

FORTRAN has been the standard scientific language for many years. A wide variety of programs are available but it has a poor ability to deal with text characters. PASCAL is a relatively recent language intoduced by Wirth (95). It is a more powerful structured language than BASIC and may be better for larger projects. Several different forms of BASIC are used which require programs written on one machine to be altered before they can be run on another computer. There is less variation in the

PASCAL language between different computers and this could be a major advantage where programs are to be transferred from one type of machine to another.

All the programs in this work were written in BASIC since it was the only language available for the Apple microcomputer when the various suites of programs were prepared. Also, no programming expertise was available and BASIC was a simple language to allow a non-computer expert to convert concepts into programs. In later programs, where speed of execution or accurate timing of events were essential, the subroutines were written in assembly language and called from the BASIC program.

MASS STORAGE

Permanent mass storage is used for programs and data since these are lost once power is removed from the random access memory. Although magnetic tape can be used for data storage, it may take a considerable time to locate a particular piece of information. A magnetic disc system allows any piece of data to be obtained without sifting through all the intervening information by moving directly to the correct location on the disc where the data is stored.

Small flexible magnetic disc drives are available for most types of microcomputer and can store from 100 000 to about 1 200 000 characters. Larger hard discs can store 5-20 million characters per drive and, although the initial investment is greater, the cost per character stored is much less and the speed of access to the data is faster.

PROJECT AIMS

The overall aims of the project were :-

(1) To develop computer systems which fulfilled educational, clinical and research requirements.

(2) To determine if inexpensive microcomputers could replace large central computers for certain functions.

(3) To assess whether a clinician with no previous experience or training involving computers could develop programs which could serve a valuable function.

Three different areas of experience are presented. These range from relatively simple programs where the attitude of the users to this new form of technology was of paramount importance to a complex data acquisition and analysis system where the user requested the utmost assistance from the computer.

Computing applications can be allocated to two groups :-

off-line where the data are entered into the computer by the operator usually by typing them with a keyboard.

on-line where data are acquired by the computer directly from equipment such as a cardiovascular monitor.

The off-line application of computers for undergraduate teaching in anaesthesia and of similar programs for self-assessment of qualified doctors is examined in chapter 2. The second example describes the use of microcomputers for on-line data collection in the intensive care unit and operating theatres to assist in the management of critically ill patients. Chapter 4 illustrates how microcomputers can be used for on-line control of experimental procedures, automatic collection of data and subsequent analysis of the information.

Many visitors have come to the University Department of Anaesthesia at Glasgow Royal Infirmary to view these applications of microcomputers to anaesthesia and intensive care. All the visitors were either doctors or physicists who had little knowledge of programming but could identify areas of their own practice where computers would be of benefit. They wished to find out how to solve problems or achieve some particular result. The format of program description used in this thesis resulted from these discussions since it proved the most effective method of communication.

The ideal method of assessing the programs would be to observe them in operation. In an attempt to demonstrate the type of information presented to users of the systems, most of the illustrations consist of photographs of screen displays. The screen was photographed with Ektachrome 400 ASA film using 0.25 s exposure and aperture settings of f11-f5.6. The colour slides produced by this method were printed using the Ilford Cibachrome system.

Text and tables for this thesis were produced by the author using an Apple II microcomputer and the word processing program, Apple Pie.

CHAPTER 2

COMPUTER ASSISTED LEARNING AND COMPUTER ASSISTED SELF-ASSESSMENT

SECTION 1 - COMPUTER ASSISTED LEARNING

INTRODUCTION

Education involves the acquisition of information about a subject and from this, the understanding of concepts and relationships between facts.

Students can acquire information about a subject in a variety of ways. Books and journals contain all the information which students will be required to know and have the advantage that each student can progress at his own pace. However, the ease of acquiring information in this manner depends to a large extent on the flair of the author in expressing himself and explaining to the reader what may be extremely complex and subtle concepts.

The student has to sift through the information and attempt to extract and understand the facts which appear to be important to him. This sifting process can be difficult and tedious, and although the student may be successful in this task, he has no feedback during this form of learning to indicate the success or otherwise of his acquisition of knowledge.

Audiovisual aids such as video-cassette and tapeslide lectures can by-pass this sifting process by presenting the important points clearly and unambiguously. They allow items which are of major significance and importance to be strongly emphasised by the use of colourful graphical displays.

A lecturer may also achieve this result but some have a greater knowledge than others and a greater ability to explain and communicate their knowledge and experience to a group of students in a formal lecture. With a large number of students, it may be difficult to progress at a rate sufficient to maintain the interest of the more able students while ensuring that those with less ability can grasp adequately the points made by the lecturer.

Instruction with audiovisual aids is independent of the knowledge and skills of the individual lecturer and has the advantage that the student can progress independently at his own pace of working. However, in a simliar manner to the use of books and journals, when the student has completed such a teaching session, he does not know whether his knowledge and understanding of the subject are adequate.

When the student is taught in a small group tutorial, he can be involved in a true interactive learning process in which his understanding can be assessed as he proceeds. The tutor may, for example, ask a question to which the student has to reply and, depending on the reply given, the tutor may decide to accept the answer as correct, to give the student the correct answer, or to ask further questions to determine if the student really understands the topic under discussion.

With this method of teaching, the student's understanding can be tested as he learns and immediately corrected if wrong. It requires a sufficiently small teacher/pupil ratio and is therefore impractical to use widely because of the large numbers of students usually involved.

The aim of developing computer assisted learning (CAL) for teaching in anaesthesia was to produce interaction between the student undergoing the learning process and the teaching material presented to him. The learning process is more active than reading books or using audiovisual aids since the student is required to think actively about problems presented during the teaching session. The student then becomes involved in a dialogue with the teaching process rather than merely a monologue.

CAL is intended to bridge the gap between audiovisual learning and the tutorial system. This is possible because of the ability of the computer to present information to the student in many different ways, to assess the student's reply and to correct or confirm the answer. It may also present a further series of questions if necessary, by branching off the main flow of the program.

The student can be provided with the reasons why his response has been accepted, or rejected as incorrect. The teaching material is presented in a uniform manner so that if it is well prepared, all students can receive the best instruction in the subject. The student cannot by-pass sections of the teaching session since the computer controls the flow of events.

Casbergue (11) has reviewed the use of CAL for teaching and reported that Christopher (12) had identified three major obstacles to the use of CAL :-

1. undeveloped state of CAL
2. large costs of suitable computers at that time
3. resistance by educators.

He found that a structured experience with CAL developed favourable attitudes in those using it and that this experience decreased previous apprehension about the use of CAL. A further report (71) described the assessment of 256 teachers and school principals and also found that exposure to CAL produced positive attitudes to this form of tuition. The most critical obstacles to the development of CAL were shortage of individuals with the appropriate skills to set up a suitable system and lack of sufficient funds for running the system and for research and development (48). A further factor was lack of suitable software.

Anastasio and Morgan (4) found that the most inhibiting factor was good, readily available programs. They also commented on the large capital investment required and on the need for a change in the existing pattern of instruction and the role of the teacher.

The teacher's role changes from a purveyor of information and record keeping to a specialist in educational management, diagnostics and prescriptive procedures (90). The student's role shifts from passive to active involvement in the teaching process and towards increased control of learning activities. The attitudes of students to CAL and other learning modes were studied by Hess and Tenezakis (30). They reported that students viewed computer instruction more favourably than the teacher or text books alone. This appeared to be caused by the increased opportunity for more personal and creative instruction with the teacher.

The favourable attitude of students to CAL was supported by Bitzer and Bitzer (7) who found that nursing students learned as well or better with CAL than a control group taught in a conventional classroom and took one third to one half of the time. The system they used was Programmed Logic for Automated Teaching Operations (PLATO) which is a central computer with remote terminals connected by telephone modems. These convert electrical pulses into sound tones for transmission along the telephone network and vice versa to receive data. The PLATO terminal is unique to this system and has a grid of infra-red beams across the screen. Touching the screen breaks a beam which indicates to the system the portion of the screen which has been touched. No RETURN key has to be typed and the system therefore is relatively simple to use.

However, attention still had to be given to the operation of the terminal and 54% of the students initially had difficulty concentrating on the lesson. At the end of the course, more than 50% of students considered the computer system to be the easiest to use and the most preferred teaching medium when compared with lectures, films, and text books. Fifteen percent of the students considered the computer to be the worst teaching medium or the most difficult to use. Merola, Pengov and Stokes, (51) reported that medical students who had volunteered for computer supported independent study performed better than students who had received the standard lecture-discussion teaching. However, these were self-selected students who may have been more highly motivated to succeed than those who did not volunteer.

CAL STRUCTURES

Teaching programs for computers can be arranged in many different forms, ranging from the provision of initial instruction in a subject to an assessment of knowledge of facts which should be already well known. The latter use is perhaps of particular relevance in the Medical Faculty since medical students learn a large number of facts but the precise relationship and importance of these facts can be complex and not completely understood by the students. This defect may only be revealed when the student becomes involved with patients in the clinical environment and the use of CAL to simulate medical problems would be be expected to improve the student's ability to deal with real clinical situations,

A computer model suitable for medical teaching could provide a linearly organised simulated patient management structure in which there is minimal deviation from a prescribed path during the teaching session. This allows subsequent detailed analysis of the errors incurred by students which can assist in improving the quality of the teaching material. It also enables analysis of the performance of the students as a group to be easily measured.

The Computer Aided Simulation of the Clinical Environment (CASE) teaching system is an example of a less structured approach to CAL in medicine. It was developed at the Ohio State University School of Medicine by Harless and colleagues (27). In this model, the student assumes the role of a doctor and can request various items of information. When he considers that he has enough information to be able to make a diagnosis, he enters his decision into the computer and the author's assessment of the important aspects of the case and the correct treatment are presented to the student for comparison. It is more difficult to trace the path of the student through this form of clinical simulation and to extract information to enable assessments to be made of the students and the CAL programs.

A further development for CAL in medicine has been the 'MAC' series of models of several human systems (8,20). These provide true dynamic models of physiological and pharmacological systems in which many interrelated variables can be altered, the effects calculated and presented on a visual display unit for detailed examination by the student. Programs have now been developed which allow simpler use of these excellent models for teaching (1).

THE GLASGOW SYSTEM

CAL was introduced in Glasgow in 1974 as part of the National Development Project in Computer Assisted Learning. It was based on a central computer installation located in Glasgow University and remote terminals were sited in various departments in the University, at two major teaching hospitals and at a General Practice health centre. Each terminal had a screen on which the teaching material was presented to the students and on which the questions to be answered were displayed. The student typed his reply to the question into the system using the keyboard in front of the screen. Communication between the central computer and the remote terminals was by Post Office telephone lines using modems which convert electrical signals into sound patterns for transmission of data and vice versa to receive data.

The system was made available on an open access basis to undergraduate medical students and consisted of a case study model in which the user was asked to consider himself, for example, as a doctor in a casualty department. The student was then led through the problems associated with diagnosis, treatment of the principal condition and any complicating features He was required to make decisions on these problems using a simple scoring system. A panel of experts provided the correct scores, with which those entered by the students were compared.

An emergency simulation model was also available to the students which presented problems of clinical management requiring emergency treatment and which involved time penalties which heavily penalised inappropriate actions. However, unlike the 'MAC' series, the time penalties were fixed and this was not a true dynamic model.

The case study model was introduced into the teaching of general practice topics in 1975 and Murray and colleagues (53) reported the improvement in clinical decision making in undergraduate medical students while receiving tuition in general practice with the computer system. However, their study did not compare the CAL system with the traditional methods of teaching and a study was undertaken by Kenny and Schmulian (44) to compare the efficacy of this form of CAL with the traditional teaching of intensive care topics in small group tutorials.

METHODS

The students participating in this evaluation were final year students undergoing instruction in anaesthesia as part of their intensive surgical tuition in Glasgow Royal Infirmary. The topics selected for this comparison of CAL and teaching in small group tutorials were :-

(a) management of self-poisoning

(b) management of a serious burning accident with associated pulmonary damage.

Sixteen students completed the evaluation and were randomly allocated to two groups, A and B. Students in group A received tuition on the management of self-poisoning in a small tutorial group and the management of burns by CAL. Group B students received tuition on the management of burns in a tutorial group and the management of self-poisoning by CAL.

The topics were taught on two successive days and two days later, the students' knowledge of the topics was tested by a questionnaire. Because of the constraints of the medical curriculum, the assessment had to be completed within the week assigned to anaesthetic tuition and a later reassessment, which would have been desirable, proved impractical. The content of the tutorials was based entirely on the text of the corresponding CAL program to ensure that the information given to the students was comparable.

The case study program had between eight and seventeen sections (nodes) and the text and its length could be modified according to the preference of the author. At each of the nodes there were between five and nine options. The student had to grade each option on a 5-1 scale according to the decisions he thought appropriate. The scoring system was :-

5. must do / must consider
4. should do / should consider
3. could do / could consider
2. should not do /should not consider
1. must not do / must not consider

The students had two opportunities to score the options. If their first attempt to score all the options was correct, they were congratulated and continued to the next node. If the student was incorrect in his decisions, he received feedback after his second attempt at grading the options. There was feedback available for every option and this gave the correct information to the student if his decisions were incorrect.

RESULTS

Fifteen of the sixteen students obtained a greater score following CAL than with the traditional method of teaching (table I). The scores achieved by the students using CAL were compared with the scores gained by the same students after teaching by tutorial. The scores obtained after CAL were found to be significantly greater (unpaired t-test) than the scores achieved after traditional teaching (table II).

Comparision of the results obtained after CAL with the results after traditional teaching within the same teaching topic also showed significantly greater scores after CAL (table III).

A comparison between the scores obtained after CAL for group A and group B showed a significant difference (unpaired t-test; table IVa). This suggested the possibility of either a difference between the abilities of the students in the two groups, or else a difference between the degree of understanding of the CAL programs by the two groups.

However, when the scores obtained by group A following tutorial teaching were compared with the scores achieved by group B after tutorial teaching, there was found to be no significant difference between the groups (table IVb). This would suggest that there was no significant difference between the abilities of the two groups of students. Thus, the difference between the CAL scores for the two groups could be attributed probably to the more didactic presentation of the CAL program dealing with the management of the burned patient compared with the more complex and involved structure of the program on the management of self-poisoning.

Group A			Group B		
Student	CAL 'poisoning'	Tutorial 'burns'	Student	CAL 'burns'	Tutorial 'poisoning'
1	74	35	1	85	51
2	69	58	2	81	72
3	59	60	3	77	38
4	69	42	4	77	51
5	68	50	5	69	49
6	64	40	6	73	54
7	49	40	7	50	49
8	62	56	8	88	49

Table I. Students' percentage scores in test questionnaire.

	Group A	Group B	Groups A+B
CAL	64 +/- 2.6	75 +/- 4.0	70 +/- 2.7
Tutorial	48 +/- 3.2	52 +/- 3.0	50 +/- 2.3

Table II. Average scores obtained after CAL and tutorial teaching, mean +/- S.E.M.

'Poisoning'		'Burns'	
Group A CAL	Group B Tutorial	Group A Tutorial	Group B CAL
64 +/- 2.6	52 +/- 3.0	48 +/- 3.2	75 +/- 4.0
0.01 <P> 0.02		P> 0.001	

Table III. Comparison for each topic of average percentage scores obtained after CAL and tutorial teaching, mean +/- S.E.M.

(a) Group A	Group B	(b) Group A	Group B
64 +/- 2.6	75 +/- 4.0	48 +/- 3.2	52 +/- 3.0
0.02 <P> 0.05		no significant difference	

Table IV. (a) Average percentage scores for Groups A and B following CAL, mean +/- S.E.M.

(b) Average percentage scores for Groups A and B following tutorial teaching.

DISCUSSION

The principal difficulty of an evaluation of two different educational methods is to ensure the validity of the comparison between the methods. This was achieved in this study by presenting the same information for both the tutorial and CAL to the two groups. Thus, a comparison between the scores and the educational methods is valid. The results indicate that in the context of this study, CAL was superior to the tutorial method of teaching. The student using CAL had to respond actively to the question presented to him on the screen. If his response was incorrect, he was given immediate corrective feedback. This can be compared with the student who may sit passively listening to a tutorial and remain unaware of the deficiences in his knowledge (54).

Simpson (82) has emphasised the importance of immediate feedback for learning. He reported the assessment of Davies (16) that the overall efficiency of teachers is about 12% and stated that learning was more efficient and long lasting when the student was kept well informed about his abilities and disabilities, how he was approaching his objectives and how much further he still had to go. These educational criteria are fulfilled by the case study model which interacts with the student so that those areas in which his knowledge of the subject under study are deficient, can be defined and corrected at the same time.

Walton (93) stressed the educational benefits of teaching in small groups. However, with the more traditional method of tutorial teaching, students often sit passively and may be reluctant to become involved and commit themselves to any one course of action. They frequently do not participate unless forced to do so and may leave discussion to one of their more vocal colleagues. One impressive feature noted when the students used the computer was the often heated discussion it provoked.

It did not have the aura of a tutor which may somtimes act to suppress discussion by students. All of the students usually became involved and contributed their own opinions on the problem under discussion. To a certain extent, the clinical simulation was viewed as a game to be played against the computer, but the clinical lessons learned by the students were real.

There were several disadvantages to the system, of which the major one was expense, both for the large capital cost of the central computer and the recurring costs of rental of telephone lines, modems and computer staff (37). The advent of the microprocessor has allowed smaller and very much cheaper microcomputers to provide an alternative approach to CAL. The initial work on the evaluation of the use of a microcomputer for anaesthetic teaching illustrated that this had the potential of being even more successful than the use of a centralised system (41).

Accessing CAL programs with a remote terminal and central computer is more complex than with a microcomputer. The user must be familiar with the operation of a modem and know how to establish contact with the central computer. They must be able to control the central system and select the correct program. Large systems do not usually respond well to even minor errors which can cause complete loss of the program. The operating systems which control the interface between the users and the computer are much more complex in the large centralised systems and decrease their reliability compared with microcomputers.

Casbergue (11) suggested that a technically competent person is needed to assist teachers and students to overcome frustrations which can occur with errors in input or output, unanticipated disconnections or system failures since such frustrations can cause loss of interest in the use of CAL. Microcomputers have less electronic components than a central computer and do not require connection to the telephone system or other forms of electronic communication. Because of these features, they are inherently more reliable than a central system.

The response time with a microcomputer is virtually instantaneous whereas a central system may have an unacceptably long delay between the student's entry and the computer's reply because of the number of terminals connected to the single processor. If it is necessary to increase the number of users connected to a central system, eventually some items such as discs or even the entire computer will have to be replaced. When a microcomputer system is used for CAL, each unit is completely independent of the others and the system can be expanded simply by purchasing more of the same units.

The students' use of the central system in Glasgow was dependent on the presence of trained computer staff to load the programs and was therefore restricted to certain times of the day when students were not attending lectures or clinical teaching. The computer language used was FORTRAN and specialised programmers were required to convert the text produced by an author into a program suitable for entry into the central computer. It proved difficult and time consuming for an author to have the program revised and updated subsequently.

A further disadvantage of the central system was that items such as e.c.g. recordings or X-ray films could only be described to the student in the text of the CAL program. An e.c.g., for example, could only be reported as showing ventricular extrasystoles or atrial fibrillation; the student's ability to interpret an e.c.g. could not be tested. It would obviously be more satisfactory to have such visually important material presented to the student in picture form so that he could examine them and then have his understanding of the record or film questioned by the computer and corrected if necessary.

This is possible using the microcomputer. Interface circuits were developed which allow the computer to control an unmodified Caramate slide projector and so present slides of items such as radiographs to the student when required (fig. 1; 41). The circuit could also be used to allow the computer to control the delivery of a tape slide lecture and halt the presentation to

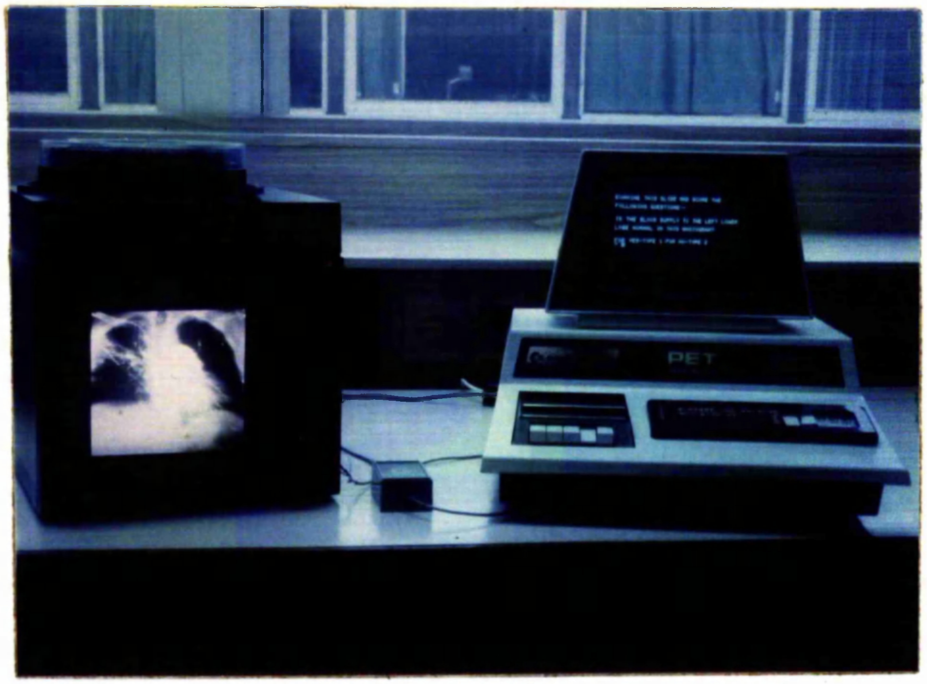

Figure 1 Computer control of Caramate slide projector.

Figure 2 Apple II microcomputer, disc drive and floppy disc.

give the student questions about the teaching material he has viewed. A similar interface has been produced which allows the same degree of control over a video-cassette lecture (17).

Thus, not only did the introduction of microprocessor-based computers dramatically reduce the cost of the equipment and improve its reliability, it also extended the potential of CAL by interfacing to other audiovisual teaching devices. Since microcomputers are easily transported, completely self-contained and do not require connections to telephone lines, they offered the use of CAL on a much wider scale than was possible previously.

The Commodore PET was one of the first comparatively inexpensive computers available and our initial experience using a microcomputer was obtained with this machine. The memory capacity was 8k bytes which allowed slightly more than 7,000 characters to be stored in the memory of the computer. The size of a typical CAL program was about 20k bytes and it had to be loaded and run in three separate sections. A BASIC interpreter was stored in read only memory (ROM) and the CAL programs were written in this computer language. The feasibility of using a microcomputer for teaching was demonstrated with this machine but the cassette deck supplied did not prove sufficiently reliable for routine teaching.

A wide range of microcomputers is now available and the rapid development of this technology makes it difficult to select an optimum system. Guidelines have been published which identify the important aspects of various systems (38). The Apple II appeared to offer the best combination of cost, memory size and flexibility for possible future projects. It has a 6502 microprocessor and 48k of RAM available. Memory addressable slots are incorporated into the system which enable a wide range of additional interfaces such as communication cards, clock/calendars and analogue to digital converters to be fitted.

A further advantage was the provision of high resolution colour graphics as a standard function. It also had a BASIC interpreter stored in ROM which had been written by the same company which had produced the interpreter for the Commodore PET. No computer operators were needed with this system since the students quickly learned to place the magnetic discs into the disc drive unit. The only other operation required was to switch on mains power.

The Apple II microcomputer, disc drive and floppy disc are shown in figure 2. The system is entirely self-contained and requires only mains power to function. The disc operating system (DOS) is loaded from the magnetic disc and this automatically runs the greeting program which on our systems is always named the HELLO program. This program then calls the required CAL program from the disc.

The original CAL material written for the Commodore PET computer could not be stored as text files on the cassette tape drive and had to be incorporated into the BASIC program. The CAL programs which had been produced for the PET were transferred to the Apple II with the assistance of Strathand Ltd. The programs were listed to the IEEE port of the PET and the output connected to the keyboard of the Apple computer. The two forms of BASIC were sufficiently similar and only slight modifications were required to allow them to run satisfactorily on the Apple II. This automatic transfer prevented the need to re-type manually the entire programs into the Apple.

An attempt was made to evaluate any differences between attitudes of the students to the large central system and the microcomputer system but was abandoned because of the continued failure of various components of the central system at the times students were available for tuition. All development was then transferrred to the Apple II microcomputer systems.

The programs on the central computer had only provided feedback to the questions if the student had answered incorrectly and the students expressed the view that they may have entered the correct answer for the wrong reason. The CAL programs developed for the Apple were therefore organised to always provide feedback once the questions had been answered. Other features were incorporated such as the ability to recall the text which had been displayed previously and to have several forms of question presentation.

The overall organisation of the CAL program was to allocate to a series of character strings the text, questions and feedback to be presented to the students. These were then entered into subroutines which organised the presentation of the text and questions, accepted and scored the answers and provided appropriate feedback.

A short case history was displayed which gave the essential features of the patient's condition. A main topic question was placed at the top of the screen followed by a series of individual questions, all of which had to be answered before progessing to the next section. The answers entered were compared with the scores deemed correct by a panel of experts (as with the central system). However, after each reply, the student was always informed if his reply was correct or not and the feedback for each individual question was always displayed.

One feature which cannot be overstressed is the need to ensure that any computer system is simple for a non-computer expert to operate. Tebbutt (86) has emphasised that the aim should not be to teach the users to deal with computers in the form in which they have normally been presented, but to make computers more acceptable to the people who will use them.

In an attempt to make the CAL system as easy as possible to operate, a simple selection menu was always displayed which provided a guide to the options available at that time. The teaching programs were controlled using the GET command which obtains a character from the keyboard without the need to type the RETURN key. The character is then checked to ensure that it is within the predetermined acceptable range.

Only the numeral keys are required to control the teaching programs, so that after a few minutes, the students could use the system entirely on their own. CALOB-TEXT is an example of this form of CAL program and will be descibed to illustrate the construction of the program.

CALOB-TEXT

As explained previously, the CAL programs had been originally developed for use with the Commodore PET computer. The first versions of this machine had an error in ROM which prevented the satisfactory storage of text files on cassette tape and the text required for teaching had to be incorporated within the program itself. The information was allocated to string variables and passed to subroutines which presented the text on the screen, obtained the student's response from the keyboard and displayed the required feedback.

The string variables used and their functions are :-

T$(X)	contains the blocks of text which provide information such as the case history.
G$, H$	contains the main topic questions which provide headings for the subsequent individual question strings.
X$(X)	contains the individual questions to be answered by the students.
B(X)	contains the scores deemed correct by the experts.
A$-E$(X)	contain the feedback text.

The program is controlled by the student entering various numbers using the keyboard. The GET command collects a character from the keyboard buffer and the first action of the program is to clear the keyboard buffer in case a key had been typed before or during loading of the program. If this had occurred and the key typed was a valid entry, the program would proceed immediately to the next section.

The reply given by the student is stored in an array, Q$(X) and this is dimensioned together with the array used to store the text statements, T$(X). The array, Q$(X), already had 8 entries from a previous program and subsequent entries are indexed by the variable, 'I'.

The program passes to the subroutine at line 9000 which explains and illustrates the scoring method used in this program (fig. 3). Once the student has understood the instructions and is ready to proceed, the program passes to the subroutine at line 90. This is used frequently by the program and prints the message :-

'WHEN READY TYPE 1'

The program waits at this line until the student touches the numeral, '1' on the keyboard. The program then returns to the calling routine.

Data are allocated to the first text strings, T$(1)-T$(9) and these are printed on the screen (fig. 4). Once again, the program passes to the subroutine at line 90 until the numeral '1' is typed. G$ is set equal to the first topic question string :-

'THE MOST EFFECTIVE ANTACID IS ?'

Two forms of question presentation are possible with this program and the form to be used is indicated by setting the flag 'F' to 1 if only one answer is to be entered from those displayed on the screen. For the first question block of CALOB-TEXT, only one answer is correct and this value is allocated to the variable, 'B'. The computer therefore displays the main topic question and the group of individual questions from which the correct answer is to be selected (fig. 5).

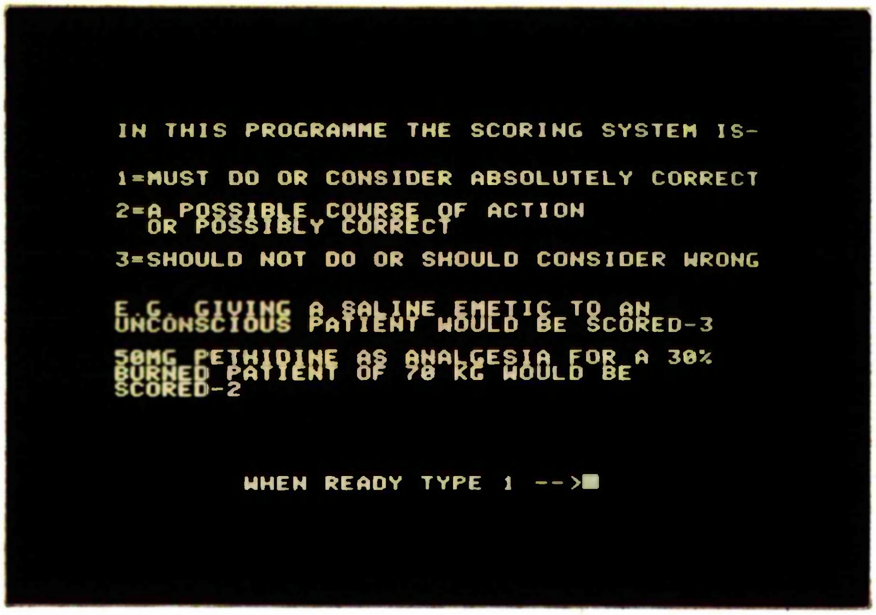

Figure 3 Scoring system used in CALOB-TEXT.

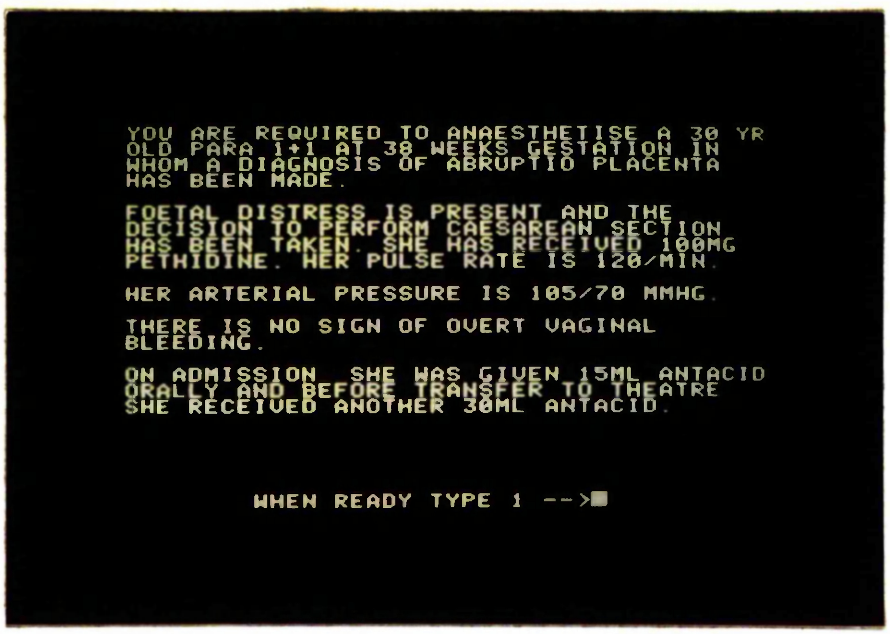

Figure 4 First text page of CALOB-TEXT.

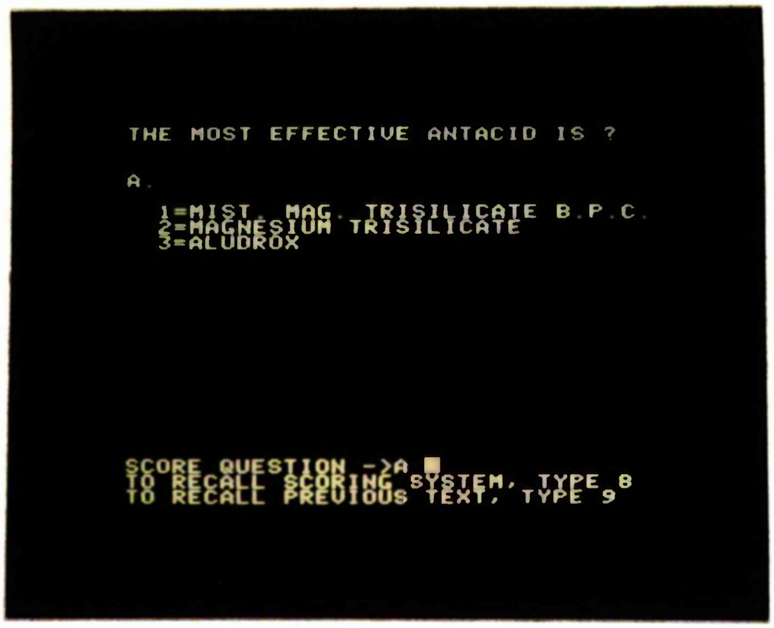

Figure 5 Main topic question and first question group of CALOB-TEXT.

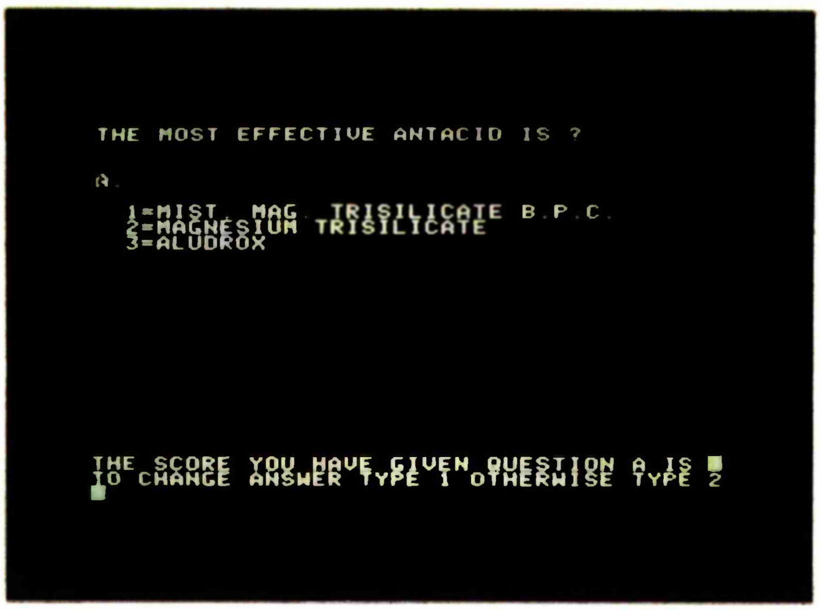

Figure 6 Choice given to student to change his answer.

The original program, as stated earlier, scored the questions as follows :-

5 = must do / must consider
4 = should do / should consider
3 = could do / could consider
2 = should not do / should not consider
1 = must not do / must not consider

However, students who used this scoring system suggested that the scoring system should be reversed so that a score of '1' would indicate 'must do / consider'. This was arranged most simply for this program by altering the allocation of scores in the display/score handling subroutines at lines 30000 and 50000. It was also decided for this program to allow only three choices and the scoring system used was :-

1 = must do or consider absolutely correct
2 = a possible course of action
3 = should not do or should consider wrong

The individual question strings are allocated to the array, X$(X) if several questions are to be asked, or to the single string, X$, if only one answer is required. The arrays A$(X) to E$(X) store the feedback strings unless only one question is asked when the single variable strings, A$-E$, are used. Once the required text strings and values have been allocated, they are passed to the relevant handling subroutine.

SINGLE QUESTION

When a single question is to be answered, the program passes to line 50001. The subroutine at line 40000 is then called and the student is informed that only one answer is required for this section. The scoring system and the previous text can be recalled at this time to refresh the student's memory. The single question is displayed on the screen and the student enters his reply. This is obtained from the key board at line 52000 using the GET command.

The character which has been typed is allocated to the string variable, Z$. A real variable is not used since the entry of a letter would cause a formatting error to occur. When a string variable is used to store the entry, it can be subjected to a variety of tests to ensure that it is a valid character. If it is invalid, it is ignored and a further character is obtained.

Entry of the numeral '9' causes the previous text to be displayed page by page on the screen by the subroutine at line 60000. The return address is then removed from the microprocessor stack by the POP command and the entire routine is re-run from line 50001

When the student wishes to review the scoring system, he enters the number '8'. The display used at the beginning of the program to explain and illustrate the method of scoring is repeated. Once again, the return address is removed from the stack and the entire routine is re-run from line 50001.

The student's answer is allocated to X% and is displayed to allow him the reply to be altered in case an incorrect key was typed (fig. 6). If the answer is to be changed, control is returned to line 50001.

When the student is satisfied with his reply, the answer is corrected to the old form of scoring at line 50030 and compared with the score allocated by the experts. A correct score is indicated to the student and this is stored in the array, Q$(X). If the answer is incorrect, a counter, W% is incremented, and the word 'INCORRECT' is displayed on the screen. The feedback strings are printed (fig. 7) and the program waits until the student types the number '1'.

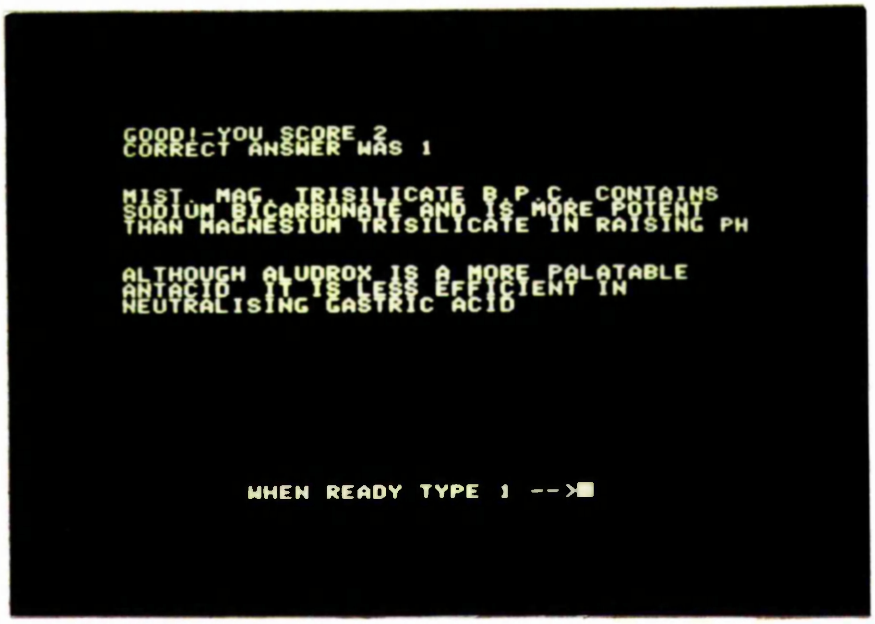

Figure 7 Feedback text and correct answer.

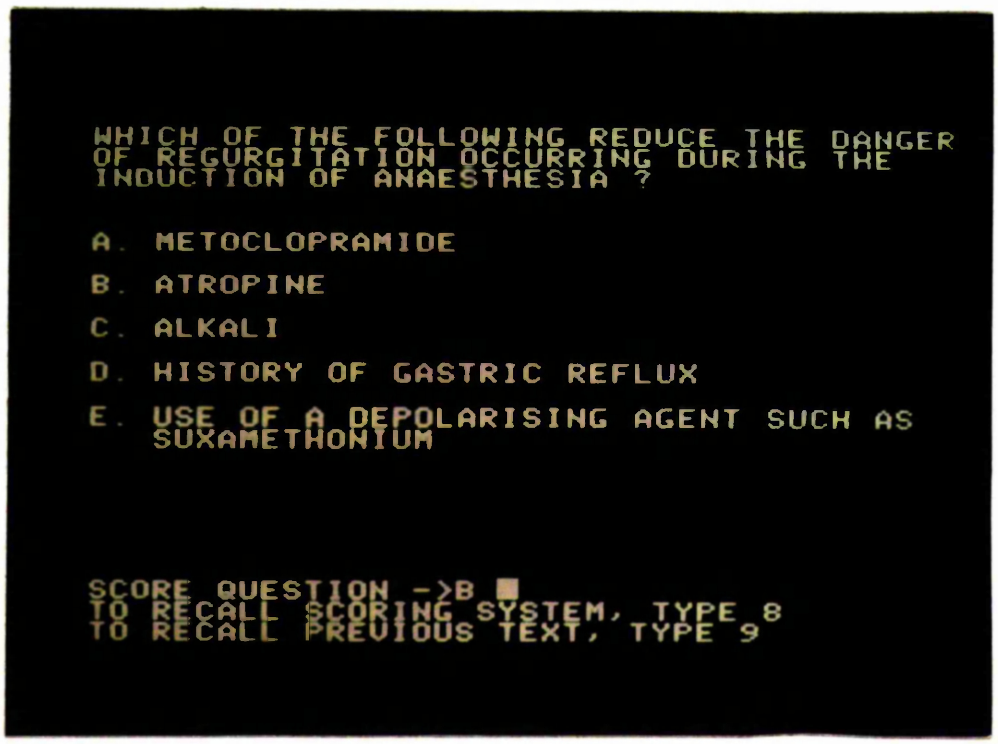

Figure 8 Form of question presentation when multiple questions are to be answered.

MULTIPLE QUESTIONS

The alternative form of question presentation is to display several questions together on the screen. Each question has to be answered in turn and the variable, 'F' is set to zero. The individual questions are allocated to the array, X$(X), and the correct scores to the array, B(X). Feedback for each individual question is allocated to the arrays, A$(X)-E$(X).

The strings which have been set previously in the program are handled by the subroutine at line 30000. The topic question strings, G$ and H$ are printed at the top of the screen and the individual question strings are printed below (fig. 8).

The student is instructed to score each question in turn and has the same ability to alter his score before it is finally compared with that of the experts. The previous text and the scoring system can be reviewed if required. Feedback is provided for each individual string and a counter is incremented if the replies are incorrect. The score achieved for each question is allocated to the appropriate element in the Q$(X) array.

When all the questions in each section have been answered, control is returned to the main program and a new set of text, topic, question and feedback strings and scores are allocated to the string variables and the process is repeated. When all the sections have been completed, the student is given an overall score and this, together with the scores for each question, is written on to the floppy disc for a permanent record. The structure of the programs have been translated into a form suitable for use with the inexpensive Sinclair ZX81 microcomputer and have been reported to be of value in teaching junior anaesthetists preparing for examinations (100).

Other languages exist which have been specifically designed for teaching. Some, such as COURSEWRITER (28) and MENTOR (22), are available only for large mainframe computers, but others, such as PILOT (83) have been successfully modified to operate with microcomputers. These have simpler structures than standard computer languages and are therefore easier for non-computer experts to produce their own teaching material. PILOT was available for the Apple computer but did not have the facility for recording the individual responses of students on disc.

REVISION AND MODIFICATION

The time and effort required to write and print a text-book or to prepare a tapeslide or video-cassette lecture is considerable. Inevitably, revision becomes essential because of completely new information or new interpretations of previous information. Such modification can be extremely expensive and time consuming. For example, to update a tapeslide presentation may require not only the preparation of new slides, but the re-recording of the entire sound track. Alteration of a video-cassette lecture is even more complex since the camera crew, lighting, sound and production staff as well as those taking part in the film have to be assembled and the changes edited into the original film with possible degradation in quality of the final result.

CAL programs, however, have the considerable advantage that only the section which requires revision need be altered. Modifications for the central computer system had to be passed to the programming team who may have been involved with other projects and unable to update the programs immediately. Program texts which had been written in BASIC on the microcomputer proved relatively simple to modify and the ammended program could be rapidly saved on to magnetic disc. Not only the text, but the structure of the program was easy to alter to include suggestions made by students and other users of the system. This improved the presentation and display of the information and facilitated the use of the program.

ANALYSIS OF PERFORMANCE

A principal benefit of this form of CAL for the student is that he receives positive feedback as he progresses through the teaching session and his total score when the teaching session is completed. However, the teacher normally has no similar feedback mechanism to inform him of the success or otherwise of his teaching methods. The CAL programs used with the Apple II store the result for each question which the student answers in the computer memory. At the end of the teaching session, the entire block of scores is written on to the magnetic disc. The scores can then be analysed to provide information about the students' performances. Areas of weakness can then be identified and individual questions analysed to detect those which were poorly answered.

The success of CAL for teaching medical students led to the development of the concept of computer assisted self-assessment (CASA). This involved programs with the same structure as those used for undergraduate education but presented to allow qualified doctors to assess their own level of knowledge in anaesthesia and intensive care. The ability of the computer to collect and analyse the performances of the participants in CASA was central to this part of the project.

SECTION 2 - COMPUTER ASSISTED SELF-ASSESSMENT

INTRODUCTION

The Alment Report on Competence to Practise (2), and the Royal Commission on the National Health Service (72) have suggested that regular peer review sessions are of value in the maintenance of standards in medical practice. While these sessions should be educational, it is important that the practising doctor has access to an objective measure of standards of knowledge and skill (76).

These reports have supported the use of self-assessment and recommended that the procedures be initiated voluntarily by practising doctors rather than by imposition from outwith the profession as has occurred in other countries.

Medical audit implies the concept of an external systematic examination of performance. Peer review suggests a review of a doctor's performance by his colleagues where clinical decisions are debated with a view to improving subsequent patient care. Quality assurance involves the assessment of a doctor's performance and its comparison with an acceptable standard. Self-assessment constitutes a major aspect of quality assurance but the method chosen for self-assessment must ensure the participants of anonymity and confidentiality if the results are to be collated and subjected to subsequent analyses.

Many conventional forms of self-assessment exist including multiple choice questions (MCQ) and case presentations in journals. Forbes (24) reported the use of MCQ for self-assessment by the Royal Colleges of Physicians but this was not universally accepted by the participants. Forbes suggested that a method which incorporated case histories was closer to real clinical practice and would be more beneficial.

Harden (26) posted patient management problems to general practitioners. Each doctor had to decide on the mangement of the patient and compared his responses with those deemed correct. Feedback was available on the forms sent to the doctors. Of those who were sent the forms, 18.1% returned their responses with additional comments. Most were favourable and the approach was considered educational and stimulating. However, since it is difficult or impossible to obtain information about the performance of those using these methods, they cannot readily be used for quality assurance.

Computer-assisted learning (CAL) had already been shown to be of value in medical education (53, 44), and the advent of microprocessor-based computers provided the possible mechanism of an objective method of self-assessment while fulfilling the function of continuing postgraduate education. A preliminary study of the use of computer-assisted self-assessment (CASA) was undertaken at the 1979 Annual General Meeting of the Association of Anaesthetists (10).

METHODS

The design of the test programs was the same as those used with the undergraduate medical students and simulated the diagnosis and management of clinical problems in anaesthesia and intensive care. A short case history was presented to the participants who were then asked to score a series of questions using the numeric keys of the computer. All other keys had been blocked. Those taking part in the evaluation were asked to complete a questionnaire presented on the computer and asked for the following information :-

1. place of undergradute education
2. place of postgraduate education
3. place of present employment
4. grade of anaesthetist
5. whether specialist in the topic under assessment
6. number of years anaesthetic experience
7. number of years post-Fellowship

This allowed comparison of the results obtained by different groups of anaesthetists. Four microcomputers were available, arranged in separate booths to ensure privacy for the participants, and two CASA programs were used by the 56 participants. The first dealt with aspects of obstetric anaesthesia and was scored on a three point scoring system. The second program involved the mangement of a patient with endotoxic shock and was scored on a five point scoring system. Full explanation of each scoring system with illustrative examples was given at the beginning of each case.

The response which the participant entered for each question was compared with the response deemed correct by an authoritative panel of doctors practising in the sub-specialty concerned. After answering each question, the anaesthetist was informed immediately whether or not his reply was correct and the reason for this. The score for each question together with the total score was stored on magnetic disc for later analysis.

When both CASA programs had been completed, the anaesthetists were asked to complete a further questionnaire to obtain their subjective views of the presentation.

In March 1980, a similar assessment was undertaken at the Annual Scientific Meeting for Junior Anaesthetists. Three computer systems were available and 30 participants underwent self-assessment with two programs; one on the management of burns with pulmonary complications and the same obstetric program used at the senior meeting. Subjective questionnaires were completed by the participants. Both the groups who used CASA at the two Scientific meetings had attended them voluntarily. Furthermore, they had then selected themselves to undergo the self-assessment process.

During March 1981, the system was taken to four district general hospitals in the West of Scotland. In this part of the trial, a total of 53 anaesthetists completed two programs - the obstetric anaesthetic case study and one on the management of a post-tonsillectomy haemorrhage in a child with sickle cell trait. The anaesthetists comprised the entire complement of the anaesthetic staff of the four hospitals with the exception of those who were on leave or whose theatre commitments prevented them from participating. This group were different from the previous two groups in that they were not self-selected.

A further trial of CASA was conducted during the Faculty of Anaesthetists' Annual Assembly for Regional Educational Advisers and Faculty Tutors held in July 1981. This was undertaken because of the specialist educational interests of those attending the meeting. Sixty anaesthetists completed the same programs which had been used in the district general hospitals.

At the three formal meetings where CASA was available, the number of participants was limited by the free time and number of microcomputers available. In total, 199 anaesthetists have taken part in the study.

RESULTS

All who took part in the evaluation completed the obstetric program and the results of the analyses of this program are presented. Programs were written to analyse the results obtained by all the participants and by the smaller subgroups to which they were allocated based on their replies to the computer questionnaire.

CASAGRAPH

This program was written in conjunction with Mr. Paul Davis of the West of Scotland Health Boards' Department of Clinical Physics and Bio-engineering. It enables the distribution of the total scores recorded for the participants in the different self-assessment trials to be graphed. The mean, standard deviation, standard error of the mean and total number of participants are also displayed.

CASA-ANALYSIS

The data which was written on to the discs during the self-assessment sessions was analysed by this program. The data collection routine from lines 20-370 places the final percentage scores obtained by each doctor into the array Q$(I,J).

Random access files are employed and checks are made to ensure that only the scores obtained from anaesthetists are used since later trials allowed some other specialists to attempt the programs. Record number zero stores the character 'A' if the participant was an anaesthetist and the file is only used in the analysis if this character is present. Record 50 stores the character 'Y' if the anaesthetist had used the CASA program previously, and once more, the file is not used in the analysis of results. The entries for the computer questionnaire were stored in records 1-7 and the results for the obstetric program in records 8-49.

Thirteen of those who took part had previously used the self-assessment programs and were excluded from the analyses. Figure 9 is an example of the results obtained using CASA-GRAPH. It shows the distribution of scores and the overall average score for the anaesthetists who took part at the Senior Scientific Meeting. The results obtained with the obstetric program are displayed in table V. This reveals an acceptable level of knowledge of the subject by all the groups taking part in the assessment. In particular, the unselected anaesthetists in the District General Hospitals achieved an average score close to those who selected themselves for the assessment.

The scores were then analysed using the program CASA-ANALYSIS to compare the results of the individual subgroups to which the participants had been allocated by the qusetionnaire. The most striking comparisons of the results obtained from the obstetric program, which had been used by all the participants in the study, were between those who had practised for different periods of time. Anaesthetists who had

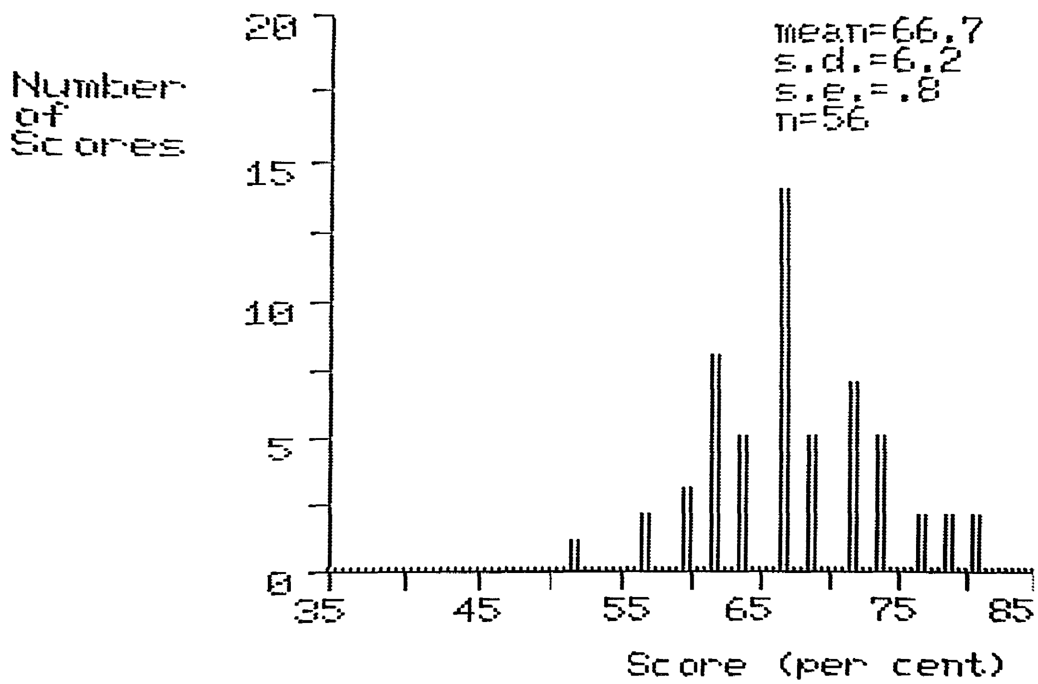

Figure 9 Computer-generated distribution of total scores obtained by participants at the senior Scientific Meeting plotted directly from the computer.

Scientific Meetings	Regional Advisors and Tutors	DGH	All
68.4 +/- 0.9	66.2 +/- 1.0	66.1 +/- 1.4	67.1 +/- 0.6

Table V. Percentage scores achieved in the OBSTETRICS assessment program, mean +/- S.E.M.

Participating group	% Score	No	P value
Practising 0-4 yr	70.4	33	
v practising 10-14 yr	66.4	41	<0.05
v practising 20-24 yr	65.2	22	<0.02
v practising 25-29 yr	65.2	12	<0.05
v practising 30-34 yr	61.6	14	<0.01
v practising 35-39 yr	61.4	7	<0.05
Practising 15-19 yr	68.3	21	
v practising 30-34 yr	61.6	14	<0.02
Registrars and SHOs	70.4	33	
v consultants only	65.7	127	<0.05

Table VI. Analysis of performance in OBSTETRICS program.

practised for less than 5 years scored significantly better than those with more than 10 years experience. The detailed results are shown in table VI.

The individual anaesthetist may wish to compare himself with the group whose years of experience match his own. When the mean score of each group in the obstetric program was graphed against years practising anaesthesia, an inverse relationship was found ($r = -0.95$; fig. 10).

OBJECTIVE QUESTIONNAIRE

A principal aim of the study was to measure the acceptability of the method for self-assessment in anaesthesia. The results of the questionnaires are shown in table VII.

The participants at the two scientific meetings had selected themselves to undertake the assessment. The acceptability of the method to those who took part was 93% at the Senior Meeting and 100% at the Junior Meeting. The Faculty Tutors and Advisers were also self-selected and their acceptability was 91%. Although those who took part in the district general hospitals were not self-selected, the acceptability for this group was 96%.

At the Senior Meeting one of the four video display units had not functioned properly and the legibility of the text was not considered satisfactory by 31%. This would tend to verify that the responses to the questionnaire were a true indication of the opinions of the participants. When the defective video display unit was replaced at the Junior Meeting, the legibility of the text was considered satisfactory by 100%.

The anaesthetists at the Faculty Meeting were a group who had a particular interest in the educational aspects of CASA. They were asked if they would consider this form of self-assessment helpful in their own teaching. Sixty seven percent replied that they would. They were also asked if they would support central organisation of the programs by the Faculty

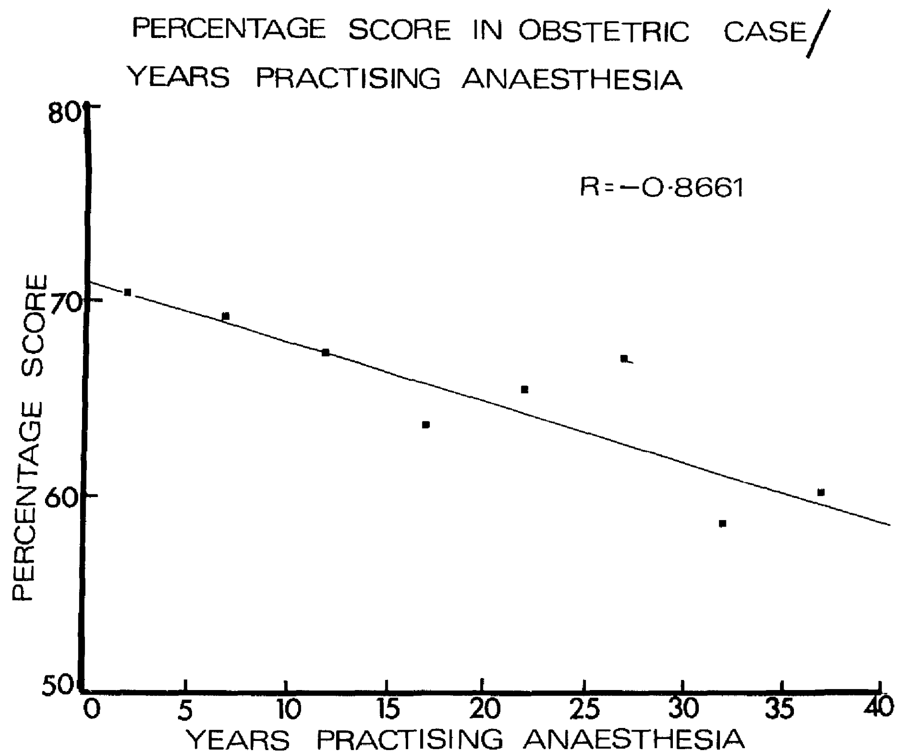

Figure 10 Mean percentage scores in obstetric anaesthesia / years practising anaesthesia.

	Senior meeting	Junior meeting	REAs and Tutors	DGH
Did you find the presentation helpful/unhelpful?	93%	100%	91%	96%
Is the scoring system for the obstetric case acceptable/ unacceptable? (3-point scale)	88%	86%	84%	89%
Is the scoring for the second case acceptable/ unacceptable? (5-point scale)	84%	54%	62%	36%
Is the legibility of the text on the screen satisfactory/ unsatisfactory?	69%	100%	---	---

Table VII. Response to subjective questionnaire: percentage of positive replies.

of Anaesthetists, and 72% considered this would be worthwhile.

DISCUSSION

Shaw (78) has reviewed the subject of medical audit and has proposed that it should be voluntary, objective, repeatable and undertaken by clinical peers. It has also been proposed (75) that in addition, the method should :-

1. assess adequately the level of knowledge of the participants.
2. have some educational value.
3. be acceptable to those who use it.
4. allow easy retrieval of information on performance to permit analysis of results.
5. be easily used throughout the country at relatively low cost.

The purpose of this study was to assess the use of microcomputers for quality assurance to determine whether these criteria could be met by a relatively inexpensive computer system which was transportable and simple to operate, unlike large centralised computers.

There was an inverse correlation between years of experience in anaesthesia and scores obtained for the obstetric program. This might have been anticipated since the younger anaesthetists are studying for the Fellowship examination or are engaged in research which particularly requires them to remain up-to-date with current advances. These pressures might have largely been removed from many older anaesthetists unless they are actively involved in academic work or research.

This is not to say that a high score in self-assessment programs can be directly equated with clinical competence or vice versa, since anaesthesia is a practical specialty. However, the level of professional competence must be related to some extent to theoretical knowledge. The obstetric programme was selected for use in all the trials since the concepts of obstetric anaesthesia are central to good anaesthetic practice.

The analyses suggest that CASA is able to differentiate between varying levels of knowledge of the participants, which was the first requirement stated. The method of presentation of the program has already been shown by several workers to have educational value.

CASA was well received both by those who were self-selected and by those who were not. Ease of operation of the system would allow doctors to undertake self-assessment on an open access basis when time permitted. Most users enjoyed the programs and found them helpful, although a few considered CASA to be of limited value.

The programs were designed to simulate real clinical problems which the anaesthetists had to consider and evaluate before replying and they became actively involved in the self-assessment/teaching process. The scores for each question from each user could be rapidly retrieved and analysed unlike printed programmed texts, the results from which would require considerable time and effort to analyse.

The method requires a more extensive trial to determine how effectively it can be used throughout the country. However, the programs have been sent by post to many other centres in the U.K. and abroad where they have been used without direct guidance. The responses from such centres have been equally favourable and the possibility exists for the use of CAL/CASA programs in District General hospitals which may be remote from a main teaching centre. Consultants in such hospitals may be unable to devote the time required to provide lectures and tutorials for their junior staff. A computer system could supplement their teaching efforts and assist trainees to acquire the necessary theoretical information.

Of the two scoring systems used, the simpler 1-3 scale was more popular but the 1-5 scale may permit more discrimination when dealing with situations demanding clinical judgments when there is no absolute correct answer.

Computer-assisted self-assessment combines the functions of continuing postgraduate education and quality assurance. It would appear to be able to differentiate between varying levels of knowledge and it was acceptable to most of the anaesthetists who took part. The system allows information to be collected and analysed relatively simply. It may offer a possible solution to the requirements for a method of quality assurance in anaesthesia and in other specialties. The quality of the assessment programs is of major importance in determining this aspect of quality assurance and the mechanism for achieving acceptable standards of programs would be a matter for debate within the specialties concerned.

One of the most commonly expressed difficulties in establishing developments of this nature is the availability of suitable software. Most specialists have the knowledge to prepare suitable texts for teaching but do not have the ability to program a computer. A suite of programs has been written to allow personnel who are not experts in computer programming to enter, and perhaps more importantly, to edit teaching and self-assessment texts in a simple manner.

SECTION 3 - CALWRITER SUITE

The suite consists of five programs which call each other as required by the user. When mains power is switched on, the greetings program, HELLO, is automatically run and the user is presented with a menu :-

1. write / edit cal program
2. run teaching program
3. instructions

CALINSTRUCT contains the instructions which explain to the author how to use the suite of programs and these can be displayed if requested. CALWRITE is used to prepare teaching materia, while CALREAD is used to present the teaching material to the student. The two other programs, CALFILE-EDIT and CALFILE-SORT, allow the list of teaching topics to be edited and sorted into alphabetical order.

CALWRITE

This program has been designed to allow the simple preparation of teaching material for use in the Glasgow simulated patient management model. On entry to the program, the user is presented with a menu from which he can select :-

1. enter a new teaching text
2. edit a previously entered teaching text
3. edit the list of text names
4. return to the original menu

If a new teaching text is to be entered, a flag, SELECT$, is set to 'NEW'. If a previous text is to be edited, then this flag is set to 'OLD'. The names of texts which have already been entered into the system are contained in CALFILE and the number of names in CALFILE is obtained from CALFILELEN.

These names are printed out and the user is invited to type the name of the text he wishes to enter or to edit. This name is compared with those obtained from CALFILE and if it is a new entry, CALFILE is re-written with the new name added. The entry of a new program text then continues at the main program routine from line 100.

EDIT PREVIOUS TEXT

If a previous text is to be edited, the data is obtained from disc at line 1100. Data is stored under the name of the text program + 'CAL DATA' and this is loaded in the order :-

1. number of pages - PAGE
2. number of text strings in page(I) - NT%(I)
3. number of question strings in page(I) - NX%(I)
4. the J text strings in page(I) - T$(I,J)
5. topic question strings for each page - G$(I)
6. the I individual question strings in each page - X$(I,J)
7. correct scores for each individual question - B%(I,J)
8. the K feedback strings for each of the J individual question - TE$(I,J,K)

Standard error trap routines are used to detect disc read errors which usually occur when there is no data in the file requested. The user is informed that the text is unavailable and returned to the first menu.

Once the data has been correctly loaded into memory, the EDIT OLD FILE routine presents a list of possible options on the screen.

1. correct / view / print pages
2. add pages to the end of the file
3. insert pages to the file
4. delete pages from the file
5. save file to disc
6. rename the text data
7. return to main menu

1. CORRECT / VIEW / PRINT PAGES

A flag EDIT$ is set to 'Y' if this option is selected and the program passes to the EDITOR SUBROUTINE at line 150. From this subroutine, the user can :-

(a) return to the previous menu if the EDITOR SUBROUTINE was selected inadvertently.
(b) select a page to edit.
(c) print out the text on the screen or a printer using the PRINT SUBROUTINE at line 6900.

When a page is to be edited, the page number selected is checked to determine if it has been completed. If the page has not been completed, an error is generated and the program returns to the previous menu.

When the page number selected for editing is valid, the variables PAGE and the flag array F(I) are stored temporarily and PAGE is set equal to the page number to be edited. X is set to the number of individual questions in the required page and the program passes to the input subroutine at line 200.

TEXT ENTRY

The input subroutine is used for the entry of new text as well as for the editing of previously entered text. The variables which are entered are all indexed by PAGE and this is incremented by the MAIN PROGRAM ROUTINE at line 100 for the entry of a new text. The input subroutine consists of five principal sections:-

1. text frame input
2. topic frame input
3. question frame input
4. answer frame input
5. feedback frame input

1. TEXT FRAME INPUT

A brief list of instructions is displayed which inform the user how to control the input routines (fig. 11). The available free memory can be obtained in case an excessive amount of text is entered. The data is typed in using the INPUT command and the option of automatic word wrap can be selected from the JUSTIFY LINE subroutine at line 8000.

JUSTIFY LINE

This subroutine allows text to be typed into the computer in a continuous block. The block is entered as the text string, AA$ and then scanned. Words which are split over two lines are extracted and recombined into the complete word. For example, the following line could be typed on to the screen :-

The patient was admitted following a roa
d traffic accident.

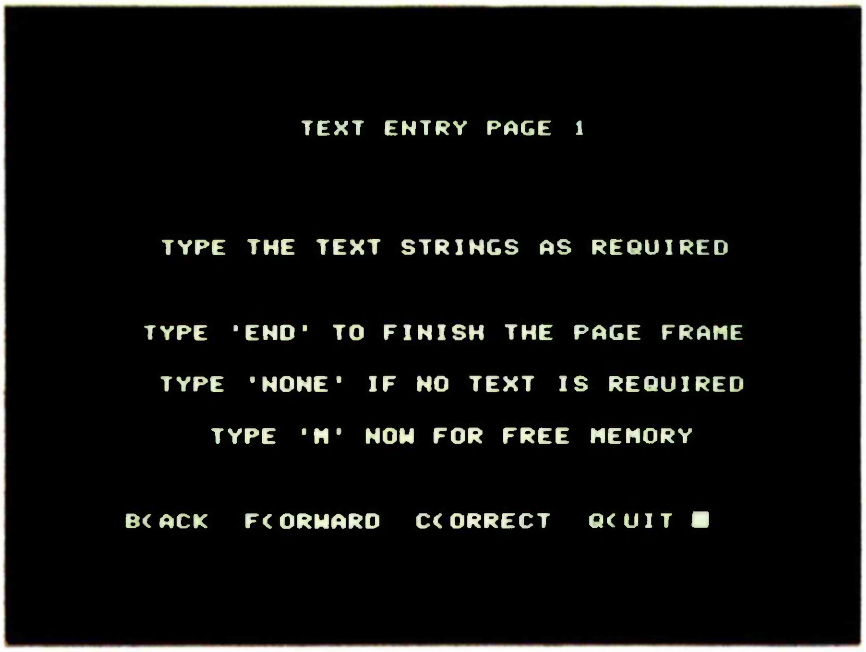

Figure 11 Instructions for controlling text input.

Figure 12 Display of text strings identified by number to allow correction.

This would then be reorganised to :-

> The patient was admitted following a
> road traffic accident.

If the JUSTIFY LINE option is selected, each block of 40 characters in the text string AA$ is checked for the presence of a space at the end. When a space is detected, the portion to the left is retained as the temporary variable A1$ and the rest of the text string is added to it. The text string AA$ is then replaced by A1$ which has the space at the correct position. If a space is not present at the end of a line of 40 characters, the 41st character is checked and if it is a space, the first 40 characters are acceptable.

If the 41st character is not a space, AA$ is scanned backwards from the 40th position to find the first space. A1$ is set to the portion of the text string to the right of the first space detected and AA$ is set to the portion to the left. A1$ now contains the first part of the split word and AA$ contains the first text line without the first part of the split word. AA$ then has the corect number of spaces added to fill up to the 40th character and is recombined with A1$ to check the next block of 40 characters. The 80th position is then checked and so on until the 240th position. The newly formed string is then printed on the screen and control returned to TEXT FRAME INPUT subroutine at the continuation of line 210.

The last text string to be entered is the word 'END' and when this is detected the screen is re-formated to display the text as the student will see it. The maximum number of strings which can be entered is nine and if this is exceeded, the re-formated display is automatically displayed.

A list of the possible options is printed at the bottom line of the screen. These allow movement through all frames of the page being edited. Corrections can also be made by typing 'C' whereupon the text strings are displayed accompanied by their identification number (fig. 12). The number of the string to be corrected is entered and the new string typed in as before. The corrected text page is displayed on the screen along with the control options.

2. TOPIC FRAME INPUT

The topic question is entered and can be corrected if required. The previous text frame can be returned to and further corrections made if necessary or the program can progress to entry of the individual questions using the same command letters as before.

3. QUESTION FRAME INPUT

A maximum of six question strings can be entered. If less than six are required, the word 'END' is typed. The topic question and the individual questions are displayed as the student will see them (fig. 13). Corrections can be made by typing 'C' and the string required for correction is selected by entering the identification number in the same manner as for the text strings.

4. ANSWER FRAME INPUT

Each question string is printed on the screen and the correct score is entered. Correction can be made using the same keys as before (fig. 14).

5. FEEDBACK FRAME INPUT

A maximum of five feedback strings can be entered for each individual question, the final again being the word 'END'. Automatic justification of the lines can also be selected. When the last feedback string has been entered, the user is asked if

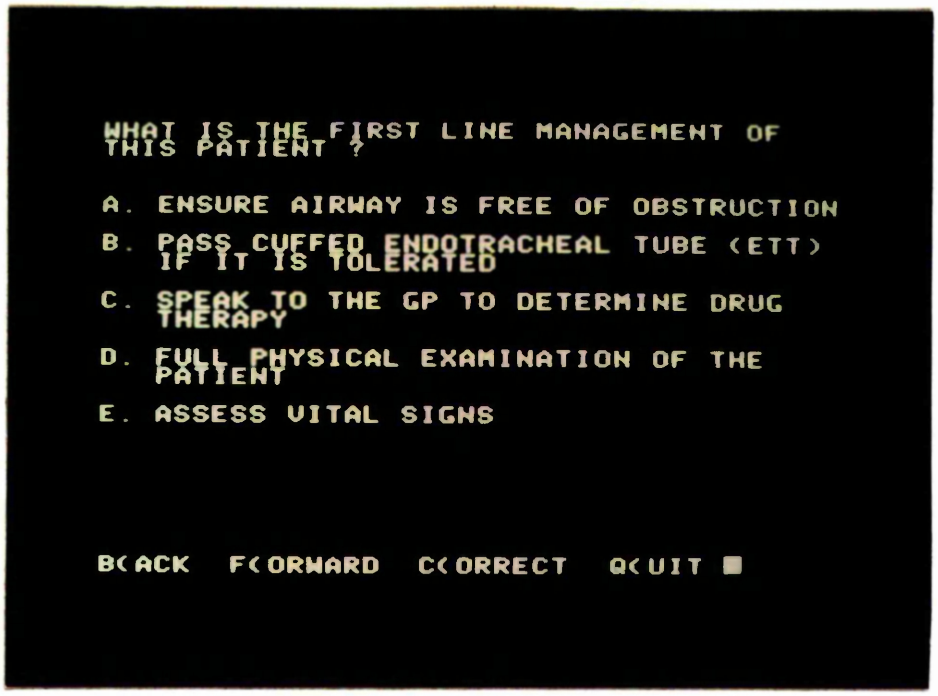

Figure 13 Main topic question and individual questions displayed as they will be presented to the student.

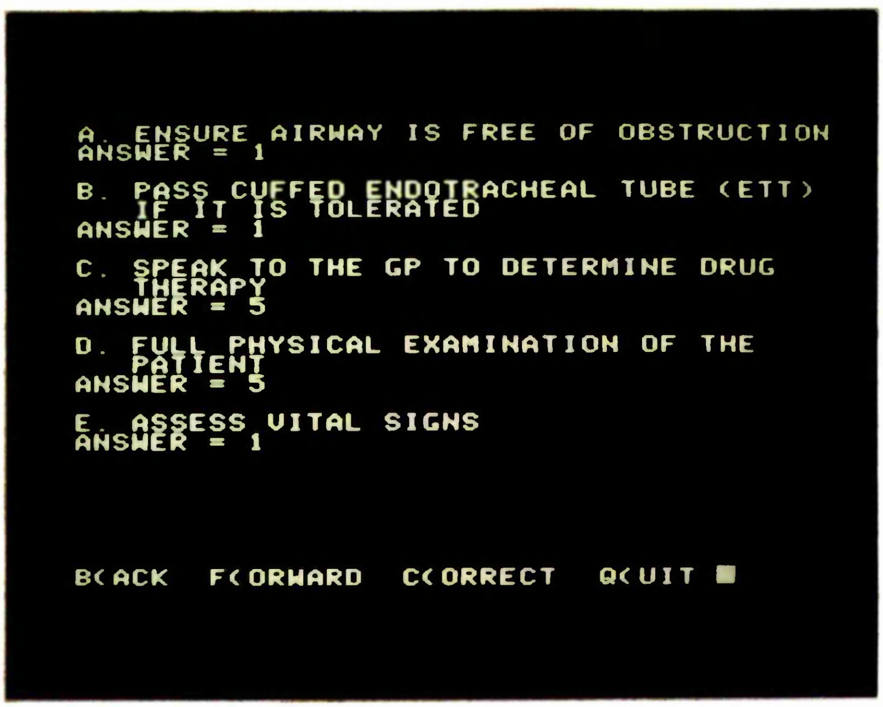

Figure 14 Individual questions and correct scores.

he has completed editing the page. If not, he can return through all the previous frames and make further alterations as necessary. If the text input subroutine has been called from the EDITOR SUBROUTINE, control is returned to the EDIT OLD FILE routine at line 1000 without automatic saving of the amended file.

If the EDIT$ flag is not set to 'Y', a new teaching text has been entered and the entire file is saved to disc since it is assumed that text entry has been completed. The flag for text selection, SELECT$, is then set to 'OLD' and the program directed to the EDIT OLD FILE routine in case editing of some of the pages is required.

PRINT PAGES

The PRINT SUBROUTINE prints out each of the five frames from a selected page number to the end of the file. This can be either on to the screen or to a printer with the interface in slot 1. The user is asked to indicate which option he requires and the number of the start page. When the printer is not used, a reduced print speed can be selected which prints the page numbers and the information which has been entered on the screen at a slower rate than normal (fig. 15). A pause of approximately 3 s is provided between each separate page to permit examination of the screen display.

2. ADD PAGES TO END OF FILE

The EDIT$ flag is set to the null string and the SELECT$ flag to 'NEW'. Control is passed to the MAIN PROGRAM ROUTINE at line 100 and the program continues as if new text was being entered.

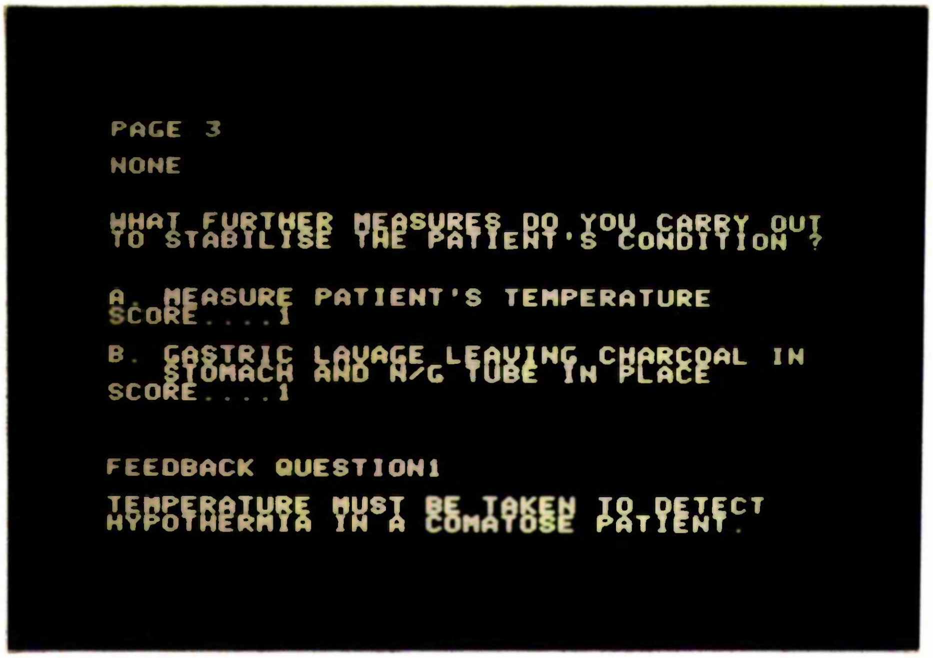

Figure 15 Printout to screen to review information which has been entered.

3. INSERT PAGES TO THE FILE

This subroutine at line 200 verifies that the maximum number of 30 pages is not exceeded and asks for the page number after which the new page should be inserted. An attempt to insert after a page which has not yet been completed is detected and the user informed before being returned to the EDIT OLD FILE subroutine. If these tests are satisfactory, the data is re-arranged with space left for the insertion of the new material. The variable PAGE is set to the new insertion number and the program passes to the text input routine at line 200.

4. DELETE PAGES FROM THE FILE

The page number selected for deletion is checked as for the insertion routine and if satisfactory, the data is re-arranged to delete the required page.

5. SAVE FILE TO DISC

The routine at line 5000 asks the user for the required scoring system and saves the data under the name which was selected at the beginning of the program concatenated with the string '+ CAL DATA'.

6. RENAME TEXT FILE

The standard DOS command RENAME is used to alter the existing name of any text file to the new name required.

7. RETURN TO MAIN MENU

Selecting this option returns the user to the original greetings program from which he can edit names of the teaching texts or use CALREAD to display the teaching material to students.

CALREAD

This program allows presentation to students of the pages which have been compiled using CALWRITE. If a disc was to be used only for students, CALREAD would be the only program necessary and would be run when power was applied to the system.

The first function performed is to read the file CALFILELEN and obtain the number of teaching texts which are currently stored on the disc. CALFILE contains the names of these texts and they are displayed on the screen to allow the student to select the one required.

Once a satisfactory file name has been selected, the teaching material entered by the author is obtained from the disc by the subroutine at line 5000. The data is stored under the file name + 'CAL DATA'. The arrays, X$(X) and B%(X) are used in the main data handling subroutine of CALREAD and therfore, unlike CALWRITE, cannot be used to store the individual questions and the scores.

In CALREAD, the individual question strings are stored in the array, MX$(X) and the correct scores in the array MB%(X). Otherwise, the variables used are the same as for the CALWRITE program. The variable EX is set to 1 if the author requires that only exactly correct answers are to be accepted. If it is set to 0, then answers which are incorrect by only one point are scored as 1 while those which are exactly correct are scored as 2.

MAIN PROGRAM

The main program begins at line 100 and presents the relevant scoring system which has been selected by the author with examples of how the scoring systems are applied. If SCOAR = 3, then the 1-3 scoring system is used. When SCOAR = 5, the 1-5 system is displayed. An alternative system of question handling was requested by one of the staff members at Duke University who prepared a teaching text on neuromuscular blockers. This allows the students to select only one from a list of possible options in the same manner as the first question presented in the program CALOB-TEXT. EX is set to 1 so that only exactly correct answers are accepted and the 1-3 or 1-5 scoring systems are not displayed.

PRINT TEXT PAGE

MPAGE is incremented each time a new page has been presented to the student and is used to indicate how many text frames should be displayed to the student if he requests to recall the previous text at any time during the use of the CAL unit. The variable NP in the FOR-NEXT loop at line 110 increments the pages for entry to the subroutine PRINT TEXT PAGE. This subroutine prints the relevant text frame for each page and selects the topic question, individual questions, correct scores and feedback which will be passed to the TEXT/QUESTION HANDLER subroutine at line 30000. These are indexed by the variable NP but if the topic question string is 'NONE', then the author does not require questions at this part of the teaching presentation, but wishes to display a further text frame.

Once all the pages have been presented to the student, the final score is calculated with the routine CALCULATE SCORE at line 6000 and the scores for each individual question together with the final score are stored on the disc at the WRITE SCORES TO DISC routine at line 19000. The scores are stored in a file indexed by the name of the CAL teaching text, for example, 'BURNS SCORE'. A SCORE COUNT file is used to retain the number of entries and the scores are stored as a single string in the SCORE file for that CAL text.

TEXT/QUESTION HANDLER

This subroutine is based on the previous subroutine used for programs such as CALOB-TEXT and the variables required are obtained in the correct form from the PRINT TEXT PAGE subroutine described above. The main topic question is displayed along with the individual questions. These had originally been numbered but this had occasionally caused confusion since the scoring system, recall of previous text, and the option to change answers were also numbered. CALWRITE, therefore, automatically inserts letters from 'A' onwards to identify each individual question.

The student is asked to score each question and the letter used to identify each question is obtained from the expression CHR$ (64 + N) where N is the number of the question. The student's response is obtained as before, using the GET command and checks are made to ensure that it is within the acceptable range. The answer is printed on the screen and he has the option to change it if required. The student's score is compared with the correct score and, when EX is set to zero, he is awarded two points when completely correct and one point for an answer which is wrong by only one . If the author has set EX to 1, only exactly correct answers receive points.

The feedback strings for the question which was answered are then displayed and the total score incremented as necessary. The variable TC is used as a counter to record the total number of questions to allow the calculation of the overall percentage score at the end of the session. The variable N is compared with the total number of individual questions, J, and if equal, control is returned to the subroutine PRINT TEXT PAGE to continue with the next page. If N is less than J, N is incremented and the student progresses to the next question.

DISPLAY PREVIOUS TEXT

Each time the student completes an entire page, MP is incremented. This subroutine is called if the student wishes to recall the text frames which have been displayed up to that point. Each text frame is printed in turn on the screen, and remains until the student types '1' to continue to the next text frame.

The considerable development time required to enter teaching material into the computer was a major difficulty of the previous programs where the text, questions, scores and feedback were all inorporated into the BASIC program. Editing also involved considerable time, principally because of the effort to format the text displayed on the screen. The use of CALWRITE has greatly decreased the programming and editing time required and allows staff who are relatively untrained in the use of computers to enter suitably prepared teaching material.

One of the principal difficulties in establishing a CAL system has always been the availability of suitable teaching material. Authors of teaching programs are usually willing to share their teaching material (92) and, even when translated into a different language, this transfer can be extremely beneficial. Voss, Weienhammer and Renschler (91) reported the successful transfer of teaching material for teaching German medical students. Programs such as the 'MAC' series of dynamic models and others written in COURSEWRITER III and PILOT were successfully implemented.

The acceptability of the case study model to both undergraduates and qualified anaesthetists would suggest that if suitable texts can be prepared and distributed, this difficulty could be overcome. The development suite should enable other workers in this field to prepare CAL units for other departments to use or to alter easily to their own requirements those which are already available.

CHAPTER 3

MICROCOMPUTERS IN THE INTENSIVE CARE UNIT AND OPERATING THEATRES

DATA ANALYSIS

SECTION 1

MICROCOMPUTERS IN THE INTENSIVE CARE UNIT AND OPERATING THEATRES

INTRODUCTION

The application of computing technology in the intensive care unit and operating theatre can only take place if the computer system is demonstrated to the medical and nursing staff to fulfil certain criteria. It must either perform a particular function better or more easily than the method normally used or it must be able to undertake tasks of benefit to the patient but which are not normally able to be undertaken by the medical or nursing staff (42).

Hilbermann and colleagues (31) reported the evaluation of a computer-based monitoring system in use at the Pacific Medical Center. They emphasised the difficulty of objectively evaluating the benefit of the system but considered that changes were detected in patients by the computer which were not detected by standard monitoring and observations. They found that the nursing staff were the principal users and that the use for research purposes was less than for clinical management of patients.

One important test of any computer system is the ease of use by the nursing and medical staff. It is imperative that as many errors of data entry as possible are detected and that the inevitable errors which will occur can be easily corrected. There will always be new users of any system because of the normal changeover in nursing and medical staff and it is valuable to have instructions available to them, built into the programs which they will use. Screen displays should be designed to provide instructions which enable the user to control the flow of the program easily. The control routines should be arranged so that where possible, only single keystrokes are required and each character typed is validated for correct entry.

The development of microprocessor based systems has allowed computers to be used at the bedside in the intensive care unit and also in operating theatres with minimum disruption to routine work because of their small size. The use of computer games and the CAL programs described previously have proved valuable in dispelling fears of the use of computers by non-technical staff.

During the development phase of a complex suite of programs, it is essential that the co-operation of the medical and, perhaps more importantly, the nursing staff is obtained so that program errors can be detected and satisfactorily corrected. One of the advantages of a computer system is that it can prompt users to enter the required data by printing suitable requests on the display screen. Frequently, it is the nursing staff working in the intensive unit who will enter this information and they must be made aware of the potential benefits for their patients which should result from this.

If data which nurses would normally have to measure for themselves can be collected by the computer on-line and made available to them easily, they will have more time to spend nursing the patient. Conversely, if the computer involves a significant increase in workload without any compensatory benefit, the attempt to introduce computing technology will fail.

ON-LINE CARDIORESPIRATORY DATA COLLECTION

Severely ill patients require a considerable number of therapeutic interventions to optimise their chances of survival. To permit this optimisation, the variables measured must be collected frequently and data derived from these variables made easily available. The ultimate requirement is to provide an adequate supply of nutrients to the tissues and to remove the waste products of metabolism. Therefore, the cardiovascular and respiratory systems are those which are of principal interest.

Therapeutic procedures which alter one of these systems frequently cause some effect on the other, which may be detrimental to the overall management of the patient. The use of inotropic drugs such as dopamine to improve cardiac output and tissue perfusion may increase the degree of intrapulmonary shunting (33) or may have no effect (29). Vasodilators can also be of benefit in increasing tissue perfusion by decreasing afterload and improving cardiac output. The effect on intrapulmonary shunting may be of no significance (84) or increased venous admixture can result (15). Changes in cardiac output can also cause alterations in the degree of intrapulmonary shunting (49).

The use of intermittent positive pressure ventilation in the treatment of acute respiratory failure may improve arterial oxygenation but can decrease cardiac output. This can be a major disadvantage of positive end expiratory pressure (PEEP) which, although it can dramatically improve arterial PO2, may alter cardiac output significantly.

The concept of optimum PEEP was introduced by Suter, Fairley and Isenberg (85). This attempted to minimise the deleterious effects of PEEP by measuring the oxygen availability and selecting the value of PEEP which resulted in the maximum delivery of oxygen to the tissues. However, Gallagher, Civetta and Kirby (25) considered that the aim of therapy for patients with adult respiratory distress syndrome should be to set PEEP to reduce the intrapulmonary shunt to 15% and support the cardiovascular and renal systems by appropriate means.

The overall effects of such forms of therapy on an individual patient are not predictable and it is, therefore, essential to undertake the required measurements and calculations to follow the effects of therapy (39). Simple programmable calculators can be used for this function (36, 77), but they have a limited memory capacity and cannot automatically begin to execute a program. Nor can they readily prompt the user with a selection menu to provide instructions on the possible options offered.

The measuring system must be organised so that the information is made available with the minimum of effort to the medical and nursing staff so that they can direct their attentions to the patient (9, 57). A survey at the Westminster Hospital (52) showed that using the computer for on-line collection of cardiorespiratory measurements and manually entering other data, 7% of the nurses' time was spent in clerical duties compared with 19% when standard charts were used. Sheppard (80) reported that the use of a computer-based monitoring system increased the time spent on direct patient care by the nurses from 2% to 24%.

Microcomputers have the ability to collect data automatically, to calculate the required derived data and to present this information to the clinician as required (80). The maximum amount of data can thus be obtained from any form of invasive monitoring. Long term storage can be achieved to permit subsequent analysis of what may be large quantities of data. This is an extremely important feature which is omitted from some of the more sophisticated and expensive monitoring equipment (19).

The computing system which has been developed for use in the Respiratory Intensive Care Unit and operating theatres at Glasgow Royal Infirmary is again based on the Apple II microcomputer (fig. 16; 40). The system has been designed to use standard readily available components and to perform as many tasks as possible in software. Expert technical assistance can be difficult to obtain and the more tasks which can be undertaken in software, the less technical help is required.

An important design feature of all the programs produced has been ease of use by the medical and nursing staff. Various methods have been attempted to improve the interface between the user and the computer. Bailey, Wyatt and Allwood (5) have used a digitising bit pad to allow the nurses to enter fluid data simply. Sarnat, Quinn and Ty Smith (74) have reported the use of a voice recognition unit for data entry in the operating theatre with some success.

One simple method of facilitating the use of the computer has been the requirement to use as few keystrokes as possible in directing the flow of the programs. The Apple II has the GET command available in BASIC which, as explained previously, obtains a single character from the keyboard so that the need to type the RETURN key is eliminated. This command is used to obtain a single character string which can then be examined in the program and rejected if outwith an acceptable range. Where possible, errors of data entry are detected and corrections simply made.

Instructions are incorporated into the program suite to provide an outline of the system and its capabilities and in particular to allay the fear of causing loss or alteration of data stored in the computer. The overall organisation of the program suite is shown in figure 17 and the individual programs are described below. The system is organised so that when power is applied, the computer automatically starts the disc drive and loads the greetings program, HELLO. Systems have been developed using magnetic tape for storage (68), but discs have the

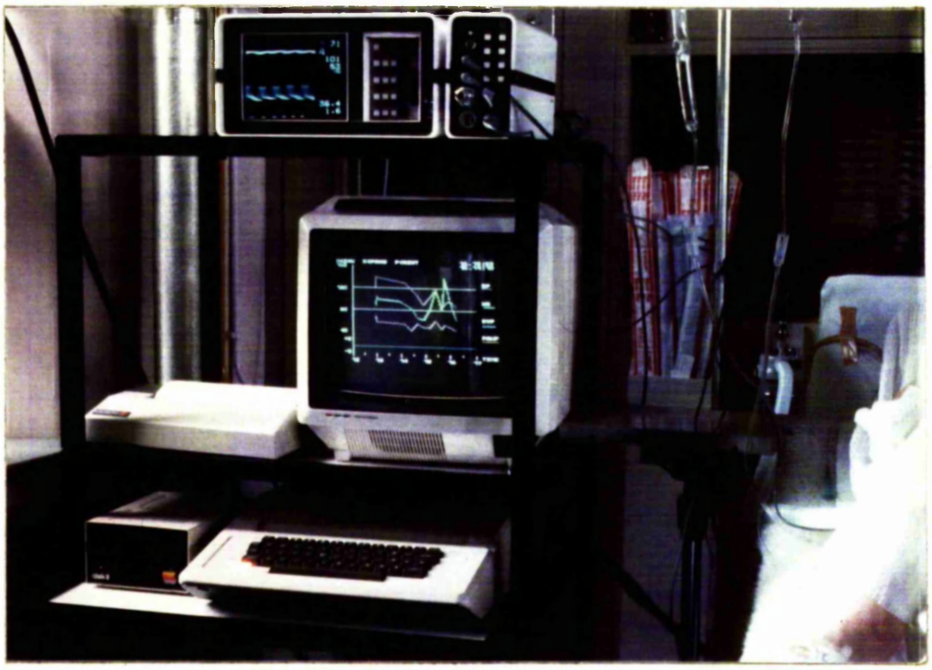

Figure 16 Apple II computer monitoring system used in the Respiratory Intensive Care Unit, Glasgow Royal Infirmary.

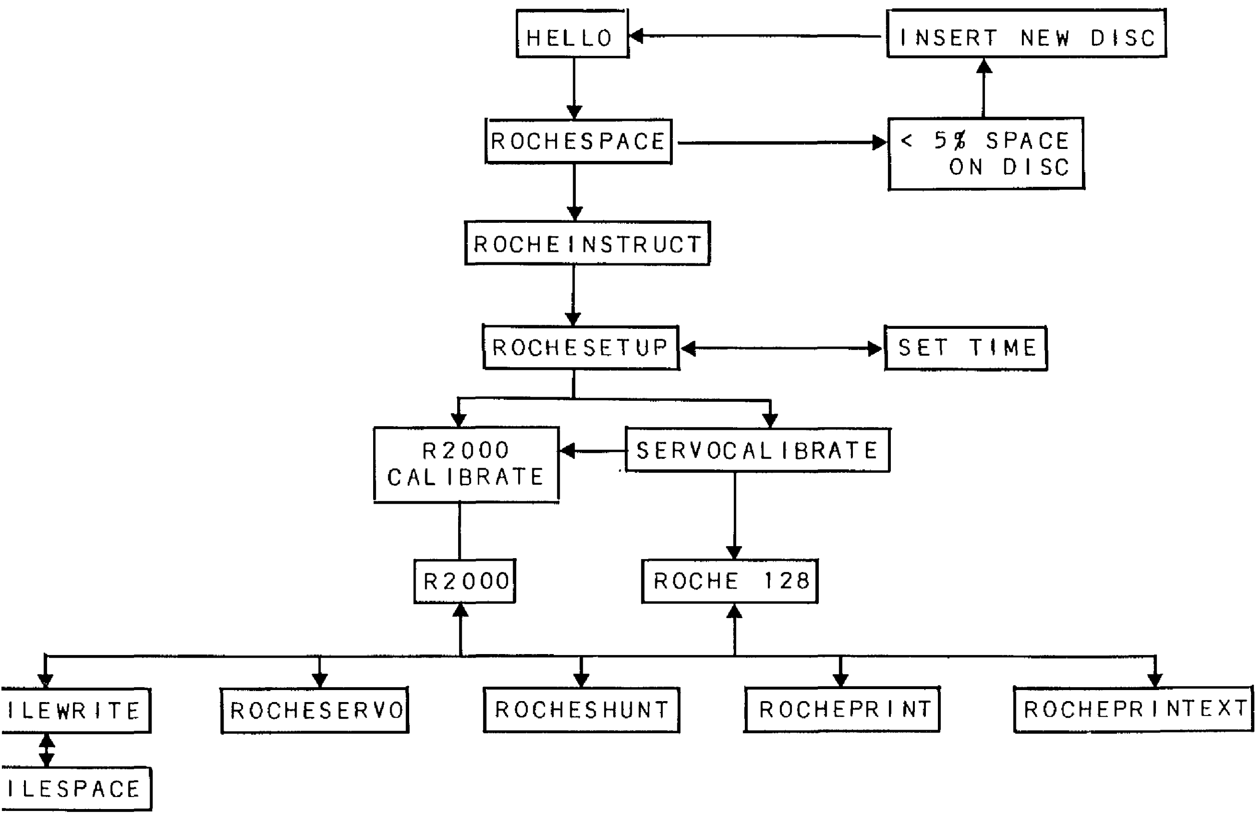

Figure 17 Organisation of the ROCHE128 suite of programs.

advantage, as explained earlier, of rapid access to any part of the stored information.

ROCHE SUITE

HELLO

Two principal multichannel cardiovascular monitors are in use in the Respiratory Intensive Care Unit and operating theatres in Glasgow Royal Infirmary; the Roche 128 and the Roche 2000 systems. Different programs are employed to collect data from these monitors and the user is asked to indicate which program is required. A file called NEXTPROG is then opened and the string 'ROCHE128' or 'R2000' is written into this file depending on the monitor selected.

NEXTPROG is a file which is used by several of the programs in the Roche suite to store the name of a program which will be called subsequently. It is frequently used when one program calls another. Once the second program has accomplished its task, it can then read the name of the calling program from the file NEXTPROG and re-run the original program. Thus, the user is unaware of the individual programs used. The HELLO program then calls ROCHESPACE.

ROCHESPACE

This is a modification of a routine provided on the Computec Utilities disc which measures free space available on the magnetic disc. If the space is less than 28 sectors or 5% of the total space, the disc is declared to be full since this repesents approximately the space required to store one complete collection of patient files from the memory. When less than 28 sectors are available, the program is halted and the user informed that another disc is required. Once the new disc has been inserted, the RETURN key is typed and the HELLO program is run once more. If there is adequate space available on the disc, ROCHESPACE calls the program ROCHEINSTRUCT.

ROCHEINSTRUCT

This program provides the user with the option of obtaining instructions for the system or passing directly on to the next program. When the instruction option is selected, an explanation is given of the signals which are collected and of the method for controlling the programs (fig 18). The progam reads the file NEXTPROG and displays the instructions appropriate for the Roche 128 or Roche 2000 cardiovascular monitor which has been selected. It also provides information about the various additional interface cards required and the positions in which these are placed within the computer (fig. 19).

ROCHESETUP

If the instruction option is not selected, ROCHESETUP is run. The first operation of this program is to obtain the current date and time on the Mountain Hardware real time clock. This is used in the Roche programs to label the data with the date and time of collection and has a battery back-up supply which provides power for the clock for four days if the computer is disconnected from the mains supply. To maintain the DOS pointers intact, the character 'CTRL D' has to precede the instructions to read the clock. The date and time are then displayed and the choice of altering them is offered using the GET command.

Program SET TIME is run if the decision is made to correct the readings from the clock. This program is based on the routines supplied with the Mountain Hardware clock and has been modified to improve the ease of use and trapping of errors.

Before SET TIME is called, the program name ROCHESETUP is written into NEXTPROG. However, since NEXTPROG already contains the principal cardiovascular program to be run, either ROCHE 128 or R2000, ROCHESETUP is the second entry in this file. Once the date and time have been corrected, ROCHESETUP is re-run and the

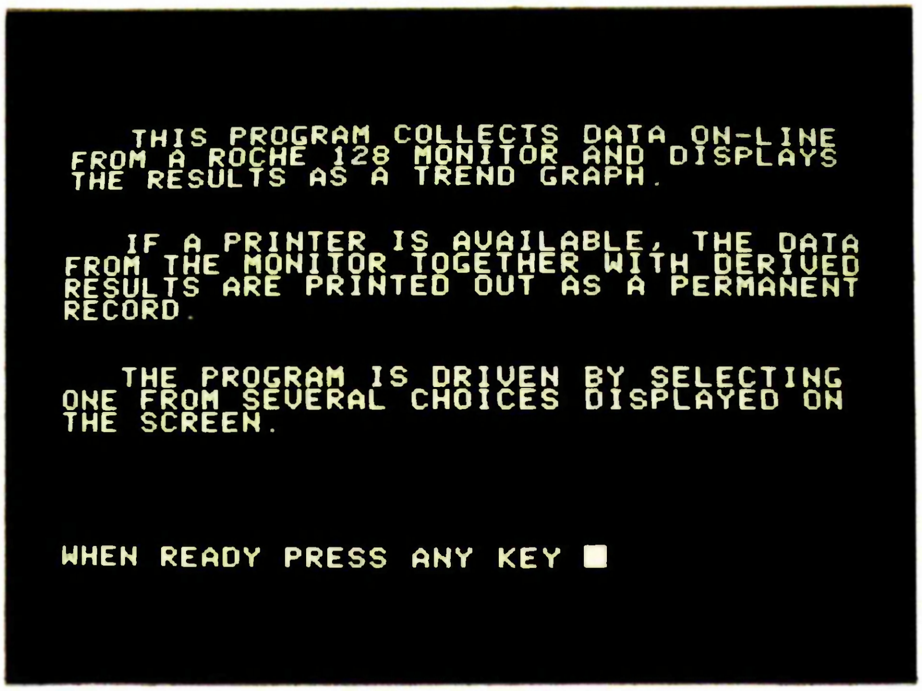

Figure 18 Explanation of signals collected by the system and the method of controlling the program.

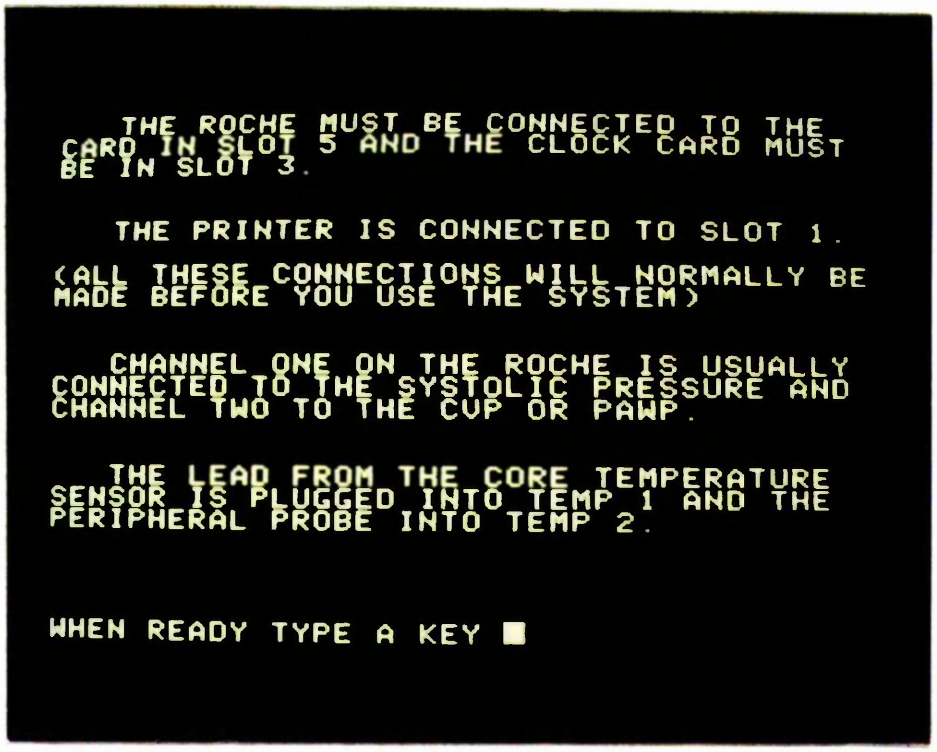

Figure 19 Instructions for insertion of interface cards and connection of patient signals.

new date and time are displayed.

If the date and time are correct, a file called NAMEFILE is read and the name of the last patient connected to the system is obtained and displayed on the screen. If this is the name of the patient to whom the system is attached, the user can progress to the next section. If the name has to be changed, the program passes to the ADDNAME subroutine. When the NAMEFILE is read, an error is generated in the system if the file contains no data. The ON ERR instruction is used to detect an empty namefile and direct the program immediately to the ADDNAME subroutine.

ADDNAME SUBROUTINE

This subroutine requires the patient name, hospital number and body weight to be entered using the INPUT command and instructions are given to explain the necessary format. Checks are made to ensure that the name contains no numerals and that one space exists between the surname and the initial of the forename. The entry of more than 20 characters for the entire name generates an error and the program returns, after a suitable prompt, to the input routine. Once a satisfactory entry has been made for the patient name, blank spaces are added to the name to produce a constant record length of 20 characters.

The hospital number and body weight are then entered and scanned to check that no letters have been entered and that the correct number of numerals have been typed. If any of these checks reveal an error, the program is returned to the relevant input routine for re-entry of the data. Once the data have been satisfactorily entered, they are stored in NAMEFILE for later use by other programs in the suite.

The user is then asked if he wishes to calibrate the Seimens Servo ventilator output when this is connected to the computer. If this is required, the program SERVOCALIBRATE is run. If calibration of the ventilator is not required, NEXTPROG is read and if it contains ROCHE128, this program is loaded and run. If NEXTPROG contains R2000, a preliminary calibration program, R2000CALIBRATE is called since offset and calibration factors are required for the R2000 system to convert the digital values obtained from the analogue to digital converter (ADC) into pressure and rate measurements. This calibration program is discussed later.

One of the principle features of the Apple II is the provision of high-resolution colour graphics. However, the memory used for this occupies a block from 16k-24k. The machine code programs used to generate text on the graphic page are stored above this block and all programs which use the graphic facility must be stored below the graphics page and therefore cannot exceed 16k in length.

It is not possible, therefore, to have all the facilities required stored in a single program and each program must be able to call others in the suite and be in its turn recalled. Overall organisation of the two main programs, Roche 128 and R2000, is shown in figure 20. R2000 was developed from ROCHE 128 and is identical in most respects. Where there are differences between the two programs, these will be pointed out during the description of ROCHE 128 but will be explained in detail under R2000.

ROCHE 128

The first series of instructions in ROCHE 128 detect if the program is returning from calling another or whether it is being run for the first time since start up. When ROCHE 128 calls one of the other programs, a file is written called SHUNTFILE in which flags and constants used in ROCHE 128 are stored.

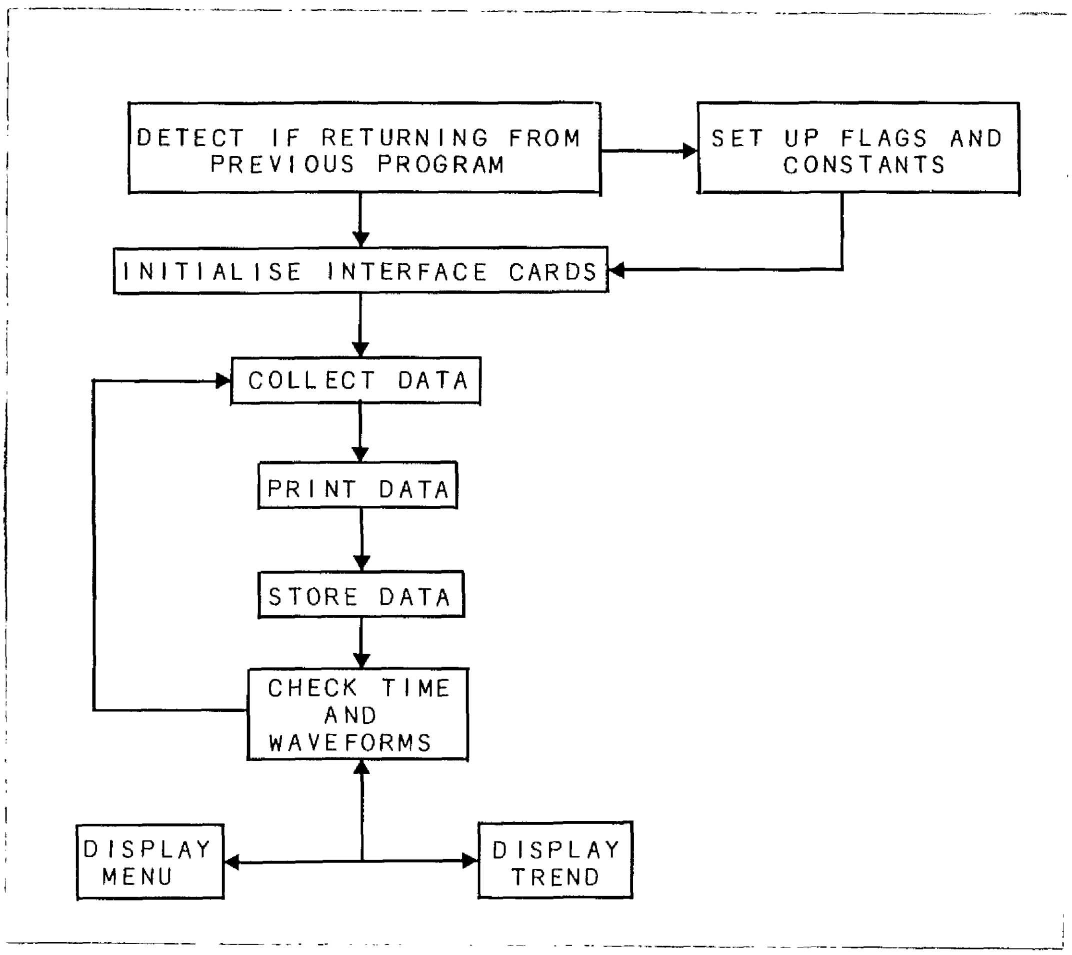

Figure 20 Organisation of the ROCHE128 and R2000 cardiovascular data collection programs.

SHUNTFILE contains :-

1. data retrieval time - DR (default = 30 min).

2. printer connected flag - PR (1 = printer in use, 0 = not in use).

3. pulmonary artery catheter inserted flag - (CVP$ = 'N' if inserted).

4. FLAG$ = 'Y' indicates that the patient identification and data headings have already been printed out and do not require to be reprinted.

5. channel used for systemic arterial pressure - (PT$ = 'Y' if the systemic pressure signal is connected to channel one). In R2000, this flag is not used and IPPV$ is stored in its place. It is set to 'Y' to indicate that the patient is receiving intermittent positive pressure ventilation (IPPV).

If ROCHE 128 is being recalled, SHUNTFILE will contain the flags and constants and these are entered into the computer memory. The program then loads the machine code program routines which enable the printing of text on the high-resolution graphics page and the machine code routines necessary for data collection. If no data are found in SHUNTFILE, an error is detected and the system prompts the user to enter information concerning the signals available from the Roche 128 monitor, whether hard copy output is required and whether a Seimens ventilator is connected to the computer.

DATA COLLECTION

The Roche 128 monitor is itself a microprocessor based device with inputs for e.c.g., two pressure channels and two temperatures. The information produced by the Roche 128 can be transmitted to a remote computer via an optional communication interface which is fitted within the monitor. A California Computer Supplies (CCS) serial card is fitted in slot five of the Apple after the standard interface ROM has been removed from the card and replaced with a RAM integrated circuit.

LOAD / CARD

Two machine code programs were developed in conjunction with Mr Ian Boyle of Strathand to allow the Apple to communicate with the Roche 128 and these are loaded before data collection procedes. Program CARD is loaded into the RAM on the CCS Serial card and permits data transfer between the Roche 128 and the computer. Program LOAD organises the request for data and the collection of the individual characters. It is considered in detail after the BASIC programs in this suite have been described.

When data is required from the Roche 128, the machine code program LOAD is called. The monitor requires a hexadecimal 05 character to initiate data transmission and one of the features of the program CARD is to remove bit 8 which is always set high by the computer. Hexadecimal 05 is transmitted to the Roche 128 which returns a character string containing the cardiovascular and temperature data together with any error messages which indicate faults in the system such as incorrect zeroing of the transducers.

The data obtained from the monitor is stored in memory by LOAD from decimal location 896 for subsequent analysis in BASIC. Occasionally, the monitor cannot respond to the interrupt request signalled by the computer or it may have become disconnected. A timeout delay has therefore been incorporated into LOAD so that if no character is received by the computer within 100 ms, 01 is stored in location 880 and control is returned to BASIC where the contents of location 880 are examined and visual and audible warnings given.

The subroutine at line 3000 of Roche128 is used to obtain the variables stored in memory by the machine code program LOAD. Each measurement obtained from the monitor is stored in three locations and the start location is indexed by the variable I which is given the appropriate value before entering the subroutine. Pressure measurements must allow for negative numbers and the true value is calculated by subtracting 128 and dividing the result by 2. This calculation is only performed if a pressure measurement is being obtained and this is indicated by setting the flag PFLAG$ to 'Y'. Otherwise, the actual figures located in memory are returned as the true value. If no transducer is connected to a pressure channel the value -64 is obtained and for the purposes of printing and storage, this is converted to the string '---'.

PRINTER DISPLAY

If the user has informed the computer that a printer is available, it is assumed that this has been placed in slot 1 and output is formatted for an 80 column display. Headings are printed in columns to label the subsequent information if the system has obtained the first set of results; otherwise the headings are omitted. Figure 21 shoes the printout produced by the Silentype thermal printer used with the computer system.

Every result is printed before storage on to the disc so that in the event of a system failure, no information is lost. Similarly, when the system is started, it automatically searches for any previous data for the patient and stores this in memory and further information is added to that which was acquired before. The system can therefore be used to collect data in the operating theatre, switched off and moved to the ward or intensive care unit and re-started with all the data preserved intact.

BOYLE V   002434   10/6/1982

	SYSTEMIC DATA					CARDIAC DATA			PULMONARY DATA				TEMP	
TIME	HR	SYS/DIA	MEA	CVP	SVR	C.O.	SVL	RPP	SYS/DIA	MEA	PAWP	PVR	CORE	DIF
16.22	142	101/ 55	68	---	----	----	---	143	39/ 16	26	---	----	38.1	3.5
16.37	135	102/ 41	67	---	----	----	---	137	37/ 16	25	---	----	38.0	3.5
16.54	139	107/ 42	67	---	----	----	---	148	39/ 17	26	---	----	38.0	3.6
16.55	139	107/ 42	65	---	----	----	---	148	37/ 16	24	13	----	38.0	3.6
17.10	137	110/ 44	63	---	----	----	---	150	37/ 14	23	---	----	37.9	3.6
17.17	136	111/ 44	61	---	10.7	5.7	---	150	23/ 9	18	10	1.40	37.9	3.6

TIME	QS/QT	A-V	O2CONS	O2DEL	PEEP
17.17	21	7.5	428	715	0

17.21	136	112/ 44	61	---	----	----	---	152	35/ 15	23	---	----	37.9	3.6
17.28	135	110/ 45	61	---	10.7	5.7	---	148	21/ 10	16	10	1.05	37.9	3.9

TIME	QS/QT	A-V	O2CONS	O2DEL	PEEP
17.28	24	7.5	428	715	0

Figure 21 Sample of printout produced by Silentype printer used in intensive care unit and operating theatre.

FILE STORAGE

Sequential files are used to store the data since the records may be of different lengths and considerable storage space would be wasted if random access files were used. Each file is identified by the patient's name and system, for example, Smith J CVS DATA and has associated with it a file containing the number of records stored for that patient, Smith J CVS DATALEN.

Several methods of storing data were examined. The principal problem was the relatively slow speed of data acquisition from the floppy disc when each value was followed by a RETURN character. This was eventually solved by concatenating all data collected for one date/time period into one string of characters with EOF added to indicate the end of the data block. When these data strings are loaded into the computer, each is placed in a buffer and only allocated to a variable by BASIC when the RETURN character is received.

The DATALEN file used to store the total number of records for each patient is read when the main cardiovascular program is run and the number stored as variable FI. Each time a block of data has been collected, FI is incremented and the new value is written into the DATALEN file. The block of data is then appended to the patient data file at position FI-1 since the first entry of data is at position zero. The APPEND command did not function correctly under DOS 3.2 and the POSITION command was used instead to add further information to the data file.

Each time 150 records have been collected, the first 50 are written into a separate file by running the program FILEWRITE. The flags and constants are first stored in SHUNTFILE so that on returning from FILEWRITE, ROCHE 128 or R2000 can immediately progress to data collection. Before FILEWRITE obtains the 150 patient data records, the remaining space on the magnetic disc is measured to ensure that adequate storage space is available. A separate program FILESPACE is run which measures the free space and stores a flag, XX$ in a disc file. The string variable XX$

is used to indicate whether or not adequate space remains on the disc.

FILEWRITE is then re-called, obtains the 150 patient records and the flag XX$. The flag XX$ is examined and if enough space is available, the first 50 blocks of patient data are stored in a file which is labeled with the patient name + 'CVS 50'. Each subsequent block of 50 records which is required to be stored is then labeled as patient name + 'CVS' followed by increments of 50, for example, Smith J CVS 50; Smith J CVS 100; Smith J CVS 150. The number with which the block of 50 records is named is obtained by first reading the patient file labeled with 50. If data has been placed in this file, no read error is detected. The file labeled with 100 is then read and again, if no read error is detected, data is contained in this file. Once a read error has been obtained, this indicates that that number has not yet been used to label a data file and the block of 50 records is written using the last number to label it.

If inadequate space is available on the original disc, the name of the calling cardiovacular program is obtained from NEXTPROG and the user is instructed to insert a new disc which becomes the new system disc. All the cardiovascular data is then written on to the new disc, the identification details of the patient connected to the system written into NAMEFILE and the calling program is run.

TIMING SUBROUTINE

After the data has been printed and stored, ROCHE128 passes to a wait stage. Here, the clock is examined and the elapsed time calculated. If the elapsed time is greater than the time interval selected for data retrieval, a further set of data is collected. The data retrieval time interval defaults to 30min on start-up but can be altered at any time to within the range 1-120min. If it is not yet time for further data collection, the program passes to the TEXT MENU routine at line 20000 which displays the principal selection menu.

TEXT MENU

This menu indicates the possible options which can be used to direct the flow of the program and again, selection is made by a single keystroke (fig. 22). The menus for the two cardiovascular programs have different features and those applying to the R2000 program are described later. The GET command is not used on this occasion since the processor would wait until a key was struck before continuing and data could not be acquired during this time.

The Apple II has a keyboard buffer into which is placed the ASC II value of the last key pressed. The content of this keyboard buffer is obtained from the routine TEST KEYPRESS at line 21100, stored as the variable A and the buffer immediately cleared for re-use. The ASC II value is then tested from lines 20105 to 20152 of the menu routine to check if it is valid, and if so, the new instruction is followed; otherwise, the program procedes to check the systolic and pulmonary pressures.

ALARMS

If no valid key has been pressed, the systemic arterial pressure is examined to check for damping or disconnection of the arterial line. Audible and visual alarms are set if the pulse pressure is less than 20% of the mean pressure. This was found to be more appropriate than the value of 33% recommended by Raison and colleagues (67). The user is asked to indicate during the start-up procedure if a pulmonary artery catheter has been inserted in which case CVP$ is set to 'N'. If CVP$ = 'Y', the program returns to re-examine the elapsed time. However, if a pulmonary artery catheter has been inserted and CVP$ = "N", the pulmonary arterial pulse pressure is also examined to detect damping of the pulmonary artery waveform which could indicate

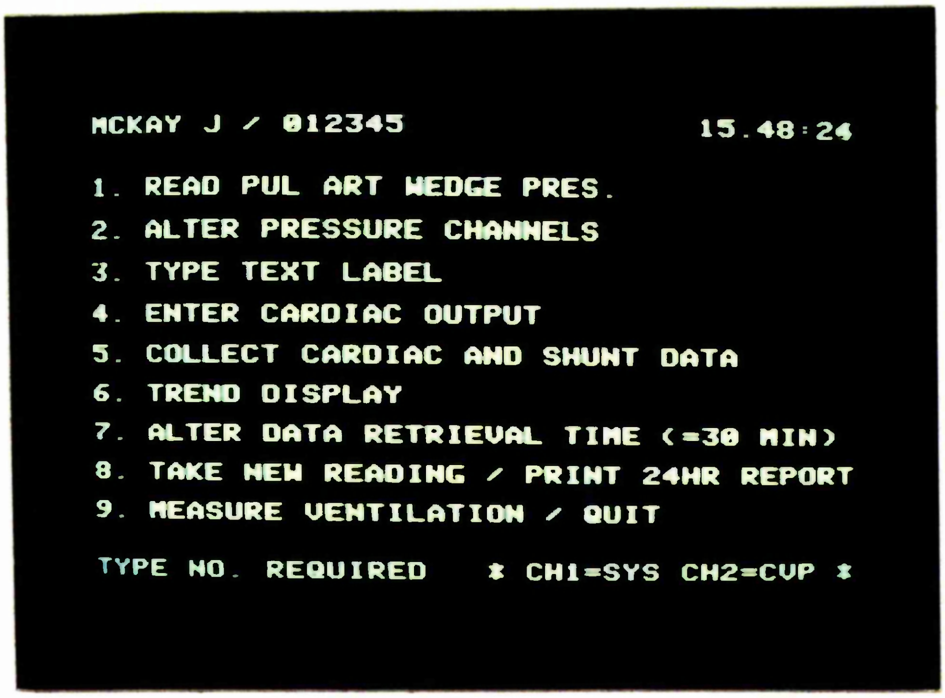

Figure 22 Text menu displayed on screen.

spontaneous wedging of the catheter in the pulmonary artery.

The standard monitor produces values averaged over 5 s but the program ROM in the Roche 128 has been specially modified by the manufacturers to allow beat by beat measurement of the pulmonary artery pressure to minimise the effect of respiratory pressure changes. The values obtained for the systolic, mean and diastolic pressures therefore refer to the same beat rather than the peak, mean and trough values obtained over 5 s as is the case with the standard Roche 128 monitor.

The subroutine for the text menu is located at line 20000 and offers the following options :-

1. enter pulmonary artery wedge pressure
2. alter pressure channels
3. enter text comment
4. enter cardiac output
5. collect cardiac and shunt data
6. trend plot
7. alter data retrieval time
8. take new reading / print 24 h report
9. measure ventilation / quit

One difficulty which occurred during development was the selection of an incorrect number from the menu. All selections therefore inform the user that the request can be cancelled simply by typing the RETURN key and they will be returned to the text menu.

1.

The suite of programs evolved out of a need to collate the large amount of information which was made available from the use of pulmonary artery catheters and the first selection allows the user to enter the pulmonary artery wedge pressure (PAWP) into the computer. The program then prompts the user to enter the CVP to allow a comparison between the measurements to be made. A possible future developments would be to automatically inflate the catheter balloon and obtain the wedge pressure without operator intervention.

2.

When this option is selected, a sub-menu is displayed which presents the following choices :-

(a) The default channel for the systemic pressure is channel one but this can be altered to channel 2 if required.
(b) If the use of the second channel is altered from recording PAWP to CVP, then this also can be indicated to the computer.
(c) The program to calibrate the outputs from the Seimens Servo 900B ventilator can be selected and will be loaded and run automatically.

When the text menu is displayed, the pressures which have been allocated to the two channels are indicated at the bottom of the screen display. This indicator flashes every two seconds as the pressures are examined and indicates that the program is functioning satisfactorily.

3.

Text comments can be entered by selecting option 3. These are printed out along with the time and also stored in a text file for later recall. This feature was suggested by the intensive care nursing staff who use it to record nursing details which can be printed out at the end of their shift and provide a summary for their colleagues who take over the care of the patient.

4.

If cardiac output is available, it can be entered along with the CVP and PAWP by selecting option 4. Cardiac output is measured using an IL 601 analogue cardiac output computer, but work is in progress to develop a suitable Wheatstone Bridge circuit which will enable the Apple II to integrate the area under the thermodilution curve and calculate cardiac output. Stroke volume, systemic and pulmonary vascular resistances are calculated , printed and stored on disc.

5.

Samples of systemic arterial and mixed venous blood can be withdrawn and the cardiovascular information necessary to calculate various derived data are collected by the system. These are stored on disc to allow the final calculations to be made when the results of the blood gas analyses are available. The program ROCHESHUNT is then loaded and run, the cardiovascular variables collected from disc, and the calculations made. ROCHESHUNT is described later.

6.

A trend facility has been incorporated into the program to enable rapid assessment of alterations in pressures and heart rate. This allows an extremely flexible graphics display to be obtained which can be labelled using special character generator subroutines.

CHARACTER GENERATION

The character generator machine code subroutine which enables alphanumeric characters to be written on to the graphics screen is loaded by ROCHE 128 into locations $6000-$6100. The characters used by the generator are contained in a table loaded into locations $6900-$6C00. The generator selects the required character and plots it on the screen.

There are several character sets which can be used for this purpose but only the COLOSSAL.SET supplied with the Apple TOOL KIT disc plots lines of double thickness. The Apple colour graphics are designed so that only every alternate point is illuminated. A single line does not appear completely white and characters drawn on a coloured screen with a single thickness appear mottled and indistinct. However, if each line is plotted and then a further plot drawn immediately beside it, a clear white character results. The COLOSSAL.SET automatically plots each character with double lines and produces a bold white character.

Once the graphics trend has been selected, the user is asked if they require a complete plot or an expanded plot. The complete plot graphs every point stored in the memory while the expanded plot allows selection of a portion of the available data.

DERIVATION OF X-AXIS

Each block of data is a concatenated string which has as its first 6 characters the time of data collection. This is expressed as the number of minutes since the first of January and the graphics routine can plot between any number of points provided there are at least 2 min time between them. Time values are calculated using the pre-defined function TIME where :-

```
FN TIME(X) = VAL (MID$ ( Z$(X),1,6) )
```

The FIRST time value to be plotted is therefore FN TIME(1) and the LAST time value to be plotted is FN TIME(FI) where FI is the latest value obtained. The number of minutes (NM) to be plotted is then LAST - FIRST. If NM is less than 2, the program returns to the text menu at line 20000.

This subroutine is used in the programs ROCHEPRINT and ROCHEPRINTEXT to obtain the results which are required to be printed within a given time span for the generation of printed reports. It is also employed in the program FILEPART1 to select the data to be used for calculation of fluid balance over any selected period of time.

Subroutines at lines 25000 and 26000 convert the number of minutes to be plotted into the date for use later in labelling the X-axis of the trend graph. It is necessary to ensure that :-

1. maximum horizontal resolution is obtained for any given time scale.
2. only integers are used to label the time scale of the graph.

The routine which performs these tasks is contained at lines 13420-13600. If NM is less than 61, the graph is labelled with the number of minutes past the hour. The variable N is set equal to NM and tested at lines 13422-13424 to determine if it can be divided by 2 with no remainder. If this is not possible, then the attempt is made to divide it successively with the numbers 3 to 7. If a remainder is still left by these attempts, N is incremented by 1 until a successful division with no remainder is achieved.

The value of N now becomes the new number of minutes which will be plotted to ensure that only integers are used for labelling but may differ from the number of minutes calculated for the trend plot. The trend plot may therefore have to begin not at the start of the X-axis but at some distance along. This distance (BX) equals N-NM and is added to the base start value for the X co-ordinate of the trend plot. The number of divisions which will be marked on the X-axis is now N and each second mark is labelled with the appropriate number of minutes.

If NM was greater than 60 then an attempt is made to label the graph with the number of hours using a 24 h clock, up to a maximum of 30 h. The time of the first value to be plotted has already been converted to hours and minutes by the subroutine at line 25000. BX is set equal to the minutes portion and added to the total number of minutes to be plotted to ensure that the start values in the plot begin at the correct place relative to the X-axis.

The minutes portion of the first time value (BX) are therefore added to NM and NM is tested to determine if it is less than 2160 (30 h). If this is true, the number of hours to be plotted is NH and equals NM/60. NH is now checked to ensure that it is an integer since if it is not, some values to be plotted will be greater than the maximum available on the X-axis and would not be included. If it is not an integer, one extra hour must be added to NM before it is entered into the subroutine at lines 13422-13424 which provides integers of hours for labeling the X-axis. The value of N which is returned from this subroutine is the number of hours required to provide integer labels for the X-axis and is re-converted back to minutes for plotting the data.

When the total number of minutes to be plotted is greater than 2160, the X-axis is labeled with the date. BX then must be the hours and minutes portions of the first time value and is added to NM to give the minimum number of minutes for the X-axis. An extra day is added since only at exactly 24.00 h would there be no points greater than the maximum available on the X-axis. N is set equal to the number of days and entered into the subroutine at lines 13422-13424. The value of N returned is the number of days required and this is reconverted into minutes for data plotting.

DRAWING OF X-AXIS

The total number of minutes which must be accomodated on the X-axis is now available as NM; the number of divisions has been calculated with the subroutine at line 13422 as NX and drawing and labelling of the X-axis proceeds at lines 13505-13600. The command HGR2 switches the screen display to the second page of graphics to allow high resolution plotting and the colour of the plot is set to white.

The statements :-

POKE 54,0: POKE 55,96

alter the system pointers so that when text is required to be printed on the screen when the display is in graphics mode, the machine code character generator routine is invoked.

EXPANDED PLOT

If the expanded plot is selected, the user is asked to enter how far back from the present time the plot should begin. This can be any number followed by a letter to indicate M(inutes, H(ours or D(ays. The new START time value is obtained by examining from the first data string in the memory until a value is found to be greater or equal to the desired start time.

The user then enters the time interval to be plotted in a similar format to that used for the start point. The LAST time value is found by adding the number of minutes to be plotted to the start time and searching back from the most recent value until a value is obtained which is lesser or equal to this. If there are at least 2 min between the values to be plotted, the new START and LAST time values are entered into the same routines used for derivation of the X-axis for the complete plot and the axis drawn.

TREND PLOT

Scales are drawn on the Y-axis and labelled appropriately for pressures and heart rate. The values to be plotted are obtained at lines 14700-14870 by specifying the position of the first character in the data string as Z0 and the number of characters to be extracted as Z1. An offset is then specified (Z2) together with a scaling factor (Z3) which positions each value at the correct point relative to the Y-axis. The variable LO is also specified which indicates values below which no point will be plotted. This is required since there will not necessarily always be a value recorded each time for every measurement and the ommission of the LO variable would cause the plot to return to a 'Y' value of zero when no result was available. Once the Z variables have been specified, control passes to the plot subroutine at lines 13280-13410.

CALCULATION OF Y VALUES

The Y value is calculated using the pre-defined function PLT where :-

FN PLT(X) = VAL(MID$ (Z$(X),Z0,Z1))

The value of Y to be plotted from the data string then equals :-

190 - Z3 (the scaling factor) * FN PLT(X) +Z2 (the offset)

CALCULATION OF X VALUES

PI was obtained previously during the derivation of the X-axis and is the first point in the data array to be plotted. PP is a temporary variable used during the plot subroutine. The other temporary variables which are used, AX and CX are set to zero. BX was calculated, if applicable, during the derivation of the X-axis and was the offset required to correctly position the start of the data plots relative to the X-axis. If no offset is required then BX=0 and the plot starts at the screen position 30. If BX has a value greater than zero, the displacement along the X-axis is calculated at line 13282.

A test is applied to detect if PP is the last value in which case the subroutine returns. The value of FN PLT(X) is compared with the value of LO and if less, PP is incremented and CX is set equal to the difference in time between the suppressed value and the next data string in the array. The program returns to line 13282 and an attempt is made once more to calculate an acceptable starting point.

If the value of FN PLT(X) is acceptable, the X and Y values which have been calculated are tested to ensure that they are not outwith the plotting range of the system. The first acceptable point has therefore been obtained and all succeeding points are calculated in a similar manner except that AX is the variable used to store the summated times of Y values which are suppressed because they are less than the value of LO.

Once all the available points for one variable such as heart rate have been plotted, control is returned to the subroutine at lines 14700 - 14870 until the trend plots for all the cardiovascular variables have been drawn. These are systemic systolic and diastolic pressures, heart rate, CVP and PAWP. The time when the trend was plotted together with the present time and a selection menu are then displayed. From this menu, the trend plot can be expanded to enlarge any section provided that the required portion is of at least 2 min duration.

A print-out of the entire graph can be obtained at this point or the user can select to return to the main text menu. All selections again are made by a single key stroke which sets the keyboard strobe to indicate that a key has been depressed. The character is then examined to detect if it is within the allowable range and if not, the program returns to the TIMING subroutine to collect data at required time and to monitor the systemic and pulmonary pressures.

7.

The default value for the sampling time of data is 30 min but can be altered to any value between 1 and 120 min using this option.

8.

Selection of option no. 8 provides the opportunity to collect a new set of data from the ROCHE 128 or print out a 24 h report by calling the program ROCHEPRINTEXT. This program is described later.

9.

The default sampling time for respiratory data collection is 1 h but option 9 allows the user to exit from the program or to collect a series of respiratory measurements from a Seimens Servo 900 B ventilator at any time by calling the program ROCHESERVO.

If no key has been touched or if the character is not in the range 1 to 9, the keyboard strobe is reset and the program rechecks the systemic pressures and, if applicable, the pulmonary arterial pressure before returning to the timing routine at line 890.

ROCHESHUNT

Option 5 on the text menu is initially 'COLLECT CARDIAC AND SHUNT DATA'. When this is selected, the user is asked to enter the cardiac output, PAWP and CVP. The computer obtains the cardiovascular data and stores all the information in SHUNTFILE along with the flags and constants used in the cardiovascular programs. Option 5 is changed then to 'ENTER SHUNT DATA'. When the results of the analyses of the arterial and mixed venous blood samples are available, the user selects option 5 and the program ROCHESHUNT is loaded and run.

The program first obtains the patient name to which all the data collected refers from the disc file, NAMEFILE. The cardiovascular variables which had been stored at the time of sampling of blood for shunt calculations are now obtained from SHUNTFILE and stored in the computer memory.

Results of blood gas analyses, haemoglobin concentration, FIO2 and PEEP are entered into the system from line 10000 and, as in other data entry routines, they are validated against acceptable ranges to detect obvious errors. If results are available from co-oximeter measurements of the blood samples, they are also entered to allow later comparison with calculated saturation values.

The equation used to calculate saturation begins at line 10800 and includes appropriate correction for pH and PCO2 (73), but no correction is made for 2,3-DPG. The co-oximeter results, if available, are used in preference to the calculated values for derivation of oxygen content (14) to provide greater accuracy. A value of 1.34 ml per g haemoglobin is used for calulation of oxygen content (23).

The following results are calculated, printed with the appropriate headings and stored on the disc :-

1. percentage intrapulmonary shunt

2. arterial-venous oxygen difference (ml/100ml blood)

3. oxygen consumption (ml/min)

4. oxygen delivery (ml/min)

The cardiovascular program which called SHUNTFILE is obtained from the file NEXTPROG. The relevant cardiovascular program, either ROCHE 128 or R2000, is loaded and run but since SHUNTFILE now contains data, this information is loaded by the cardiovascular program, SHUNTFILE deleted, and a further block of data obtained from the monitor.

ROCHEPRINT

This program is accessed from the main cardiovascular program using option 8 as described above and again, loading of this program is preceded by writing the relevent information into SHUNTFILE and NEXTPROG to ensure automatic re-starting of ROCHE128 when it is re-called after ROCHEPRINT has performed its task.

ROCHEPRINT is largely a combination of routines from other programs organised to produce a print out of the numerical and graphical data of the preceding 24 h. The cardiovascular records are obtained as for the main program and the START and LAST records are obtained from the routine at line 13001 which had been used previously for deriving the X-axis for the trend plot. START is taken as 24 h before the last recorded data and LAST as the most recent. The date is obtained from the Mountain Hardware Clock/Calendar Card using the standard subroutine at line 24000. The data are printed from the most recent (indexed by F2) to the last (indexed by PP).

Routines had been developed in the program FILEPART1 to allow manually entered data to be printed out as hard copy. This program is described later, but the print routines have been used in ROCHEPRINT and are located from line 5000. They split the data strings into the individual numbers based on the values of two arrays of variables - S(A) and L(A).

The S-array specifies the start position in the string of the number and the L-array specifies the number of characters which make up the complete number. The individual values obtained are stored in a temporary array SM$(A) with SM$(0) as the date (DD/MM) and SM$(1) as the time (HH/MM). The date is printed out for the first value and thereafter only if it alters. Each group of values are printed, preceded by the time of collection.

Once all the values have been printed, the trend plot of that 24 h period is graphed as in the graphics routine in the main cardiovascular program but the labels are written using a single plot character set 'CHARTAB A$6800 L$400' loaded at locations $6800 to $6C00. The single plot provides better definition when used with a printer compared with the double plot employed for the graphics display on the screen in the main cardiovascular programs.

The same process is repeated for the respiratory data. The program which called ROCHEPRINT is obtained from NEXTPROG and executed. The flag, FLAG$, which was used to indicate that the patient identification did not require to be printed out every time data is collected, is reset to 'N' since the 24 h report will be removed from the printer and identification is necessary of the subsequent patient data which will be collected and printed out.

ROCHEPRINTEXT

The nursing staff requested that the facility for entry of comments which merely printed them with the time of entry, should be extended to allow them to recall this information at a later time. The cardiovascular programs were therefore modified to store each text string in a similar manner to that used for the cardiovascular records.

ROCHEPRINTEXT obtains the patient name, the number of text records, and calculates the START and LAST strings to be printed in the same way as for the cardiovascular and respiratory information printed out by the program ROCHEPRINT. The text records are obtained from the patient's file and printed out along with the time of entry. The calling program is obtained from NEXTPROG, loaded and run.

ROCHE 2000 SYSTEM

The Roche 2000 patient monitor system consists of several different modules which can be inserted into a rack. Three pressure channels and e.c.g. can be monitored with this system. The waveforms are displayed on a cathode ray tube and digital values of heart rate and pressures are shown on the separate modules. This system has the advantage that systemic pressure, pulmonary artery pressure (PAP) and central venous pressure (CVP) can be measured simultaneously whereas with the Roche 128 monitor, only 2 pressure channels are available.

The analogue outputs for the mean pressures have been modified to allow rapid updating of the signal. This permits the program to extract either the smallest or the largest values for the mean PAP and CVP during variations of intrathoracic pressure with respiration, depending on whether the patient is receiving positive pressure ventilation or breathing spontaneously.

No digital outputs are provided, but the heart rate, systolic, mean and diastolic values of the three pressure channels are available as separate analogue outputs. These were connected to a 16-channel analogue to digital converter (ADC; Interactive Structures AI-02) and the data collection routines in the program altered to suit this form of data output from the Roche 2000 system.

Amplifier circuits were constructed by the technical staff of the University Department of Anaesthetics at Glasgow Royal Infirmary to produce satisfactory voltage levels for connection to the ADC. The ADC is an 8 bit device which requires the signals to be in the range of 0-5V and the amplifiers used for the PAP and CVP provide an offset to accomodate negative pressures in these signals. A calibration procedure has to be undertaken to obtain the offsets on the ADC for the CVP and PAWP channels and the calibration factors for all channels.

The full range for the different pressure channels is :-

CVP	:	-20 to +30 mmHg
PAP	:	-20 to +80 mmHg
HR	:	0 to 255 b.p.m.
BP	:	0 to 255 mmHg

The greetings program, HELLO, which is run when power is applied to the system, requires the user to select the type of instrument connected to the computer and writes the relevant program name into the disc file, NEXTPROG. If the Roche 2000 monitor is selected, ROCHESETUP calls the calibration program to allow the zero offset and calibration factors to be correctly set.

R2000CALIBRATE

The arrangement of the modules in the Roche 2000 rack and the pressures which they will record are displayed on the screen. The CVP is recorded on channel 3 and the PAP on channel 2. These can be deleted from the list of active channels if they are not connected to patient transducers so that the computer monitoring system will function with only the heart rate and a systemic pressure signal.

The calibration values which had been used previously are obtained from the file R2000VARIABLES and the program obtains data from the Roche 2000 modules by passing to the subroutine DISPLAY VALUES at line 5000. This selects the appropriate channels on the ADC, obtains the digital values, applies the zero offsets and calibration factors and displays the calculated results for the pressures and heart rate on the screen.

If no data is present in R2000VARIABLES, or if the user wishes to re-calibrate the system, the program passes to the calibration subroutine at line 700. The user is asked to set all channels on the Roche 2000 to zero and type the spacebar. The system then collects from the ADC the offset values and stores them as follows :-

1. ZRT - heart rate zero offset

2. ZSYS(1-3) - systemic pressure offsets;
   systolic, diastolic, mean

3. ZPUL(1-3) - pulmonary pressure offsets;
   systolic, diastolic, mean

4. ZCVP - central venous pressure offset

The user is then asked to set each channel in turn to the desired value to allow the gain factor to be calculated. The pressure which has been applied to the transducer is typed into the computer and the calibration factors are calculated and stored as follows :-

1. CRT - heart rate calibration factor

2. CSYS(1-3) - systemic pressure factors,
   systolic; diastolic; mean

3. CPUL(1-3) - pulmonary pressure factors,
   systolic; diastolic; mean

4. CCVP - central venous prssure calibration factor

If the screen display is inaccurate, the user can select to re-calibrate the modules. If the values displayed are accurate, the zero offsets and calibration factors are stored in the file R2000VARIABLES for use by the main program R2000 which is then called by R2000CALIBRATE.

R2000

Whereas the Roche 128 program can only function with one type of cardiovascular monitor, the R2000 program can function with any monitor which has suitable analogue outputs for the presure signals and greatly extends the range of applications of this suite of programs. The principal differences between ROCHE 128 and R2000 are in the data collection and text menu routines.

DATA COLLECTION

The data collection routine for the Roche 128 involved the use of machine code subroutines to obtain the digitised information from the Roche 128 and place it in the computer memory. Data collection from the Roche 2000 requires the program to select in turn each ADC channel to which the analogue outputs from the Roche 2000 are connected. The selection of the channel automatically instructs the ADC to begin conversion of the analogue signal.

Data collection can be undertaken in BASIC because there is no need to collect the information in a very short period of time. The delay which is inherent in the BASIC interpreter executing the next instruction is greater than the 70ms conversion time of the ADC and so the program can proceed immediately from the instruction to select the channel on the ADC for conversion to the instruction to read the converted value on the ADC. The zero offsets and calibration factors are read from the R2000VARIABLES file and used to produce the final values.

The variables used to store the data obtained in memory are exactly the same as in the ROCHE128 program except for the core and peripheral temperatures which are not available on the Roche 2000 system. While this program was under development, an Oximetrix co-oximeter was made available. This allows the mixed venous oxygen saturation (SvO2) to be measured continuously from a pulmonary artery catheter which contains two fibre-optic cables. One conducts light pulses of three different wavelengths into the pulmonary artery while the second fibre-optic cable conducts light reflected back from the haemoglobin in the red blood corpuscles to the measuring instrument.

The standard method of recording the SvO2 is on a paper chart but it became evident that it would be valuable to have the data collected on-line by the computer to allow correlation of SvO2 with the other cardiovascular data collected. An interface was therefore constructed to amplify the analogue output of saturation from the co-oximeter to a suitable level for connection to the ADC. The value obtained for the mixed venous oxygen saturation was placed in the portion of the data string and print-out normally occupied by the temperatures recorded from the Roche 128 monitor. A Hewlet Packard ear oximeter, which measures the saturation of capillary blood, was used to obtain an estimate of the systemic arterial oxygen saturation and also linked into the computer.

The R2000 program was then too large to fit into the space available below the second page of high resolution graphics and the section which allowed the user to indicate which devices were attached to the computer was re-written as a separate program, R2000SETUP. The same information regarding the printer, pulmonary artery catheter and Seimens 900B ventilator is obtained but the user is also asked if the Oximetrix analyser and ear co-oximeter are connected.

If the ear co-oximeter is in use, the haemoglobin concentration and the inspired oxygen concentration are entered. These values are formed into a character string and when the device flags are stored on disc in SHUNTFILE, the haemoglobin-FIO2 data string is stored in place of CVP$ which is not used as a flag in the R2000 program. Unfortunately, the ear co-oximeter did not provide sufficiently accurate estimation of arterial PO2, but the program alterations have not been deleted since it is possible that an alternative system such as a transcutaneous PO2 electrode may provide better data and could be used in place of the ear co-oximeter.

TEXT MENU

Alterations to the TEXT MENU for R2000 involved incorporating into option 2 the ability to alter the calibration factor for the Oximetrix co-oximeter. When the Oximetrix analyser is in use, the venous saturation is displayed at the bottom of the screen instead of the channel identification used in ROCHE 128.

Option 2 of the R2000 program also allows the mode of ventilation to be altered. The user is asked at the start of the program whether the patient is receiving intermittent positive pressure ventilation. If this is the case, IPPV$ is set to'Y' and when data are written into SHUNTFILE, IPPV$ is written in place of PT$. PT$ was used in ROCHE 128 to indicate the channel which was recording the systemic pressure. The channel identifications for the R2000 are stored in the file R2000VARIABLES and PT$ is not used in R2000. If the patient is receiving IPPV, the smallest value of mean CVP and PAWP are stored. When the patient is breathing spontaneously, the largest values are stored.

The other programs in the suite such as ROCHESHUNT and ROCHEPRINT obtain the name of the main cardiovascular program to which control must be returned by reading the disc file, NEXTPROG. A single floppy disc contains all the programs necessary for both types of monitors.

The R2000 programs merely require analogue ouputs of heart rate, and the systolic, mean and diastolic values of up to three pressure channels. The range of suitable cardiovascular monitors which could be connected to the computer is increased greatly . While this suite of programs in their present form will only run on the Apple II microcomputer, the design of the data storage, date/time identification, graphic displays and text menus has been successfully transferred to a system based on a PDP 11/23 which was recently installed in the University Department of Cardiac Surgery at the Western Infirmary in Glasgow.

ROCHE MACHINE CODE PROGRAMS

LOAD.OBJ0 / CARD

These machine code programs obtain the cardiovascular data from the Roche 128 monitor. The ROM on the Interactive Structures RS 232 serial card is replaced with a 2K RAM integrated circuit and the standard driver routine which is normally stored in the ROM is replaced with the machine code program CARD. This is a modified version of the program normally stored in the ROM and was written by Mr Ian Boyle of Strathand Ltd. The transmit request character to the communication interface in the Roche 128 monitor is hexadecimal $05 but the data transmitted by the serial card normally has bit 8 set high which would transmit the hexadecimal character $83. CARD removes this high bit and transmits the correct character to the Roche 128.

LOAD.OBJ0

LOAD.OBJ0 requests and collects the data and the organisation of this program is shown in figure X. The card is initialised and the transmit request character $05 is placed in location $C0D1. When the Roche 128 receives this character, it transmits a character string containing the cardiovascular and temperature data. The final character in this string is $04 and when this is detected by the computer, control is returned to BASIC.

The initial version of this program was written by Mr. I. Boyle, Strathand Ltd. It has since been re-written to improve its performance. Occasionally, the microprocessor in the Roche 128 could not service the data request from the computer and the string of data was not transmitted. A timeout routine was incorporated into LOAD.OBJ0 to ensure that if a character is not received within a predetermined time, the error is detected and control is returned to BASIC which generates an appropriate error message.

REPEAT

After the character $05 has been sent to the Roche 128, the 'X' register is incremented and the program passes to DELAY to allow time for the transmission of a further character from the ROCHE 128. A test is then made to detect if a new character has been obtained from the Roche 128. If none has been received, the program passes again to the beginning of the REPEAT routine and 'X' is incremented once more. When the value of the 'X' register becomes greater than $FA, the program passes to the TIMEOUT routine to signal that a timeout error has been detected. TIMEOUT stores $01 in location $370 and returns control to BASIC.

The BASIC program then examines the contents of $370 and if it is found to be 1, an audible and visual warning is given that the monitor may be disconnected from the computer. The BASIC program immediately attempts further communication with the ROCHE 128 and, if this is successful, prints and stores the data. If it is still unsuccessful, the warning is repeated until satisfactory communication is restored.

When a new character has been received from the monitor, it is stored in location $380 indexed by 'Y'. The 'X' register is set to zero and the program passes once more to the beginning of the REPEAT routine. When all the data has been obtained from the Roche 128 and the character $04 has been received, $00 is loaded in to location $370 to indicate that the data transmission has been successful and control is returned to BASIC.

ANALYSIS OF RESPIRATORY WAVEFORM

Several investigators have emphasised the value of the analysis of volume-pressure waveforms as an aid to monitoring patients in the intensive care unit (32, 46, 59, 60, 65, 98). Changes in the condition of a patient can be detected by such waveform analysis. For example, alterations in the total pulmonary compliance can be detected and analysis of the trend of these changes allows early recognition of conditions such as pneumothorax before these become apparent (62).

The analysis could be made from a volume-pressure graph plotted on an X-Y plotter but this would be so time-consuming that it would be impractical for anything other than a limited research project. In addition, even fast response X-Y plotters do not move with enough speed to follow the rising edge of the curve (61). Prakash (63) maintains that an analogue osilloscope display of the volume/pressure graph is adequate for clinical use, but derived numerical data in addition should provide more information to guide patient therapy more appropriately.

Initial workers employed an analogue system to perform the calculations with the inherent disadvantages of any analogue system (6). The use of a digital mainframe computer has also been reported, but the cost was extremely great and consequently could only be available in a limited number of hospitals (46). A microprocessor-based system has been developed (94) but is programmed in machine code which is more difficult to alter and update than a higher level language.

The Apple II microcomputer is sufficiently powerful to undertake this task (18) and has the additional advantage of providing graphical displays as recommended by Prakash (63).

The criteria set out at the beginning of the section on the use of computers in the intensive care unit are met by this application. It automatically provides for the nursing staff information about tidal volume, minute volume, various pressures during the respiratory cycle and timing information which they would normally have to collect for themselves. They therefore have to spend less time in clerical duties and have more time to spend nursing patients. This is of major importance where a seriously ill patient may have a pulmonary artery catheter inserted, is receiving dopamine and sodium nitroprusside and the nurse is measuring PAWP and cardiac output in addition to her other duties.

The same computer program provides information on pulmonary compliance and respiratory work which the medical staff in the unit would not normally have available to them to assist in determining the optimum therapy for their patients.

On-line computation of various lung characteristics can be made from values of pressure and flow obtained from transducers placed within the patient circuit. The ROCHESERVO program obtains the required data from a standard Seimens Servo 900B ventilator. The Seimens ventilator is especially suited for this purpose since it provides analogue outputs of pressure, inspiratory and expiratory flows together with a signal which indicates whether the machine is in inspiratory or expiratory mode. These signals are available at the rear of the machine without modification. An analogue lung mechanics calculator is commercially available (35) which can only provide some of the information produced by ROCHESERVO at greater cost.

The Mountain Hardware Clock/ Calendar card, which is used to obtain the date and time in the cardiovascular programs, is used for this same purpose in ROCHESERVO. In addition, it is used to time the inspiratory and expiratory portions of the respiratory cycle to an accuracy of 1 ms.

The Interactive Structures AI-02 ADC is also used in this program to digitise the analogue signals. The maximum conversion rate which can be obtained with the ADC using the PEEK and POKE commands to access memory locations, is approximately 40/s.

Three conversions must be made of pressure, flow and the inspiration/expiration signal to collect one point on the pressure-volume graph. If BASIC was used, only 13 data points would be obtained which would not provide adequate information when the waveform was changing rapidly.

Turnley and Blumenfeld (89) used a Wang 720B desk calculator to process data from respiratory flow signals and claimed acceptable results with a data sampling rate of 15 /s. However, the rapid initial part of expiratory flow is underestimated (61). Therefore, data collection in ROCHESERVO is performed in machine code and the ADC is accessed to obtain approximately 450 samples per second which provides data on the pressure-volume curve at a rate of about 150 /s.

The flow signals from the Seimens ventilator exceed the +5 V maximum allowable for the ADC and have to be attenuated while the pressure signal requires amplification to obtain maximum resolution. This signal conditioning is performed in a separate interface which was designed by Mr P Davis of the Clinical Physics and Bioengineering Department, West of Scotland Health Boards and constructed by the technical staff of the University Department of Anaesthetics at Glasgow Royal Infirmary. The machine code program which is used for data collection was written in conjunction with Mr P Davis. Power for the interface circuit is obtained from the same socket which provides the analogue signals.

The accuracy of the signals has been assessed previously (43). The accuracy and linearity of the computer system in measuring pressure were determined by comparing the pressure displayed by the computer with a water manometer at eight values of pressure from 5 cm H2O to 40 cm H2O in increments of approximately 5 cm H2O. Five separate measurements were made at each pressure.

The accuracy of measurement of the inspiratory flow was determined by passing the gas flow from the inspiratory port of the ventilator through a Parkinson Cowan dry gas meter for a period of 5 min. The average minute volume obtained was compared with the value calcualated by the computer over the same period. Five measurements were made at each of eight minute volumes from a ventilator setting of 2.5 litre/min to 20 litre/min in increments of 2.5 litre/min. The expiratory flow was compared in a similar manner by passing the gas flow from the expiratory port of the ventilator through the gas meter. Five measurements were made for expiratory flow using the same ventilator settings as for the inspiratory measurements.

The system proved to be accurate and linear for pressures and flows over the ranges measured. The results of the comparison of the computed values with the measured values are shown in table VII.

ROCHESERVO can be used alone or can be called from the Roche cardiovascular programs. The choice is made at the initial HELLO program which is run when power is applied to the system. This prompts the user to indicate whether one of the two Roche monitors is connected to the computer or if the Seimens Servo 900B ventilator is to be used alone. When the ventilator is to be used alone, ROCHESERVO is written into the NEXTPROG file and all other ancillary programs such as ROCHESETUP and FILEWRITE return to ROCHESERVO as the central program.

When one of the Roche monitors is connected to the system, the relevant cardiovascular program, ROCHE 128 or R2000, is identified as the central program and ROCHESERVO itself becomes an ancillary program. Flags and variables used in the Roche cardiovascular programs are set at the start of these programs and the user is asked to indicate if the computer is connected to a Servo 900B ventilator. If this is so, a flag 'SERVO$' is set to 'Y'; otherwise the flag is set to 'N'.

The program, ROCHESERVO, is called from the main cardiovascular program when the preset timeout in the cardiovascular program is exceeded. In addition, measurement of respiratory variables can be undertaken at any time by selecting option 9 from the main text menu. Option 9 sets the flag 'SERVO$' to'Y' if it had been previously set to 'N' and runs ROCHESERVO. This allows a patient to be transferred to a Servo from a different ventilator and indicates to the cardiovascular program that respiratory measurements can be made on-line.

ROCHESERVO

The program will be described first as if it was called from the Roche programs and then the sections which are required for independent operation will be described. The overall organisation of ROCHESERVO is shown in figure 23. The program sets the lowest memory location available to BASIC data storage at 32100 with the command LOMEM:32100. This ensures that the machine code routines and the data stored by them will not be overwritten by BASIC variables. The disc text file, SHUNTFILE, is then opened and read to collect the flags and start-up variables for the cardiovascular program which called ROCHESERVO. These are required in case an error has been made and the computer is not connected to the ventilator. The title of the program is displayed on the screen using the routine at line 7.

A test is then made to ensure that the Seimens ventilator is connected to the computer. The ADC is accessed by POKing the channel number selected for conversion into the location identified by the variable CHAN. The digital results are read by PEEKing the location identified by the variable ADC :-

ADC = -16384 + (256 X 4)
CHAN = ADC + 1

where the ADC card is placed in slot 4.

The program waits at line 50 during inspiration until either the I/E signal changes from low to high, which indicates the beginning of expiration, or more than 400 values have been obtained. This indicates that the signal is not changing and the ventilator is not connected. The program then passes to line 55 where the same tests are made during expiration. Occasionally the signal line was found to change voltage level spontaneously even though it was not connected to the ventilator. Lines 60 - 80 form the principal check to ensure connection to the

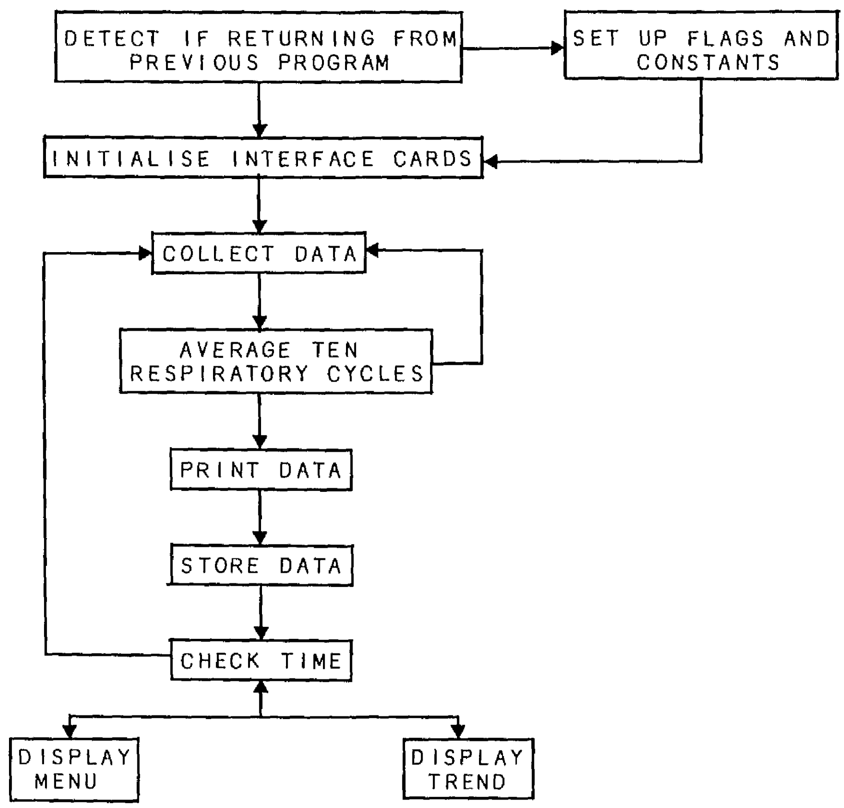

Figure 23 Organisation of the ROCHESERVO program.

ventilator.

The total number of times the program has looped during inspiration and expiration are examined at this point and only if they are each within the range 30 - 400, does the program continue to the data collection routine. A single change in voltage returns a value of less than 30. If a value of greater than 400 is returned, no alteration in voltage has been detected within the allocated time period.

If this test is not true, it is assumed that there is no Servo ventilator connected to the computer and the program reloads and executes the Roche program which called ROCHESERVO initially. The name of this Roche program is obtained by reading the file name contained in the file NEXTPROG. The flag, 'SERVO$', is set to'N' and SHUNTFILE is re-written to ensure that the cardiovascular program will not continually call ROCHESERVO. However, if the I/E signal changes adequately, ROCHESERVO continues.

The respiratory data is organised in an identical manner to the cardiovascular data in that the individual values obtained are concatenated into a string of characters with the first 6 as the date/time reference. The array used to store these blocks of data is Z$(X) and this is dimensioned at line 110 to allow up to 200 values to be stored. The number of points used in the calculations and volume-pressure plots can be altered by changing the variable 'NP' at line 115 which acts as a dividing factor in the loops where the data stored by the machine code program is extracted in BASIC.

The machine code programs, CHARGEN A$6000 L$100 and COLOSSAL.SET A$6900 L$300 which allow text to be printed on to the graphics page, are loaded followed by the machine code data collection routine RESP.OBJ0. Functions PLT and TIME which are used in the trend plot subroutine are defined and control passes to the routine READ PREV DATA at line 3000.

READ PREV DATA

This routine reads the NAMEFILE and obtains the previous respiratory data for the patient if any is available. It then collects the calibration factors for the analogue outputs from the Servo ventilator by reading the file SERVOVARIABLES. If no file is present, preset values are used instead. The default variables used for the volume-pressure scaling, HY,NY,HX and NX are set and the flag 'RPLT', which is used to indicate that a graph is being re-plotted, is set to zero. The program then passes to line 200 for data collection.

DATA COLLECTION

Location 27798 is set to zero since this is used by the machine code subroutine to indicate when the duration of the respiratory cycle has been excessive and that the memory space allocated for the data has been exceeded. The values of flow and pressure are obtained by the machine code subroutine during one respiratory cycle and stored sequentially in an allocated segment of memory. When inspiration changes to expiration, this is detected and the length of the inspiratory array stored. At the end of expiration, the total length of the array is saved in memory. The clock is also read to enable the inspiratory and expiratory times to be calculated. The subroutine is described more fully later in the section dealing with the machine code programs.

When control is returned to BASIC, location 27798 is re-examined and if it contains the value, decimal 34, the space allocated for the storage of data by the machine code program RESP.OBJ0 has been exceeded and a visual and audible warning are given to indicate that the ventilator may be disconnected. Otherwise, the program passes to line 225 where the lengths of the inspiratory and expiratory arrays are obtained.

LOCATION OF DATA

The start of the data array is location 27800. The location of the last entry in the inspiratory array is stored in 2 bytes. These 2 bytes are contained in two memory locations, 27776 and 27777. The low byte is obtained by PEEKing location 27777 and adding the low byte of the start address, $98 :-

PEEK (27777) + 9 X 16 + 8

The high byte is obtained simply by PEEKing location 27776 and the location of the last entry in the inspiratory array is :-

PEEK (27776) X 256 + PEEK (27777) + 9 X 16 + 8

L1 is set equal to the start of the array and L2 to the end of the inspiratory array. The length of the inspiratory portion (IC) is therefore L2-L1. The location of the last entry for the expiratory portion of the array is also stored in 2 bytes. The low byte is obtained in a similar manner to that used for the end of the inspiratory portion :-

PEEK (27779) + 9 X 16 + 8

The end of the expiratory part of the array is therefore :-

PEEK (27778) X 256 + PEEK (27779) + 9 X 16 + 8

L3 equals L2+1 and L4 is set equal to the end of the entire array. The length of the expiratory array (EC) is therefore L4-L3.

OBTAIN TIMINGS

Three times are stored during each respiratory cycle :-

1. TIME 0 - start of inspiration
2. TIME 1 - end of inspiration
3. TIME 2 - end of expiration

The data collection routine stores six ASC II characters for each time in successive locations from 27780. These are extracted from memory and concatenated into the strings T1$, T2$ and T3$. The first instruction of the routine OBTAIN TIMINGS is to set each of these these three strings to the null string (""). Each time-string is constructed by adding the successive characters obtained by PEEKing the relevant location, clearing the most significant bit by subtracting 128 and using the CHR$ command to produce the digit. The time-string then contains the number of seconds on the clock when it was interogated by the data collection program, for example '34.678'. The numerical values of the three time strings are found using the VAL command.

Since only the number of seconds are stored, it may be necessary to correct the time values because the succeeding time may have been recorded after the minute had changed. For example, if T1 was 59.234 s, T2 would have a smaller numerical value such as 0.734 s if inspiration lasted 1.5 s. If T2 is less than T1, 60 is subtracted from T1 to produce the correct duration of inspiration. Similarly, if T3 is less than T2, 60 is added to T3 to produce the correct value for the total time of the respiratory cycle. The inspiratory time (TI), expiratory time (TE) and respiratory rate are then calculated.

The conversion time for each sample of pressure and flow is calculated, CI for inspiration and CE for expiration. The results for volume are corrected for pressure and the data are now ready to be used to produce the derived results and a test is now made at subroutine EVALUATE KEYPRESS at line 2473 to determine what form of presentation is required by the user.

EVALUATE KEYPRESS

During data collection by the machine code subroutine, a selection menu is displayed on the screen which indicates the possible options available to the user to direct subsequent flow of the program :-

1. tabular display of results
2. plot of volume-pressure loop
3. trend plot of data obtained previously
4. menu which allows other options such as alteration of the data retrieval time or switching the printer flag on or off.

Each of these four options are obtained by typing the relevant key as directed by the selection menus. When control passes to the subroutine EVALUATE KEYPRESS, the keyboard buffer is read by the KEYPRESS subroutine and the ASC II code of any key which has been typed is stored as 'X'. The value of 'X' is then compared with those permitted, and if a valid key has been typed, the required program control flags are set. The flags used are :-

1. PLT$ - set to'Y' if the volume-pressure plot has been selected.
2. RP - set to 0 if a new graph plot is to be drawn. This is used in conjunction with PLT$ in that when the first volume-pressure loop is drawn, RP is set to 1. Successive plots are then drawn superimposed on each other.
   It is also set to 1 when a trend plot has been drawn and prevents the plot being redrawn after each breath.
3. TR$ - set to 'Y' if a trend plot is selected.
4. XP$ - set to 'Y' if a section of the trend plot is to be expanded.
5. ME$ - set to 'Y' if the user wishes to view the principal menu for the additional options.

The program returns to line 340 and, depending on the settings of the flags, is directed to the subroutine selected. On return from the selected subroutine, the calculations for that respiratory cycle are performed by the routine CALCULATIONS at line 800. If no option has been selected, the default is to display the results in a tabular form and the calculations are performed immediately before the results are displayed.

CALCULATIONS

Variables used in the calculations are set to zero at line 803, and if none of the graphics modes have been selected, the comment 'CALCULATING RESULTS' is printed at the bottom of the screen display. The arrays F(X) and P(X) are used in the loop which extracts the volume and pressure data placed in memory by the machine code program. The digitised values for flow and pressure are stored sequentially from location 27800 in the form :-

F(1) : P(1) : F(2) : P(2) :::::::::::: F(X) : P(X)

Inspired volume (VI) is calculated by adding the individual flow values and multiplying the sum by the scaling factor for inspiratory flow (SI) and the conversion time for the inspiratory data (CI). Peters and colleagues, (60) and Proctor and Woolson (66) have recommended using inspiratory work as a predictor of requirement for mechanical ventilation. Inspiratory work (IW) is calculated by multiplying the product of flow and pressure by the conversion time and the scaling factors for pressure (SP) and inspiratory flow. The maximum pressure achieved during inspiration is stored as the variable PP and the pressure recorded at the end of inspiration as PAUSEP.

The expiratory data is calculated from line 859. Expired volume (EV) is obtained in a similar manner to inspired volume, but no valid measurement can be made for expiratory work since the pressure driving gas out from the lungs during expiration is the intrapleural pressure and is not measured by the system. Minute volume (EM) is calculated and the end expired pressure is stored as the variable 'PEEP'. Compliance, and the inspiratory resistive component of work are also obtained. Although both the pulmonary and thoracic wall compliance are measured by ROCHESERVO, Osborn and colleagues (1969) reported that alterations were largely caused by changes in the pulmonary component.

If none of the flags strings, TR$, PL$, XP$ or ME$ has been selected at the subroutine EVALUATE KEYPRESS, the results of the calculations are displayed on the screen in tabular form along with patient identification and time and then passed to AVERAGING ROUTINE at line 2600. However, if one of these flags has been selected, an alternative form of data display has been selected and the the program passes directly to AVERAGING ROUTINE. The data are averaged over 10 breaths before being printed out and stored on disc as a concatenated string in a similar manner to that used for storage in the cardiovascular programs. The options available from the tabular display are :-

M(enu
V(OL pres
G(raph

The required choice is entered by typing the first letter of the option selected.

If the M(enu is required, control is passed to the subroutine at line 20000. Typing 'V' displays the data collected from the next respiratory cycle in the form of a volume-pressure loop, while typing 'G' graphs the results obtained from previous analyses as a trend graph.

MENU

This allows the options of numerical display of results, volume-pressure or trend plot to be obtained. It also permits other options to be selected which would be used principally when ROCHESERVO was operating independently from the Roche programs.

The list presented to the user is:-

1. tablular display
2. volume-pressure loop
3. entry of text comment
4. change output from screen to printer and vice versa
5. calibrate Servo ventilator analogue outputs
6. trend display
7. alter data retrieval time
8. collect new data
9. end the program

The selection made is identified by scanning the keyboard strobe during program execution in the same manner as for the menu selection in the Roche programs. This allows the user to interrupt the program if required.

1. TABULAR DISPLAY

This displays the results of the analysis of the most recent respiratory cycle. An example of the information which is provided is shown in figure 24. The volumes, pressures and respiratory rate are normally recorded by the nursing staff who require several minutes to take sample measurements every hour. When the computer system is attached to their patient, they have to simply note the results of the most recent printout.

2. VOLUME-PRESSURE GRAPH

When this option is selected, the information collected by the machine code program is used to construct a volume-pressure loop before the calculations are performed. The string variable PLTFLAG$ is set to 'Y' and the program passes to the routine AXES PLOT at line 2040 where the axes are drawn and labelled.

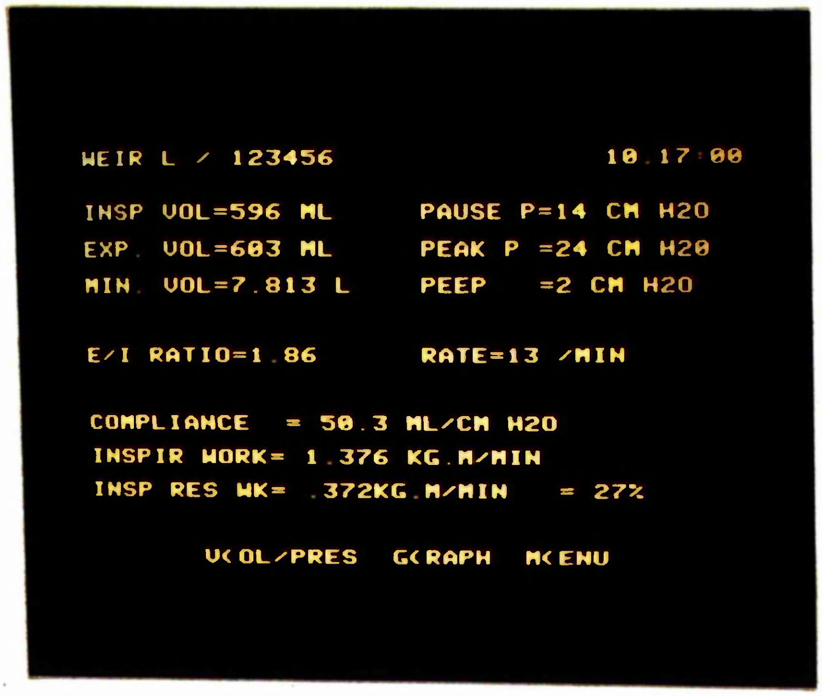

Figure 24 Tabular display of respiratory data calculated by ROCHESERVO.

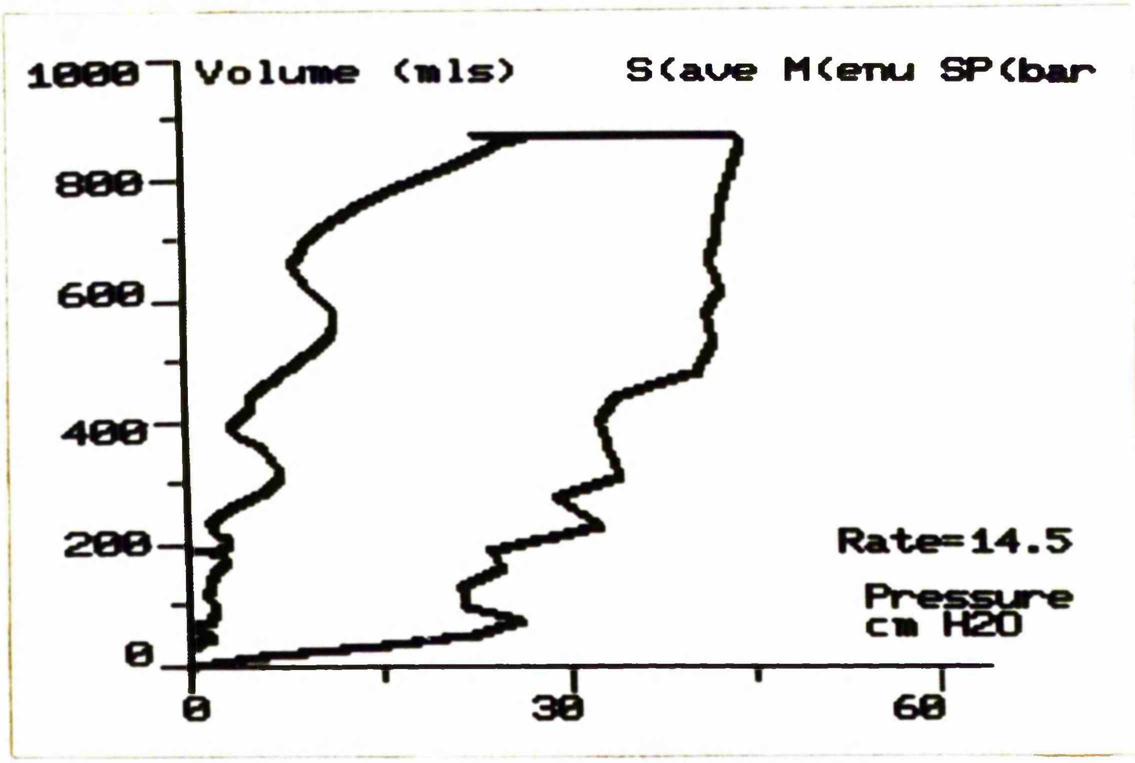

Figure 25 Volume/pressure graph obtained when the ventilator tubing had accumulated an excessive amount of water.

Axes Labels

The axes are automatically scaled to suit the maximum pressure and volume recorded. A test is made to fit the axes to the last calculated peak pressure and tidal volume at the subroutine SCALE at line 2135. The X-axis can be scaled for a peak pressure of 20, 40 or 60 cm H20 and the Y-axis to a maximum volume of 500, 1000, or 1500 ml. The smallest values are taken if no information is available and a further check is made once the inspiratory portion of the loop has been plotted and the inspired tidal volume calculated. The axes are plotted from zero and the variables LX and LY are set to 0 to indicate this. Four divisions on the X-axis and 10 on the Y-axis are drawn by setting NX = 4 and NY = 10.

The label for the X-axis is allocated to ZZ$ and that for the Y-axis to Y$. The character set used to write alphanumerics on the high resolution graphics screen has lower case letters and the type of character to be used is organised by the subroutine at line 2090. The screen is cleared, the second page of high resolution graphics, HGR2, and the plot colour selected, and the pointers in locations 54 and 55 set to 0 and 96. These indicate the position of the character set which will be used when text is printed on the graphics screen. The X and Y-axes are drawn as two lines next to each other to ensure that a solid white line is obtained. The divisions and their numerical values for the axes are drawn from lines 2260 to 2300 followed by the axes labels.

Volume/Pressure Graph

The main plot routine begins at line 2400 and prints the control menu at the top of the screen. This is organised in a similar manner to that for the trend plot where the option required is selected by typing the first letter. The respiratory rate which was calculated at line 315 is displayed and the calibration factors set for the inspiratory flow, SV(1), expiratory flow, SV(2) and pressure, PS.

The data is retrieved from the locations in memory where it was stored by the machine code program and the X and Y values calculated and plotted for inspiration. The values are checked by the subroutine at line 2466 to ensure that none is outwith the acceptable range since the generation of an illegal quantity error at this point would cause the program to stop. The maximum values for pressure and volume are then compared with the scales selected for the axes and if these are incorrect, the plot is redrawn with the appropriate scaling. An example of the display obtained is shown in figure 25. This was recorded when an excessive quantity of water had accumulated in the ventilator tubing and had not been noted by the nurse.

3. TEXT COMMENT

This selection enables any text string to be entered and is frequently used by the nursing staff to note alterations in drug therapy or to indicate when a particular nursing procedure has been carried out. The program ROCHEPRINTEXT is used to obtain a complete printout of all the comments entered within the previous 12 h.

4. CHANGE OUTPUT

The results of the analysis of each individual respiratory cycle and the averaged results can be printed out either to the screen or to a printer connected to slot 1 of the computer. The particular output device required can be selected by this option.

5. CALIBRATE VENTILATOR OUTPUT

The SERVOCALIBRATE program is loaded and executed and the analogue outputs from the Servo 900B ventilator calibrated. Once this has been performed, ROCHESERVO is automatically re-called and run. This option is dis-allowed when ROCHESERVO has been called from the Roche programs

6. TREND DISPLAY

The trend display uses the same subroutine as that used in the Roche cardiovascular programs except that when this option is first selected, all the available data is plotted and there is no choice initially, as there is in the cardiovascular programs, of an expanded plot. Data which was stored from previous analyses has been obtained from disc and place in the array, Z$(X). The variables plotted are :-

1. expired tidal volume
2. peak pressure
3. end expiratory pressure
4. compliance

Once the trend plot has been completed, the variable 'RP' is set to 1 and is used to indicate at lines 340 and 350 that if the program control flags are not altered by the user and the trend plot continues as the operative selection, the graph will not be drawn repeatedly. The available selections from the trend plot are :-

X(pand
T(able
M(enu
V(OL pres

Each choice is again entered by typing the first letter of the option selected. The X(pand facility allows any section of the trend plot to be expanded to fill the screen provided that there are at least 2 min between the first and last point selected. The routine used is the same as that employed in the Roche cardiovascular programs since the data is stored in the same manner. Selection of the T(able, M(enu or V(olume pressure options passes control to the relevant subroutine.

7. ALTER DATA RETRIEVAL TIME

The information from 10 breaths is averaged before being printed out, usually to a hard copy printer, and stored on the floppy disc. The time beween the collection and averaging of the results from each group of 10 breaths is the data retrieval time (DR) and can be set from 1 - 120 min by selecting option 7. The default time is 30 min. Even though the data from respiratory cycles may not be used to produce the averaged values which are stored and printed , the data from these breaths can be displayed on the screen in tabular form or as a volume-pressure graph.

8. COLLECT NEW DATA

Selection of this option allows the information from the next 10 respiratory cycles to be collected and averaged for storage on the disc and printed as hard copy.

9. END PROGRAM

This allows the user to leave the program in an orderly manner since if the program was halted during writing to the disc, non-recoverable errors could result.

DATA STORAGE

When the time since the last storage of information in the patient file exceeds the data retrieval time, a timeout is generated at line 2600 and the flag AVER$ is set to 'Y'. A further series of 10 breaths is then analysed and averaged.

The array D(X) is used to hold the data and the individual elements of the array are increased with each of the 10 breaths which are analysed. The average value is then obtained and the results printed and stored on the disc in the same format as for the cardiovascular programs.

INDEPENDENT OPERATION

ROCHESERVO can be used independently from the cardiovascular programs, and the organisation of the programs is shown in figure 26. Since the ROCHESERVO has not been called from one of the cardiovascular programs, the first file, SHUNTFILE, which is read at line 15, contains no data. This disc read error is detected and the flag ROCHE$ is set to 'N' to indicate that a Roche cardiovascular program has not called ROCHESERVO and that it should operate independently.

The program then attempts to read RESPFILE which would contain the flags and variables which had been set the first time ROCHESERVO was used for a patient. This file is required since when ROCHESERVO is recalled from one of the ancillary programs, such as FILEWRITE, it is necessary not to have to enter the flags and variables again. If RESPFILE contains data, then ROCHESERVO is being recalled from an ancillary program and the flag, FLAG$ is set to 'Y'. If RESPFILE does not contain data, ROCHESERVO is being used for the first time for the particular patient.

The program continues as if it had been called from the Roche cardiovascular program by testing for connection of the Seimens 900B ventilator to the ADC. If the ventilator is not connected, however, an audible and visual warning is given, the user is asked to connect the ventilator and the program is re-run.

When the ventilator is connected satisfactorily, the program passes to the GET PREV DATA routine at line 3000. If the flag FLAG$ has not been set to 'Y', the user is asked to indicate if a printer is attached; otherwise, when FLAG$ equals 'Y', this section is by-passed and the previous data is obtained from the patient's respiratory data file.

Once the data from 10 breaths have been averaged and the information added to the patient file, a test is made of the status of ROCHE$. If it has been set to 'N', the time is noted and the program analyses and displays subsequent respiratory

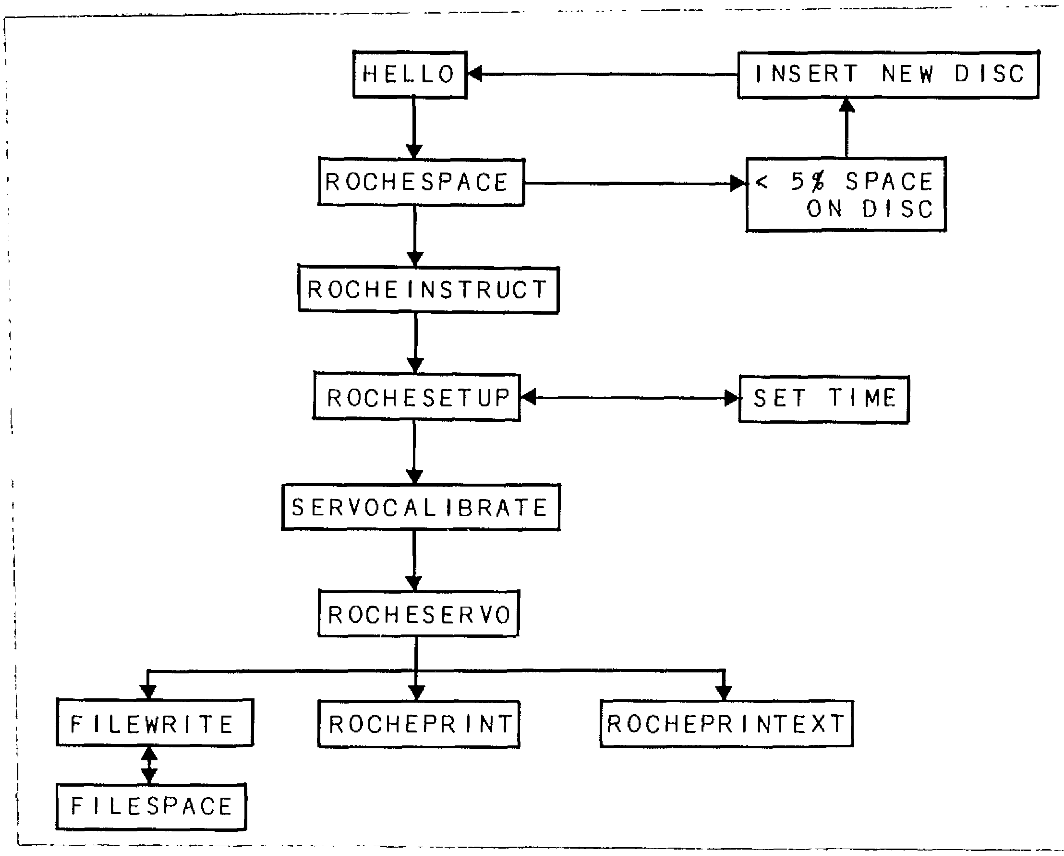

Figure 26 Organisation of ROCHESERVO when used independently of the cardiovascular programs.

cycles until the data retrieval time has elapsed and another series of 10 breaths is collected and the data from them averaged. However,If ROCHE$ has not been set to 'N', the particular Roche cardiovascular program which called ROCHESERVO is obtained from the file NEXTPROG, loaded and run.

SERVOCALIBRATE

This program allows calibration of the outputs for flows from the interface which connects the Servo 900B ventilator to the computer. It can be called from ROCHESETUP, either of the Roche cardiovascular programs or from ROCHESERVO and uses the same machine code data collection routine, RESP.OBJ0, which is used in the ROCHESERVO program. The user is asked to indicate whether a precision 1 litre syringe or a fixed tidal volume will be used to calibrate the outputs. A constant known flow of gas cannot be used since the flow amplifiers used in the Servo ventilator are AC coupled and will only function with a varying flow of gas.

When a fixed tidal volume is employed, this is entered into the computer and the ventilator cycles for 10 breaths to obtain the average outputs for inspiratory and expiratory flow for the tidal volume and calculates the calibration factors. If the precision syringe is employed, the user is asked to connect it to the patient circuit, switch the ventilator to spontaneous mode and move the syringe out and in 10 times. The calibration factors are calculated in the same way as with the fixed tidal volume calibration and stored in the file SERVOVARIABLES to be read by ROCHESERVO. The file NEXTPROG is read and the name of the program to be run next is loaded and executed.

ROCHESERVO MACHINE CODE PROGRAM

RESP.OBJ0

The organisation of this program is shown in figure 27. It collects the values for pressure and corresponding flow from one complete respiratory cycle and stores this information in an array for later retrieval in BASIC. The main program is from line 1 to 36 and the subroutines called from the main program are in the lines following.

When the ventilator is disconnected from the patient, the expiration/inspiration signal remains constant. If the ventilator is collecting data during this time, the space allocated for the array will be exceeded and will begin to overwrite the machine code program. Every time data is stored, the length of the array is checked and compared with the maximum allowable value. If this is exceeded, the number $22 is stored in location $6C96 and control is returned to BASIC where this location is examined and a warning given that the ventilator may be disconnected. This location is therefore set to zero before data collection begins.

Data is stored from location $6C98 indexed by 'Y' and the location is incremented by incrementing register 'Y' up to the maximum of 256. The machine code is self-modifying in that when 256 values have been stored, the high byte '$6C' is incremented to '$6D' by the instruction INC STORE,X at line 51. 'X' is always set to 2 when data is being stored and STORE,X is the location in the program ($797A) of the high byte for data storage. STORE,X is therefore set to '$6C' at the beginning of the program.

Data must be collected from the beginning of the respiratory cycle and two routines are used to ensure that data collection begins at the start of inspiration. The program passes to the routine I-E which calls the subroutine IECHAN. This selects the channel of the ADC connected to the inspiration/ expiration signal and returns the digital value. If the value from the I/E

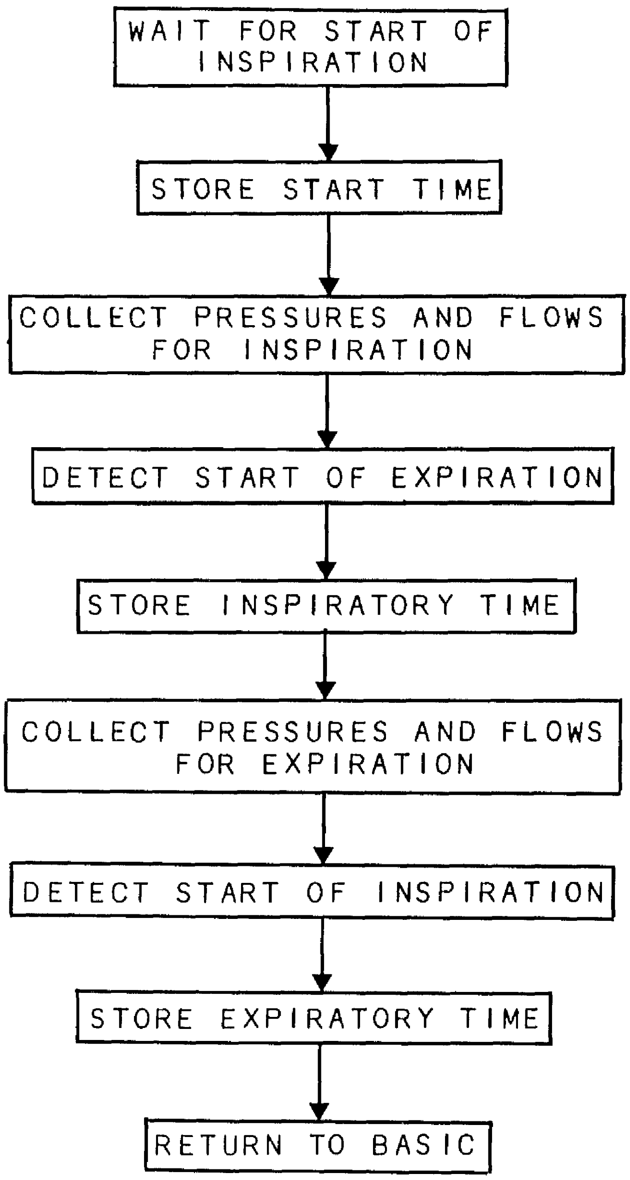

Figure 27 Organisation of RESP.OBJ0.

channel of the ventilator is less than $64, inspiration is progressing and the program remains at the routine I->E until the value of the I/E channel is greater than $64 and expiration has begun.

If expiration is progressing when the program is called, it passes directly through routine I->E to the routine, E->I. The program remains at this routine until expiration has changed to inspiration. The time of the start of inspiration is obtained and stored by passing to the subroutine TIME0, the 'Y' register which is used to increment the low byte of the locations used for data storage is set to zero and data collection begins.

INSPIRN

The inspiratory data is collected by this routine which calls the subroutine GETDATA which arranges the selection of the correct channels from the ADC and stores the data in the array. Register 'A' is returned with the value of the I/E channel and this is checked to ensure that inspiration is still progressing. If this is the case, a further series of data is collected and the check for inspiration made once more.

When the start of expiration is detected, the high byte of the last data storage location is obtained by loading 'A' with the contents of location STORE,X and this value is stored in location $6C80. The value of 'Y' which was used to index the low byte of the data storage location is stored in location $C681. These locations are read when control is returned to BASIC and provide the place in the data array at which the last values for inspiration were stored. The succeeding values in the array are data acquired during expiration. The time at the and of inspiration is collected and stored by subroutine TIME1. Register 'Y' is reloaded with the last value of the low byte used for data storage and the expiratory data is collected.

EXPIRN

This follows the same structure used for the collection of inspiratory data except that the program remains at this routine until the value of the I/E channel becomes less than $64. The high byte of the last data storage location is again obtained from STORE,X and stored in location $6C82 and the value of the 'Y' register is stored in location $6C83. The time of the end of expiration is stored by the subroutine TIME2 and control is returned to BASIC for data analysis.

GETDATA

The subroutine GETDATA selects ADC channel 11 which is connected to the flow signal from the ventilator by loading the 'A' register with $0B. Subroutine ADC is called which obtains and stores the data for that channel. The pressure signal connected to channel 3 is then selected and subroutine ADC called again. Data collection is too rapid using machine code and a wait routine is called twice to produce an acceptable sampling rate. The final task of GETDATA is to obtain the value of the I/E signal and return to the calling routine with this value in the 'A' register.

ADC

The number of the channel of the ADC to be digitisd has been placed in the 'A' register by the calling routine, GETDATA and this value is stored in location $C401. When the ADC occupies slot 4 of the computer, storing a number in this location instructs the ADC to begin the process of converting the analogue signal connected to the selected channel into the appropriate digital value.

When the conversion has been completed, which takes about 70 microseconds, an End Of Conversion (EOC) flag at location $C402 is set. The routine loops until the EOC is signaled when the data can then be read. Once the ADC has been instructed to begin conversion, the EOC flag has to be cleared which takes about 7 microseconds. The program therefore does not examine the EOC flag until a delay of this time has passed. The delay is achieved by pushing the value of 'A' on to the stack and then immediatedly pulling it off again.

When EOC has occurred, the data is read from location $C400 and stored from location $6C98. The high byte of the storage location is modified during the running of the program as explained above. Memory location $797A is STORE,X and initially contains $6C which is the high byte of the location used at the start of data storage. The low byte for data storage is $98 indexed by 'Y' and after each value has been stored, 'Y' is incremented, transferred to the 'A' register and compared with zero to detect when 256 values have been stored. If 'Y' is not equal to zero, the high byte does not need to be incremented and the subroutine returns to GETDATA.

When 'Y' is equal to zero, memory location STORE,X ($798A) is incremented and the value compared at the routine CHECKLEN with $78 which is the maximum permissable for the length of the data array. If STORE,X is less than $78, the subroutine returns to GETDATA but if it is greater, the value $22 is stored in location $6C96 before returning to GETDATA.

WAIT

The 'A' register is loaded with $FF and stored in location $6C97. This location is decremented until equal to zero and then returns to GETDATA.

TIMING ROUTINES

The times of start and end of inspiration and end of expiration are obtained using the Mountain Hardware Clock/Calendar card which is placed in slot 3 of the computer. This timing card makes available the date and time and can be read using BASIC or machine code. The date and time are set using the SET TIME program described under the cardiovascular section. The subroutines for obtaining the timings for ROCHE/SERVO are from line 90 to the end of the program ROCHE.OBJ0.

Certain locations are used by the clock. Therefore, the contents of location $45 are loaded into the 'A' register and all the 6502 registers together with the contents of locations $38 and $39 are saved on the stack. The card is located in slot 3 of the computer and is accessed by the commands :-

```
LDA  #$C3
STA  $39
JSR  $C300
```

This series of commands instructs the ROM on the card to obtain the date and time and load the data into temporary storage in memory. The locations and registers saved previously are restored and the data read from the temporary store into locations which are subsequently read from BASIC.

The time information is stored in the form of ASC II characters and six characters are acquired for each time recorded; units and tens of seconds, a full stop and units, tens and hundreds of milliseconds. These are temporarily stored by the clock in locations $282-$287 and are relocated at subroutine LOADCLK into locations 27786-27791 for TIME0 which is the start of inspiration, the next six locations for TIME1 which is the end of inspiration and the next six for TIME2 which is the end of expiration. These locations are read in BASIC and the times obtained for the required calculations.

SECTION 3

DATA ANALYSIS

DATA ANALYSIS

FILE SUITE

A computer system must be of clinical value to the medical and nursing staff who are using it at the bedside or in the operating theatre. However, one of the principal benefits of such a system is that data can be collected automatically or entered manually during the normal ward routine and stored in a form which enables subsequent rapid and simple analysis. The graphics routines which are used in the Roche suite of programs were incorporated into a separate program called FILEPLOT which allows data collected by the Roche programs to be displayed graphically at a later time.

Frequently, so much data is collected from a patient that it becomes difficult to detect accurately trends in the variables measured. FILEPLOT was therefore extended into a suite of programs which allow data obtained off-line or on-line from patients to be entered into the computer, stored and displayed. The suite consists of 7 separate programs which can call each other in a similar manner to those forming the Roche suite.

The principal difference between the linking of the individual programs in this FILE suite and those in the Roche suite, is that the variables are retained intact in memory and most programs call each other with the CHAIN command. Since routines which involve PAGE 2 of graphics are used, the lowest location available to programs for data storage is once more, 27800. LOMEM must be set by the first program which initiates the CHAIN sequence since, if LOMEM is reset by a subsequent program, the values of the variables are lost. An error is also generated if the variables are redimensioned in later programs, and all dimension statements must be included in the first program.

HELLO

The greetings program is automatically run when power is supplied to the computer. It sets LOMEM to 27800 and dimensions the string arrays :- SM$(X), IN$(X), Z$(X) and the array, V(X). Instructions can be obtained which indicate the mode of operation of the programs and explain the method used to direct the flow of the programs (figs. 28 and 29). If instructions are requested, the user is also required to complete certain examples where data has to be entered into the system. These simulate the method of data entry used in the main program to provide practice and instill confidence in a new user. When these have been entered correctly, FILEPART1 is CHAINED into memory.

FILEPART1

This program allows selection of a patient's name from a list displayed on the screen. The individual names are obtained from a disc file called NAMELIST. After selection of the appropriate name, the type of data, such as cardiovascular or respiratory, is chosen for retrieval and the patient's data for that system is obtained from the disc and stored in memory. The data may be displayed in numerical form on the screen where it can be corrected or updated; it can also be printed out as hard copy or displayed as a trend graph. Where the graphical display is requested, the data is retained in memory and the next program FILEPLOT is chained using the commands at line 150.

The program begins by allocating to an array R(X) TAB positions which will be used for the screen display. The programs are all controlled with the GET command where a single character is obtained from the keyboard and then examined to ensure its validity. The effect is to lock out all incorrect characters from the keyboard. The keyboard buffer is therefore cleared of all characters and a delay of 0.5 s introduced to prevent the confusion of a double entry from the keyboard controlling program flow.

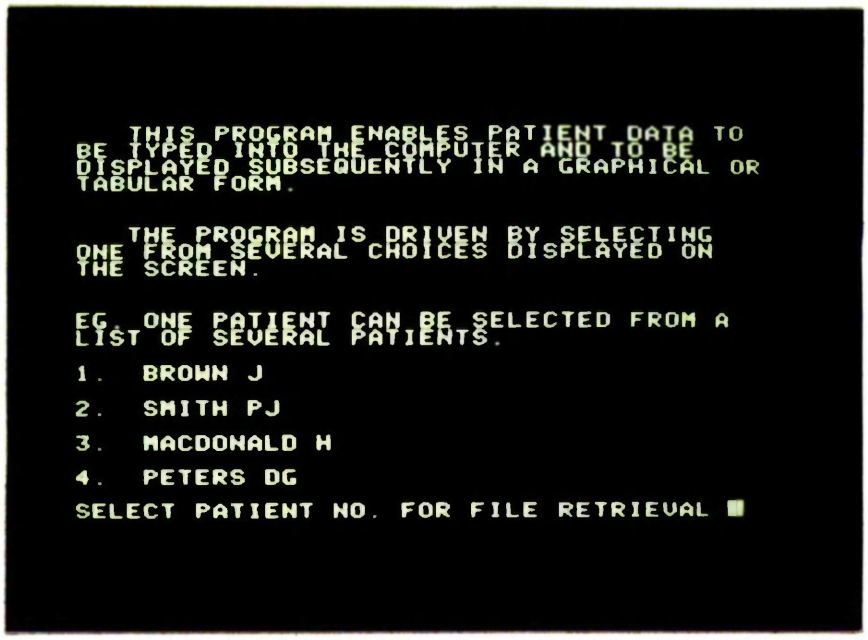

Figure 28 Explanation of function of FILE SUITE.

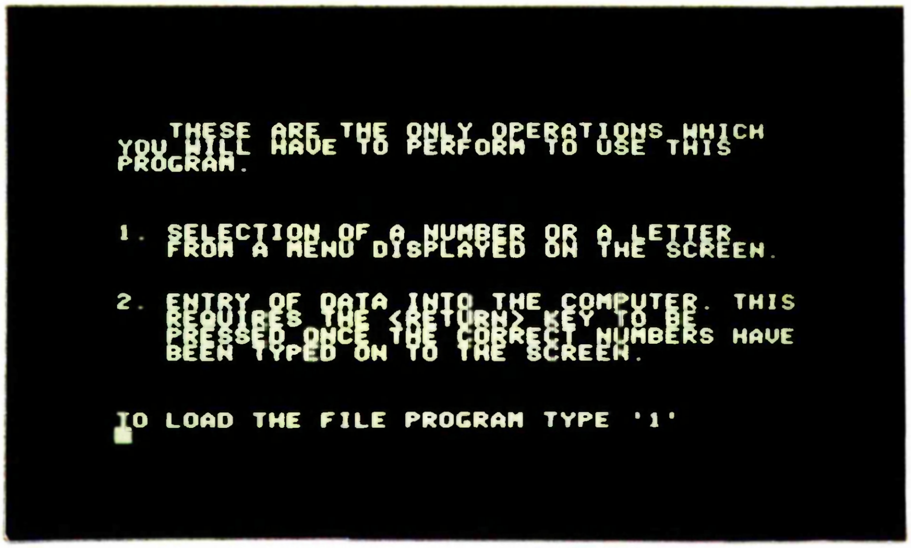

Figure 29 Instructions displayed for FILE SUITE.

The function, TIME, is defined as in the Roche programs to enable the date/time identifying characters to be easily extracted from the data strings. The data strings are placed in the array Z$(X) when patient data is obtained from the disc files in the standard manner. If FILEPART1 is called directly from the HELLO greetings program, the variable, X$ has no value. When the data display subroutine is called at line 20, the null string is detected and control is immediately returned to the routine MENU at line 100.

MENU

This is the central menu of FILEPART1 from which the major selections are made. The options which are available to the user depend on how far he has progressed through the program. If no patient name has been selected, the only possible options are to list the patient names on file and to add or alter a name contained in the file. When a patient and a system for data recall have been selected, further options become available. The complete list displayed is :-

1. LIST NAMES ON FILE
2. ADD / ALTER NAMES ON FILE
3. DISPLAY / ENTER DATA
4. TREND PLOT
5. PRINT OUT RESULTS
6. SELECT DIFFERENT SYSTEM

1. LIST NAMES

All routines which involve obtaining data from disc files have error traps incorporated into them to prevent the non-recoverable error which occurs when an empty disc file is read. Each of these sections therefore begins with an ON ERR command which instructs the program to procede to a given line number when an error is detected. A comment that no data is available is displayed to the user and the program returns to the main menu at line 100. The ON ERR command re-directs the program to a handling routine when an error is detected. However, it cannot be used in a subroutine. GOTO instructions have to be used to direct the program to the data collection routines and so permit automatic trapping of disc read errors.

The names of patients whose data has been stored on the disc are stored in the file, NAMELIST and the number of entries in this file is contained in NAMELISTLEN. The name are obtained and displayed on the screen accompanied by their file number. When the names are entered into the computer, they are automatically sorted into alphabetical order. The required name is chosen by entering the file number and this is displayed along with the hospital number and the list of possible data which can be retrieved from disc.

SYSTEM LIST

The patient data is stored in exactly the same manner as for the Roche programs in that all the individual entries for one period of data collection are concatenated together into a single string of characters. Each individual item can be extracted from the string simply by specifying the start point in the string and the number of characters used for the individual item.

The routines employed for screen display, print out of hard copy and graphics plot are therefore completely independent of the type of data involved. The types of data which can be entered and displayed at present are:-

1. CARDIOVASCULAR
2. RESPIRATORY
3. FLUID INPUT
4. FLUID OUTPUT
5. FLUID BALANCE
6. BLOOD GASES
7. UREA & ELECTROLYES
8. LIVER FUNCTION TEST
9. HAEMATOLOGY
T. TEXT COMMENTS

The variable X$ is set equal to an abbreviation of the system selected, such as 'LFT'S' or 'CVS', and is used, along with the patient name, to attempt to retrieve data from that patient's files by passing to the routine at line 3000. The user may also return to the previous menu by typing the RETURN key.

RETRIEVE DATA

This routine uses the standard method employed in the Roche programs to obtain the number of patient files stored from a separate DATALEN file and to retrieve patient information from a DATA file. If no information has been recorded in the files, an error is detected and the user informed before the program returns to the MENU routine where the complete menu is displayed on the screen.

2. ADD / ALTER NAME ON FILE

Additional patients can be entered into the computer or patient details altered by this selection which CHAINS another program into memory. The ADDNAME program can either be called by FILEPART1 or by the graphics equivalent, FILEPLOT.

The commands used to CHAIN two programs together are :-

```
PRINT D$;"BLOAD CHAIN,A520"
CALL 520"FILEPLOT"
```

The Roche suite of programs had written the calling program to which the subsidiary program should return, as a string variable into a separate file, NEXTPROG. The calling program was then obtained from NEXTPROG by the subsidiary program and the original program was re-run. However, it proved impossible to use a string variable for the name after the command, CALL 520, and the required name had to be printed in full. Two forms of the program ADDNAME are thefore provided; one called by FILEPART1 and one called by FILEPLOT.

ADDNAME contains the same routines as the ADDNAME subroutine described in ROCHESETUP for the entry of patient identification. In addition, it permits patient identification information to be altered or deleted from the file, NAMELIST, in which it is stored. Once all the corrections and additions have been made, the program NAMESORT is called which obtains the names from NAMELIST, arranges them in alphabetical order, and re-writes them on to the disc. Two separate forms of the program NAMESORT are also required; one to return to FILEPART1 and one to return to FILEPLOT.

3. DISPLAY / ENTER DATA

This is the central routine of FILEPART1 and permits the data contained in any of the different system files to be displayed on the screen, or further data added to the file. All data, except text comments, can be corrected if required. The routine consists of three main procedures :-

(a) DISPLAY DATA
(b) CORRECT DATA
(c) ENTER DATA

(a) DISPLAY DATA

A different number of data, each of varying length, are stored for each system. For example, the files for HAEMATOLOGY results contain the following separate data :-

Date/time : Hb : PCV : WBC : Platelets

The subroutine developed to display the information uses certain variables to indicate how the strings of patient data should be split into the different individual components and how these should be displayed on the screen. The variables are :-

NV = number of different values which make up the string. The haematology files have 5 values.

S(X) = position in the string at which an individual value begins. There are NV values in a data string and the start position of each is S(1)....S(NV).

L(X) = number of characters which make up the individual value. The number of characters in each value is L(1)....L(NV).

N1 = number of values to be displayed in the top part of the screen.

N2 = number of values displayed in the bottom part of the screen.

PN = number of data strings which have to be disassembled and displayed on the screen.

Lines 5010 - 5060 allocate the relevant values to the S(X) and L(X) arrays depending on the system selected by the user. The headings which are used to label the values displayed are also printed at these lines. If six or less individual values are involved in a system, such as HAEMATOLOGY, then all can be displayed in one row across the screen. The variable N2 is set to zero and PN set to 12. Up to twelve separate blocks of patient data can be printed on the screen. If more than six values have to be displayed, N2 will be greater than zero and PN is set to 6. A maximum of six separate blocks of data can then be displayed on the screen at one time.

The individual values, such as Hb and WBC, are extracted from the string by the routine at line 5090 and placed in the array SM$(X). The first value is the date/time string and is converted to the real date and time by the subroutines at lines 25000 and 26000. The top row of values from the concatenated data string is printed and, if N2 does not equal zero, the bottom row is displayed along with suitable headings.

The TAB positions are set by assigning a series of predetermined numbers to the array R(X) at the beginning of the program and these are used in the subroutine at line 5490 to position the individual values on the screen. The resulting screen display for haematology is shown in figure 30 and for urea and electrolyte results, in figure 31.

(b) CORRECT VALUES

After all the required data strings have been printed on the screen, the user is given the chance to correct the information displayed on the screen. If he selects to make a correction, the program passes to the CORRECT DATA routine at line 5135 and a flashing number is printed alongside each row of data. Typing one of these numbers causes the program to ask the user whether he wishes to delete the row of data selected entirely from the file, alter the data or leave it intact.

If data are to be altered, the program passes to the standard UPDATE DATA routine at line 5260 after a flag, ALTER$ has been set to 'Y'. When all the individual values have been entered at this routine and concatenated into one large data string, the ALTER$ flag is used to indicate that the number of data files should not be incremented since a replacment and not an additional file has been entered.

The entire block of patient files for the particular system is re-written to the disc. When the user elects to delete data,

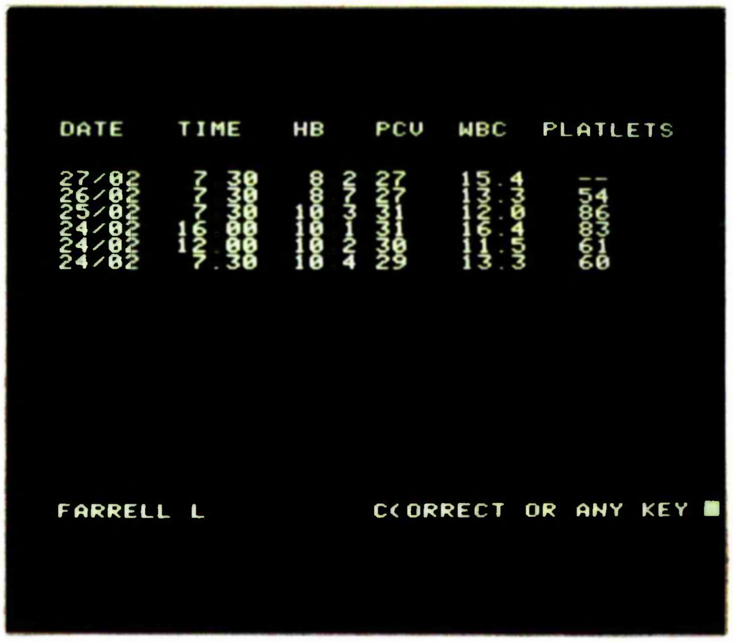

Figure 30 Tabular screen display of haematology results.

Figure 31 Tabular screen display of results for urea and electrolyte estimations.

the data string which has been selected is removed and all the necessary files are re-positioned. Once again, all the data for that system are re-written to the disc.

(c) UPDATE DATA

Data are entered into the computer using the routine at line 5250. This displays the form in which the date and time should be entered and prints an explanatory prompt line at the bottom of the screen. The actual entry of data is controlled by the subroutine at line 5330 and every character entered is obtained from the keyboard with the GET command. This allows the characters to be tested to ensure that they are within allowable ranges. For example, if a letter is entered, this is rejected and the cursor does not move to the next position but remains to obtain a further entry. If the date and time are not entered in the required manner, the user is reminded of the necessary form and the data must be re-entered.

When the RETURN key is typed, this is taken to indicate that a complete entry of an individual number has been made. If a left arrow is typed, all the characters for that particular number are deleted, the cursor moved back, and the string cleared for entry of new characters.

The individual characters entered for one complete number, which is at the Xth position in data string, are concatenated together at line 5344 and the length checked against L(X) to ensure that the string of characters formed has the same number of characters as was set at the subroutines from line 5010.

Once all the individual values for the system have been entered, the user is asked to confirm that the data are correct. If this is not true, he must re-enter them once more. However, if the values are all correct, they are concatenated into the complete data string for that date/time identification.

After the entry of the data, the user is asked if he wishes to add a free text comment. This could indicate that the last results for fluid output were obtained following the administration of a diuretic to the patient or that the patient was receiving dopamine when the cardiovascular results were entered. Any text string is added to the concatenated data string and stored in the disc file at line 2000. Control is returned then to the MENU routine at line 100.

When the fluid balance option is chosen, the program calls the subroutine CACULATE FLUID BAL at line 12000. This routine collects the fluid input data from the patient disc file and prompts the user to indicate the period of time over which the balances should be calculated. The standard routines at line 13001 used in the Roche programs are employed to obtain the first and last data strings for this period of time.

The individual input values for each entry, such as fluids administered by naso-gastric tube and intravenous infusion, are added together and loaded into an array, IT(X). Information is obtained for the output data and the individual totals for the selected time period are loaded into an array, OT(X). The total input values, output values and balances are then displayed on the screen (fig. 32). The calculated information can be printed out for a permanent record, a further series of balances can be calculated, or control can be returned to the menu.

TEXT COMMENTS

This routine was included because it has proved one of the most useful in the ROCHE suite. The selection of the TEXT entry option is detected at line 5000 and the program is directed to the routine at line 4000. Any previous results are displayed on the screen and the user is prompted to enter the latest comment (fig. 33).

When the free text has been entered, the date and time

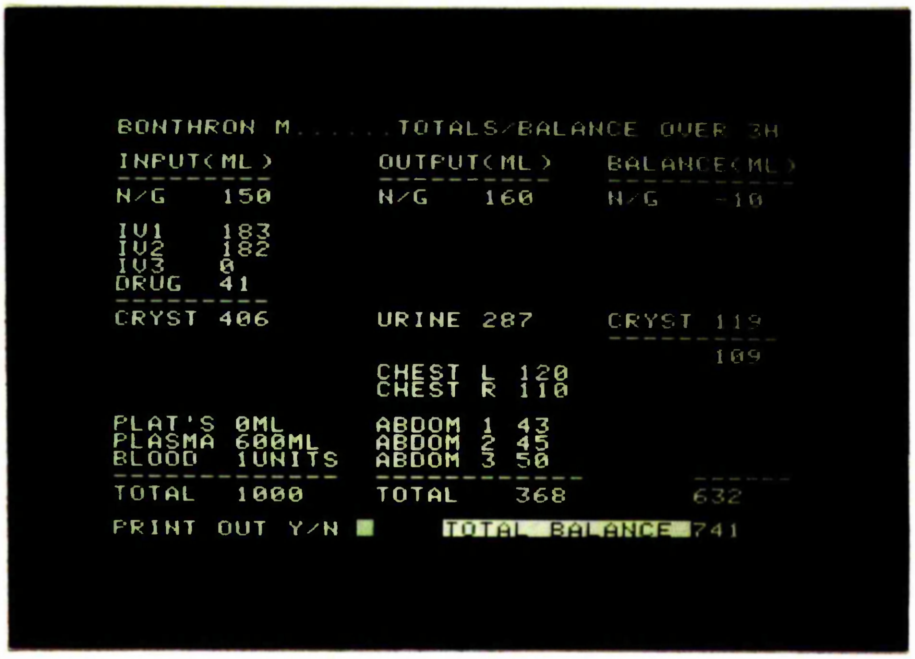

Figure 32 Screen display of fluid balance calculations.

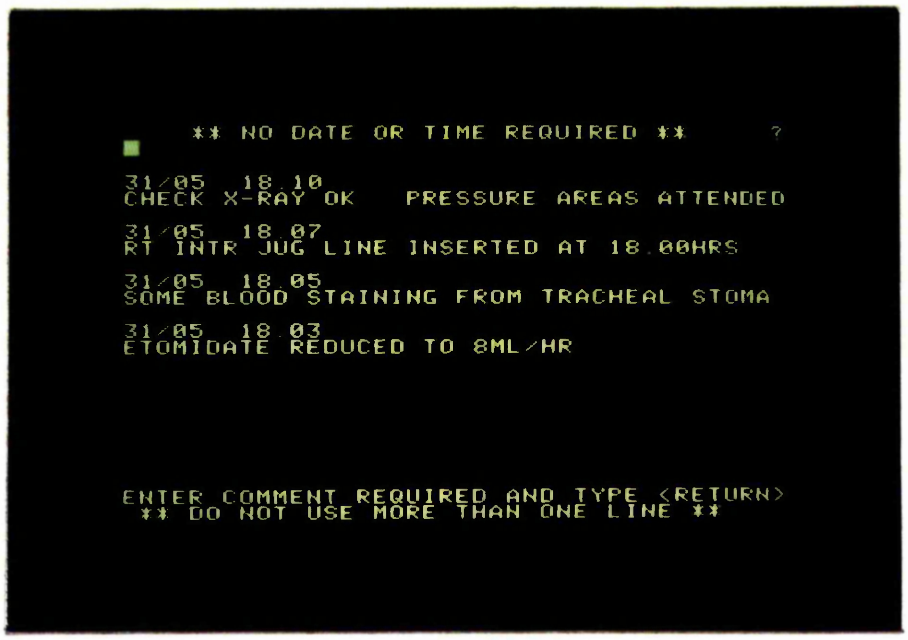

Figure 33 Entry of free text comments and display of previous entries.

identification is obtained at line 27000 from the Mountain Hardware clock/calendar card to avoid typing in the date and time manually. This routine could be altered to manual entry if the card was not available.

4. TREND PLOT

When option 4 is selected, the graphical display program FILEPLOT is CHAINed into the position in memory previously occupied by FILEPART1. Data is retained intact and plotted as a trend graph. FILEPLOT is described below.

5. PRINT OUT RESULTS

The program passes to the subroutine PRINT ROUTINE when this selection is made. The purpose of this subroutine is to enable all the results which have been entered for any system to be printed out in reverse order of entry. The routines used are the same as for printing out to the screen except that all the data for each date/time entry are printed out in a complete row. The TAB positions are obtained from the subroutine at line 5000 before calling the PRINT ROUTINE at line 6000. Any text strings which have been entered along with numerical data are printed below the data to which they refer.

FILEPLOT

The purpose of this program, is to display graphically data which have been entered manually using FILEPART1. Originally, both these routines were incorporated into the same program. However, as development progressed and more features were incorporated, the end of the program became overwritten by the graphics page. This caused peculiar line numbers and commands to be incorporated into the end of the program and data statements which are needed to label the date scale of the X-axis became corrupted. These corruptions could not be removed using normal techniques and the only method which proved satisfactory was to list the program lines which were satisfactory to a text file on disc. The EXEC command was then used to reload the text file back into memory as a program.

Patient names and the system which is to be displayed can be selected using identical routines to those in FILEPART1. The values to be plotted are extracted from the strings in the same way as for the graphics display in the Roche suite by specifying the start position in the string and the number of characters which make up each value.

The X-axis is also obtained in the same manner as in the Roche programs but a variation has been incorporated into the plot routines for liver function tests, respiratory and cardiovascular data. The enzyme concentrations are plotted on a log scale and the Y values are calculated from the concentrations at lines 13305 and 13325. A flag, LG$, is set to 'Y' to indicate that the log calculation for the Y values is to be used.

The maximum value of the Y-axis when respiratory or cardiovascular data are plotted, is allocated to the variable TS and the minimum value to BS. The value of the data point to be plotted is entered into the equations at lines 13307 and 13327 and the result provides the appropriate Y value for plotting on the graphics screen.

Selecting option 3 or 5 replaces FILEPLOT with FILEPART1 using the CHAIN command. The data which remains unaltered in memory can then be displayed, corrected, further information added or printed out as hard copy.

The FILE suite of programs enable data to be entered into the computer system by the intensive care unit staff in the course of their normal work and are under evaluation to replace the standard record sheets. The cumulated results of laboratory investigations are available for examination as a list of numbers displayed on the screen or can be printed out for a permanent record. Running totals and balances of fluids can be obtained at any time and over any desired period. FILEPLOT produces trend graphs of the data with the approximate normal limits displayed.

Figure 34 illustrates the trend graphs obtained from the haematology results printed in figure 30. Figures 35 and 36 shown the graphs obtained from the urea and electrolyte results displayed in figure 31. The trend graphs can be more easily assimilated than the raw figures, which is a desirable feature in a clinical situation where the significance of complex data has to be appreciated rapidly.

Extracting information from standard record sheets to compile discharge summaries for patients leaving the unit can be a prolonged and tedious undertaking. When data has been stored on a magnetic disc after automatic or manual collection on a magnetic disc, the required information can be displayed to the user in a few seconds with the FILE suite of programs.

When presenting information as a trend plot, it is frequently essential to indicate particular features. This can be achieved with any trend display by typing 'CTRL L'. A small arrow appears at the bottom of the screen pointing to the second data value plotted (fig. 37). When the 'F' key is pressed, the arrow moves forward to the next value. Typing the 'M' key identifies the position as one which is to be marked. A maximum of 10 marks can be made for any graph. When the final value plotted has been passed, the X-axis is re-plotted and the markers identified with lower case letters placed at the selected positions. Complete graphs can also be printed out on a Silentype printer by pressing 'CTRL P'.

More complex analyses of data can also be undertaken. Several programs were commercially available which had some of the features needed for these analyses but they required alteration to be used with data collected by the Roche or FILE suites of programs.

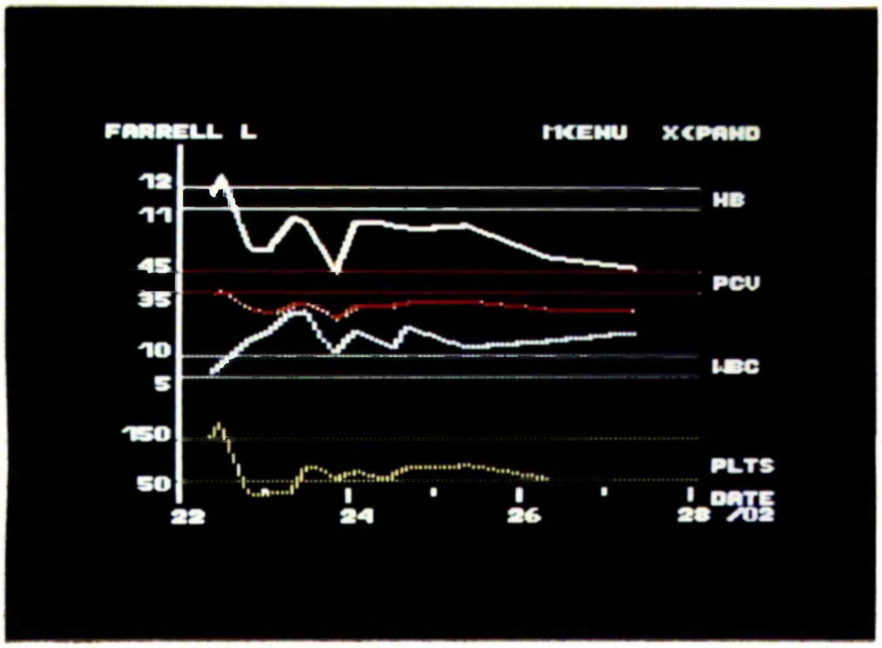

Figure 34 Trend graph of haematology results displayed in tabular form in figure 30.

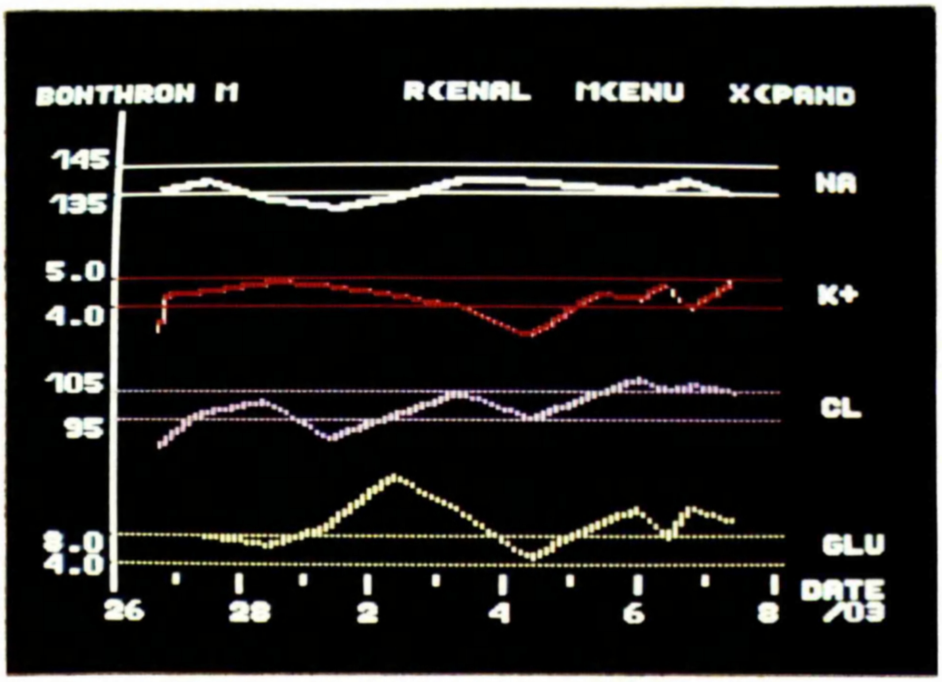

Figure 35 Trend graph of results displayed in figure 31. (part1).

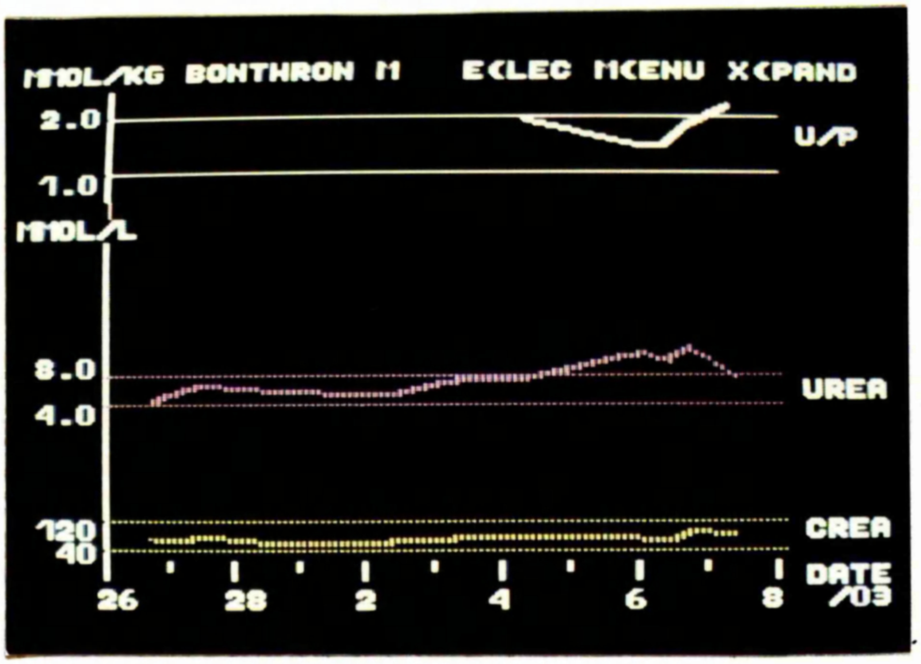

Figure 36 Trend graph of results displayed in figure 31.
(part2).

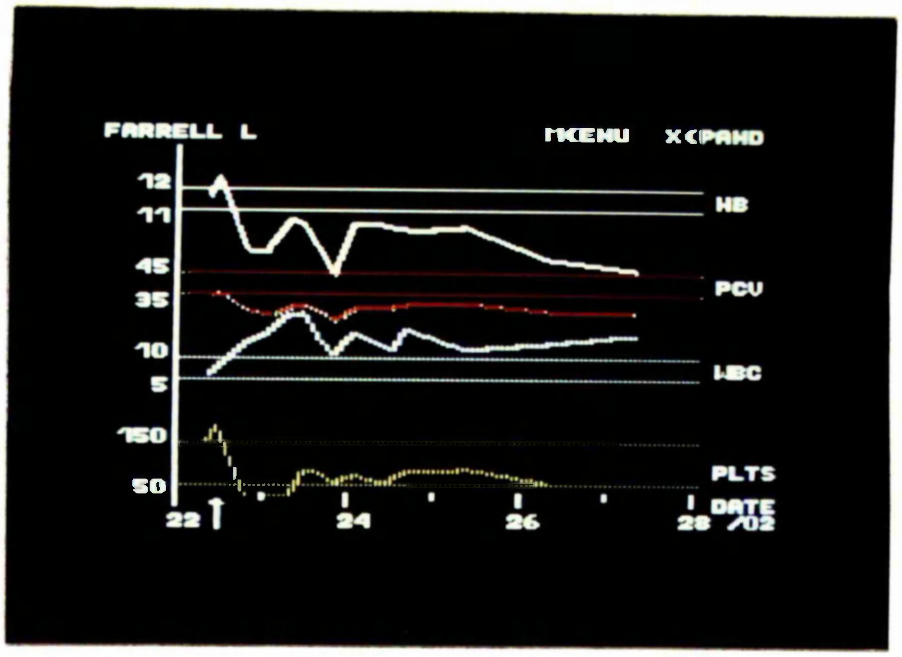

Figure 37 Marker arrow used to label features of special
interest.

APPLEPLOT is a program produced by the Apple computer company which allows the user to plot data in a variety of different modes. However, the method of data entry and editing was not satisfactory and was modified. Strathand Ltd. produce a statistical analysis program which has a sophisticated data entry and editing facility and a data analysis suite has been developed based on these two commercially available programs.

ABGSTAT

This program was developed principally for the analysis of data collected from patients who had undergone operations for insertion of aortic bifurcation grafts. These patients were monitored using one of the Roche systems and the cardiovascular data collected on-line . The particular aim of the study was to determine the value of the PAWP compared with the standard CVP measurement in assessing the circulating blood volume and to attempt to identify cardiovascular changes intra- and postoperatively.

ABGSTAT prompts the user to enter the name of the patient whose data is to be analysed. As explained previously, when 150 separate blocks of patient data have been collected, the program FILEWRITE automatically removes the first 50 and writes them into a separate file. ABGSTAT tests for the existence of these extra cardiovascular files and, if they are present, loads the data contained within them into the computer memory. The data from the main cardiovscular file is then obtained and the user asked to indicate at the routine SELECT ITEMS the correlations which he wishes to make. Selection of a pair of items for correlation assigns variables as follows :-

S1 = start position in the data strings of the first item
L1 = number of characters in the first item
N1$ = name of the first item, such as 'CVP'

The second item for correlation uses the same variables indexed by '2' :- S2, L2, N2$.

The pair of items selected for correlation are extracted from the data strings at the routine SET UP ARRAY and stored in the array X(Y,0) and X(Y,1) for entry into the analysis routines. If the CVP or PAWP was unavailable during collection of the cardiovascular information, the character string '-99' was stored at the locations in the data string occupied by these variables. A test is made at line 305 to detect if either of the pair of items obtained from the data string has a value of -99. If so, that pair is eliminated from the analysis routine. A further test is made if neither of the pair is CVP or PAWP, for the value of zero, and if this is true, that pair is also eliminated.

The pairs of satisfactory data can then be examined, printed out as hard copy, extra values added or inaccurate values deleted using the subroutine AMMEND DATA from line 3000. This editing subroutine has been taken from the Strathand statistical program. The correlation coefficient and the regression line are calculated and displayed on the screen. The data collected on-line can be saved on a disc containing the complete Strathand statistics program since this has an efficient editor subroutine which can be used if further editing of the data is required.

The data can also be saved on a disc in the form required for use by the APPLEPLOT program. The routine at line 400 saves the individual pairs of data which are used by APPLEPLOT to produce a scatter plot. Two pairs of data are also stored which are calculated using the equation for the regression line and allow the regression line to be drawn superimposed on the scatter plot.

The labels for the X and Y axes are N1$ and N2$ and these are stored on the disc along with the title of the graph, which is the patient name, N1$, N2$ and the correlation coefficient. APPLEPLOT allows scaling of the axes and the number of divisions required to be selected easily. Further data from the patient can be analysed or APPLEPLOT can be run for final display of the complete analysis.

Data which has been entered manually using the Strathand statistics program can also be organised in the required form for use by APPLEPLOT using the program STAT>PLOT. This obtains data entered with the statistics program and prompts the user for the X and Y axes labels. The correlation coefficient and regression line are calculated, the user instructed to insert the APPLEPLOT disc into the drive and the data is stored in the required sequence.

Frequently, a permanent record of graphical data is required and can be obtained using the program, HP PLOT. This was developed to allow data which had been graphed using APPLEPLOT to be plotted on a Hewlett Packard 7470A plotter.

HP PLOT

The program can accept up to 400 pairs of X and Y coordinates and these are obtained from a disc file compiled using the Strathand statistics program and STAT>PLOT.

The range of values used to label the axes and the number of divisions required for each axis are selected. The scales can be compressed to place several on a single sheet of paper or overhead projector transparency and one of two pen colours may be selected. The points plotted can be joined together with a line, or drawn as a scatter plot with the regression line superimposed.

The plotter is initialised at the routine from line 1000 and proceeds to draw and scale the axes at the subroutine from line 3000. A cross is drawn to indicate each point plotted and the text labels for each axis and graph are then printed. Further data can be graphed or the program can be terminated.

The entire collection of analysis programs and their functions are :-

1. FILEPART1

This allows the direct entry of data using the keyboard. Data can also be collected from files stored from on-line collection of patient information. The results can be viewed, corrected or printed out as hard copy.

2. FILEPLOT

The same patient information used in FILEPART1 is used by FILEPLOT to construct trend graphs. The graphs can be expanded to examine one particular section as long as there are at least 2min between the first and last values to be plotted. Markers can be inserted to point out certain features and the graphs can be printed for hard copy.

3. ABGSTAT

Cardiovascular data collected on or off-line can be selected for correlation and the results stored in a suitable form to allow a graph to be produced using the commercial program, APPLEPLOT or HP PLOT. Labels for the graph and axes are automatically incorporated into the data which is stored for APPLEPLOT.

4. STAT>PLOT

Data entered manually using the Strathand statistical program can be re-organised in a suitable form for use with APPLEPLOT or HP PLOT. This also automatically produces labels for the graph and axes.

5. HP PLOT

Data written by ABGSTAT or STAT>PLOT can be plotted with a Hewlett Packard 7470A X-Y plotter for a permanent record.

SECTION 4

CASE REPORTS

The application of the data collection and analyses programs in clinical use can be illustrated by three case reports.

CASE 1

A 15 year old girl was admitted to the acute medical ward suffering from severe respiratory failure. She had undergone an uneventful tonsillectomy 5 days previously and had been discharged home 3 days before admission to the medical ward. Her pulmonary function continued to deteriorate and she was admitted to the respiratory intensive care unit to receive intermittent positive pressure ventilation (IPPV).

Arterial PO2 was 65 mmHg and PCO2 was 65 mmHg while receiving IPPV with 100% oxygen. Her systolic arterial pressure was less than 90mmHg and she received dopamine by infusion. Her condition continued to deteriorate until her arterial PO2 was 42 mmHg and PCO2 was 68 mmHg.

An Oximetrix pulmonary artery catheter was inserted and the outputs from the Oximetrix analyser and the cardiovascular monitor were connected to the computer system. Data were recorded by the computer approximately every 2 min, printed out at the bedside, and stored on the floppy disc.

A comparison of results for mixed venous oxygen saturation (SvO2) obtained with the Oximetrix catheter and an IL co-oximeter is shown in figure 38. This graph was obtained using the Strathand statistical program to enter the results for the measurements of venous oxygen saturation from both analysers. STAT>PLOT configured the data for HP PLOT which was used to produce the final graph with a Hewlett Packard plotter. The Oximetrix analyser was found to provide a valid indication of SvO2.

Systolic arterial pressure and SvO2, recorded over the first 3 h were graphed using FILEPLOT (systolic pressure in green; SvO2 in white; fig. 39). FILEPLOT was modified since the two graphs are not normally displayed together. The increase in SvO2 at

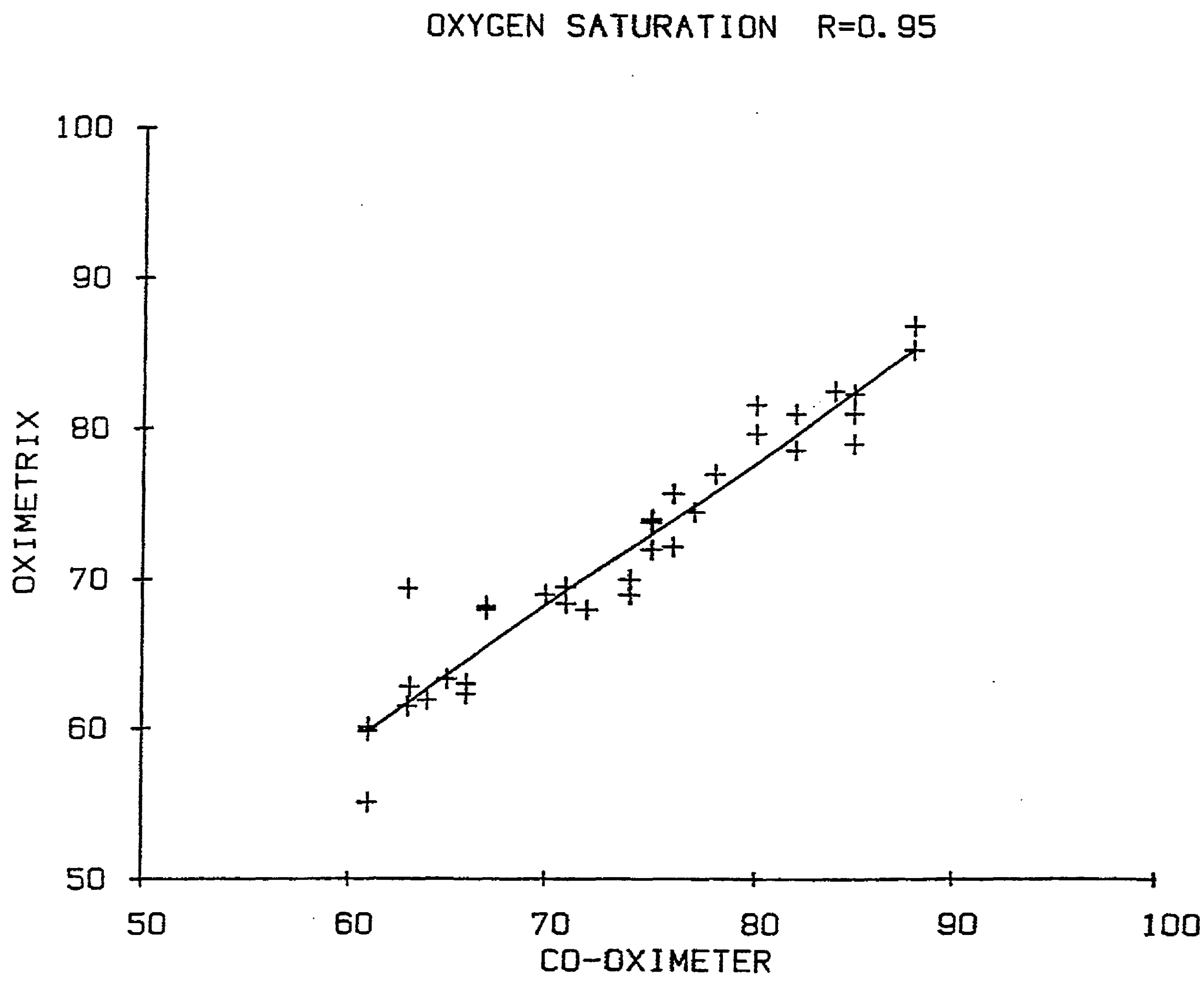

Figure 38 Comparison of SvO2: Oximetrix/co-oximeter.

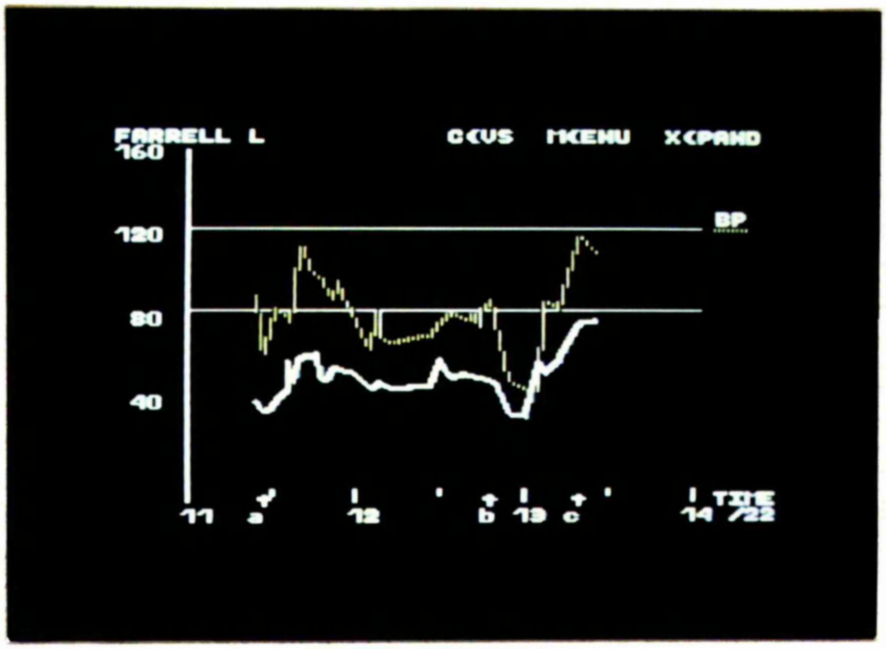

Figure 39 Trend display of systolic arterial pressure (green) and SvO2 (white).

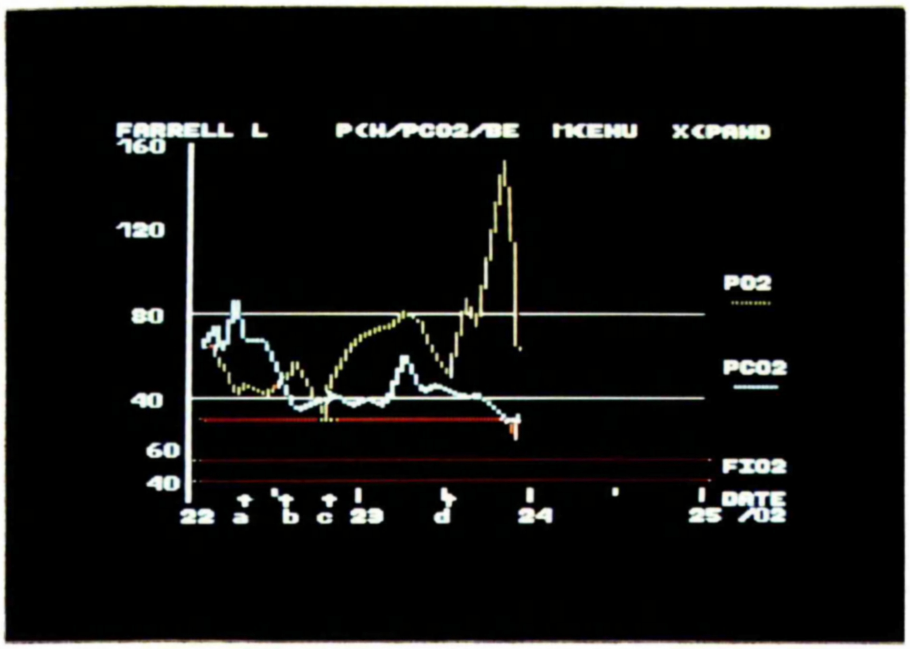

Figure 40 Arterial PO2, PCO2 and FIO2 displayed using FILEPLOT.

point 'a' occurred when the positive end expiratory pressure was increased to 15 cm H2O. Entry of the results of arterial and venous blood gas analyses into the computer system showed the intrapulmonary shunt to be 60% at this time.

SvO2 and arterial PO2 continued to decline and the hypercarbia could not be controlled by increasing minute ventilation. All the measures normally available to treat acute respiratory failure had been attempted without lasting success. The decision was then made to begin extracorporeal membrane oxygenation (ECMO).

ECMO was instituted at 12.55 h, indicated at point 'b' in figure 39. During insertion of the by-pass cannulae, the systolic arterial pressure and SvO2 decreased markedly, but once the full flow of 1 500 ml/min was established, the results improved considerably (point 'c', fig. 39).

Results of arterial blood gas measurements were entered into the computer using FILEPART1 and the values obtained from admission to the intensive care unit until midnight on the 23rd are shown in figure 40. At point 'a', a slight improvement occurred when 15 cm H2O PEEP was introduced. However, a much greater improvement resulted from ECMO and is shown at point 'b' in figure 40.

Initially, a bubble oxygenator had to be used in the circuit which would have caused excessive destruction of blood cells, and a membrane oxygenator was substituted at 18.00 h. The circuit was only disconnected for a few minutes but the reliance of the patient on the ECMO is demonstrated by the considerable reduction in arterial pressure and venous saturation at point 'c' in figure 40.

Pneomococci were cultured from tracheal aspirate and the appropriate antibiotic was commenced. By-pass flow was reduced to 200 ml/min at 13.30 h the following day but although arterial PCO2 was well maintained, PO2 and SvO2 decreased markedly (point 'd', fig. 40). A further attempt was made to withdraw ECMO at midnight on the 23rd but without success. Forty eight hours after beginning ECMO, the lungs had improved to the extent that by-pass could be discontinued.

The results calculated for intrapulmonary shunt were entered using the Strathand program and the results plotted with the Hewlett Packard plotter HP PLOT. The improvement in intrapulmonary shunt over the period of ECMO is shown in figure 41.

The lungs had been ventilated using a Seimens Servo 900B ventilator but at the time of the patient's admission it was not possible to measure SvO2 and pulmonary mechanics simultaneously. When the arterial PO2 appeared to be stable, the ventilator was connected to the computer and serial measurements were made on-line. Changes in pulmonary compliance, total respiratory work and PEEP are shown in figure 42. These demonstrate the alterations which would be expected from successful resolution of the severe pneumonia.

Unfortunately, the patient developed signs of cerebral damage and although the pulmonary lesion was resolving, a diagnosis of brain death was made and therapy was discontinued. Post-mortem examination revealed a gross intraventricular haemorrhage. Haematology results are plotted in figure 43 and indicate that following ECMO, the concentration of platelets decreased to 17 000 /ml. However, there was no indication of any haemorrhagic event at this time. Platelets were transfused to maintain an apparently satisfactory concentration therafter and heparin therapy was controlled by maintaining the activated clotting time at about twice normal.

Intracerebral haemorrhage was reported in a British study of ECMO (96) and the American ECMO study (55) reported haemorrhage as a cause of death in approximately 25% of patients. It would appear therefore that finer control of coagulation is required for successful use of this form of therapy.

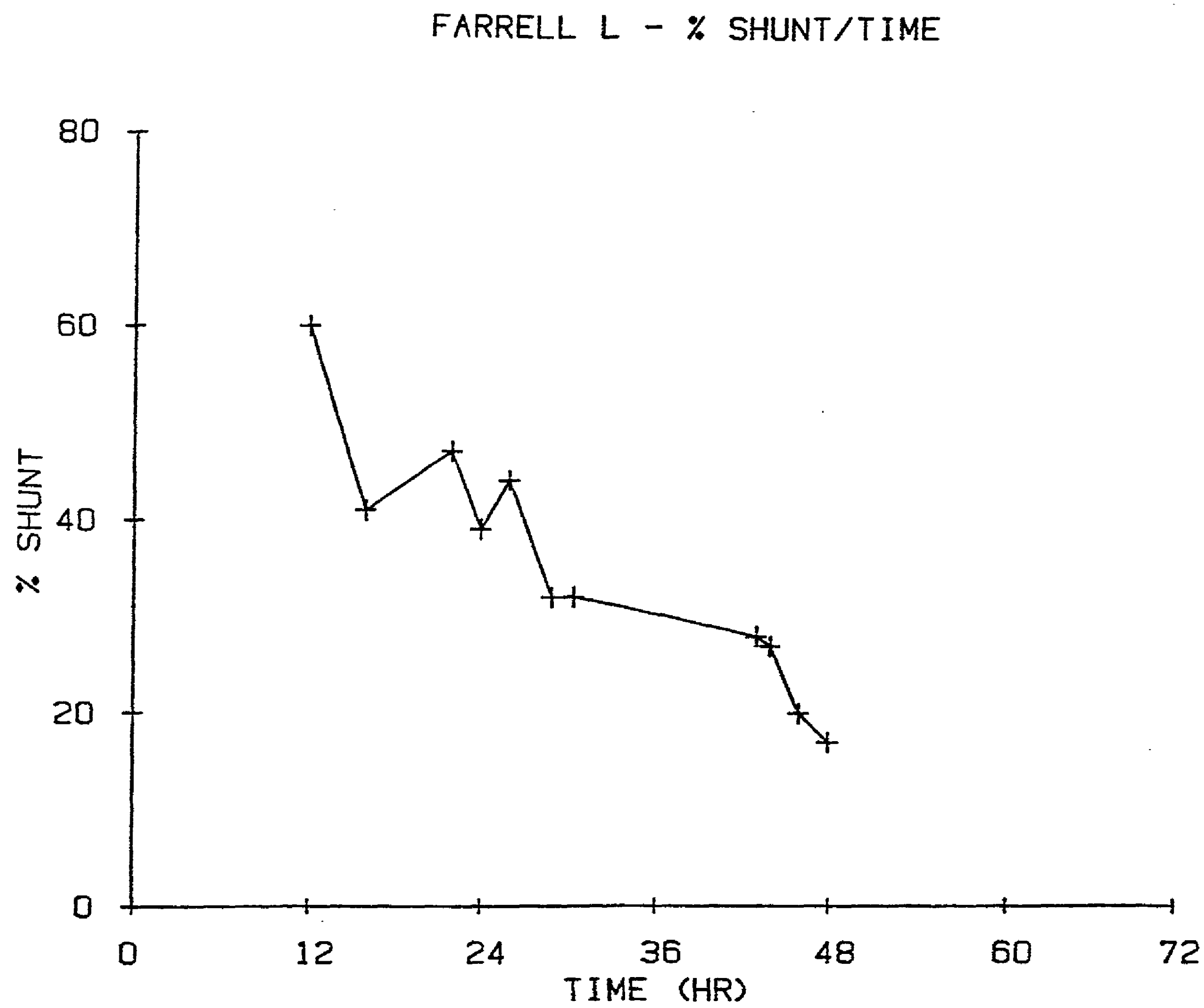

Figure 41 Improvement in intrapulmonary shunt during ECMO.

Figure 42 Improvement in respiratory mechanics measured on-line following successful antibiotic therapy.

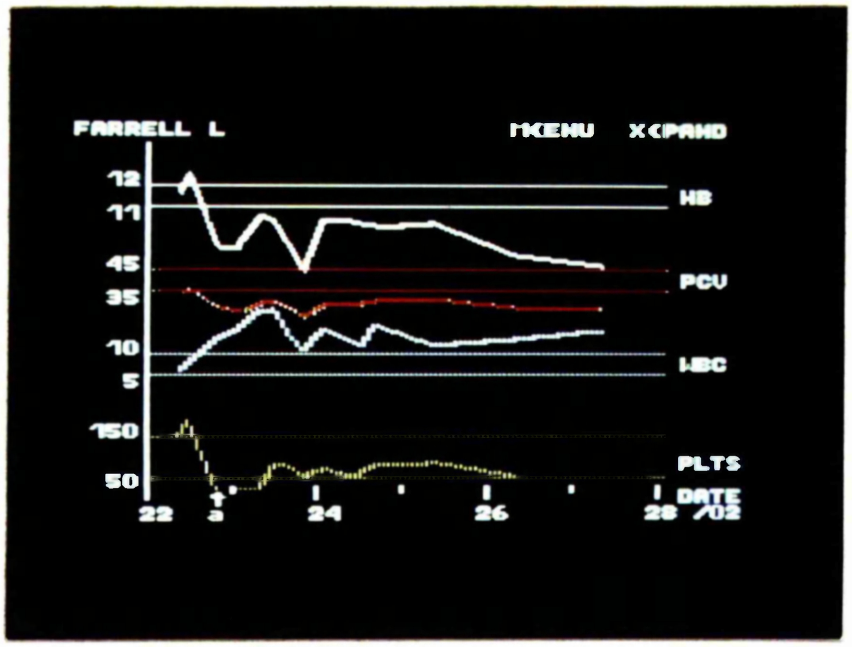

Figure 43 Alterations in haematology measurements during, and after ECMO.

The successful management of a patient with this degree of complexity requires the measurement and appreciation of many variables and it is not possible for nursing staff to adequately monitor, record and nurse those patients. Automatic collection and graphical display of the trends enable the monitoring, presentation and recording of a large part of the necessary data to be undertaken. Subsequent analysis is made much simpler and should enable possible areas for improved management to be identified.

CASE 2

A 60yr old female was admitted to the intensive care unit following laparotomy for perforated bowel. Her cardiovascular system was extremely unstable (fig. 44) and required support with dopamine. Intermittent measurement of the arterial pressure would have provided an incomplete, if not inaccurate, estimate of the patient's cardiovascular system. This supports the conclusions of Hilberman (31) and Sheppard (80) that computers can provide more accurate data than standard monitors or observations.

THe patient suddenly developed marked peripheral vasoconstriction which was first identified by a decrease in SvO2 (point 'a', fig. 45). Systolic arterial pressure and PAWP also decreased. Sodium nitroprusside therapy was commenced with subsequent improvement in SvO2 and arterial pressure.

Her sensitivity to dopamine is illustrated at point 'b' in figure 45. A small bolus of dopamine had been inadvertently administered to the patient which caused the arterial pressure to increase but Svo2 to decrease, indicating that the therapeutic concentration had been exceeded. Nursing staff could not be expected to note such rapid alterations in events and if doctors are occupied elswhere, the significance of these changes may not be appreciated fully. The patient required increasing amounts of dopamine for cardiac support and a combination of respiratory and

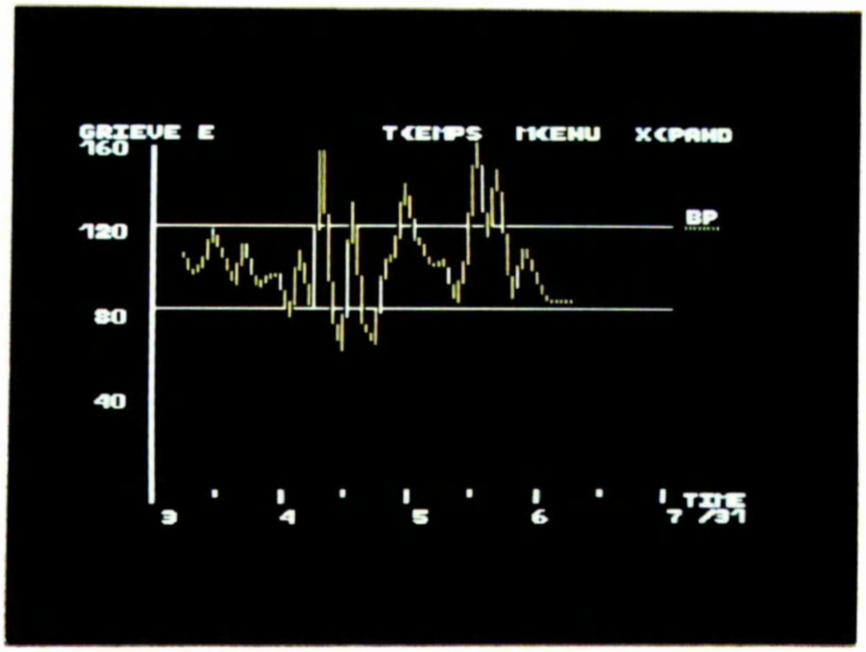

Figure 44 Display of systolic arterial pressure collected on-line by computer showing extreme instability.

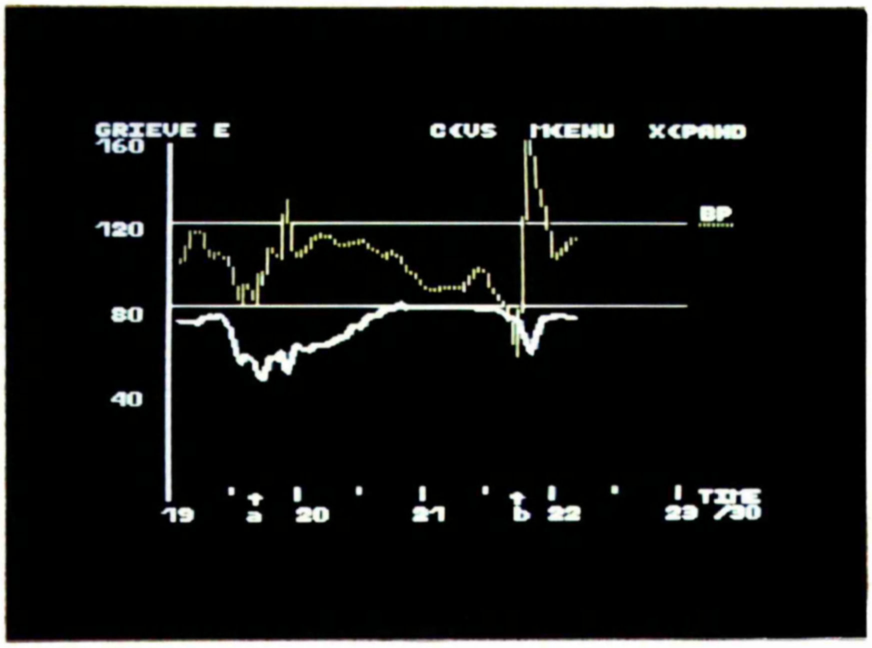

Figure 45 Alterations in systolic arterial pressure and SvO2 folowing (a) infusion of sodium nitroprusside
(b) bolus of dopamine.

cardiac failure resulted in her death.

One consequence of this form of monitoring has been the introduction of pulmonary artery catheters with an additional venous line for administration of drugs such as dopamine. This should prevent the effect shown at point b in figure 45 caused by alteration in the flow of fluid which carrieds the concentrated dopamine infusion into the patient's circulation.

CASE 3

As 56 year old male underwent elective operation for insertion of an aortic bifurcation graft. Cardiovascular data was collected on-line during the operation and subsequently in the ward and stored on magnetic disc. After induction of anaesthesia, a pulmonary artery catheter was inserted and PAWP measured intermittently.

ABGSTAT was used to compare values of PAWP and CVP measured and the results were organised for entry into the HP PLOT program as described previously. A correlation coefficient of 0.78 was obtained and the regression line together with the individual results are graphed in figure 46. CVP was sometimes well correlated with PAWP for this patient; but at other times, for example at a CVP of 11 mmHg, it provided only a poor indication of the effective circulating blood volume indicated by PAWP. If CVP alone had been used to guide the requirements for infusion of fluids, the patient may easily have been overloaded.

The changes in cardiac output and oxygen consumption with time are shown in figures 47 and 48 and it can be seen that both increased in the ward after operation. A possible explanation for this result is that oxygen demands increased postoperatively and required considerable increase in cardiac output. Although systemic vascular resistance decreased (fig.49), it was presumably limited by the condition of the patient's vasculature since arterial pressure increased and proved resistant to

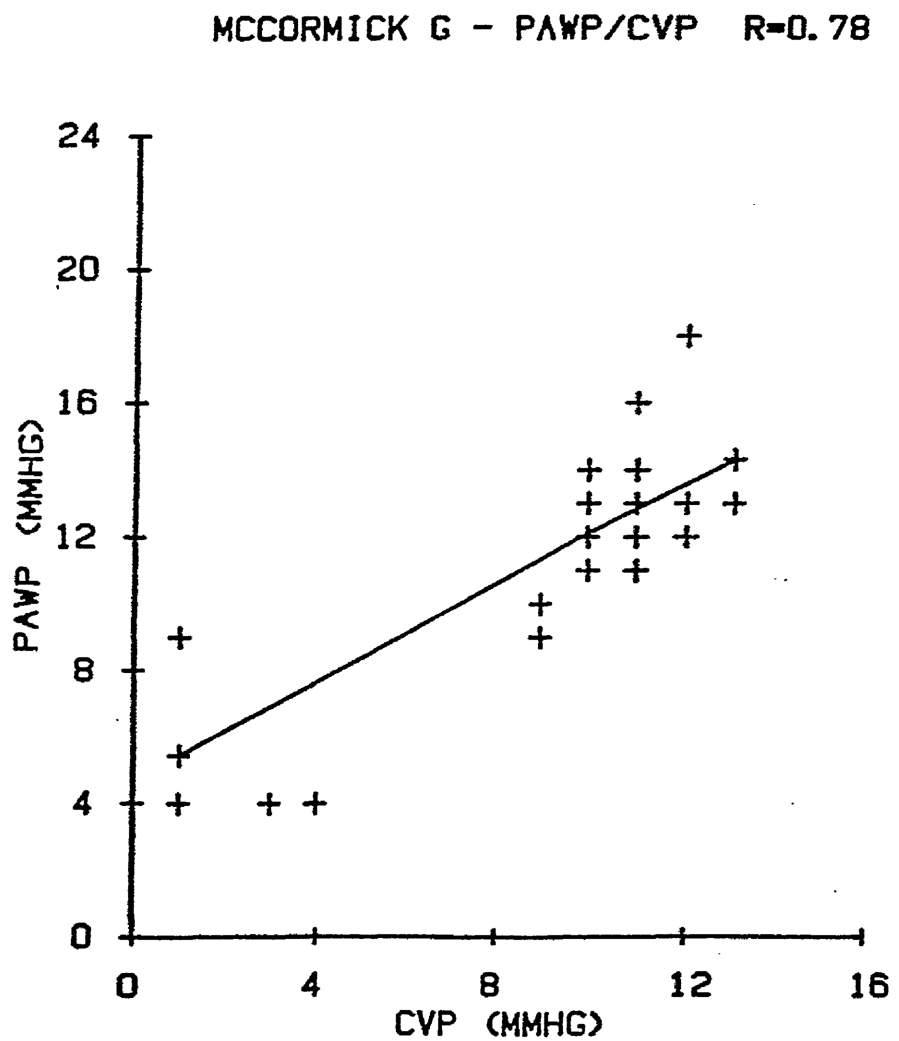

Figure 46 Comparison of CVP and PAWP intra- and postoperatively.

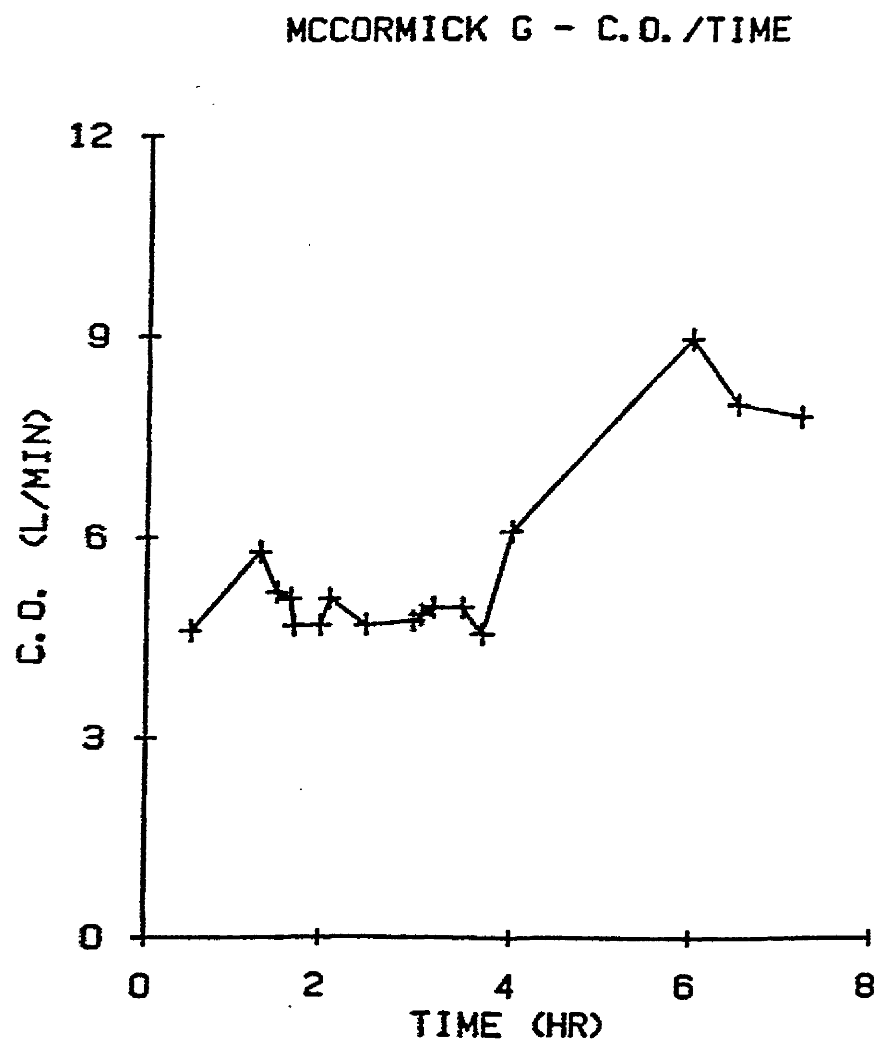

Figure 47 Alteration in cardiac output intra- and postoperatively.

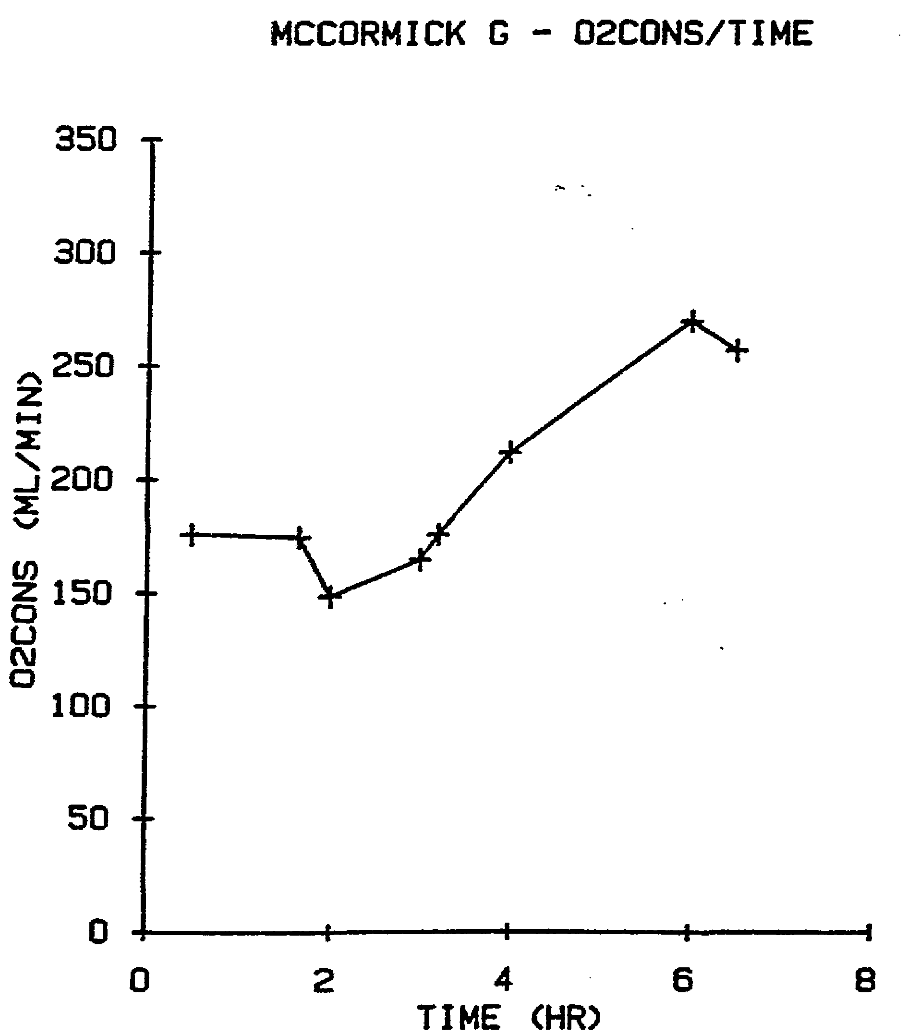

Figure 48 Alteration in oxygen consumption intra- and postoperatively.

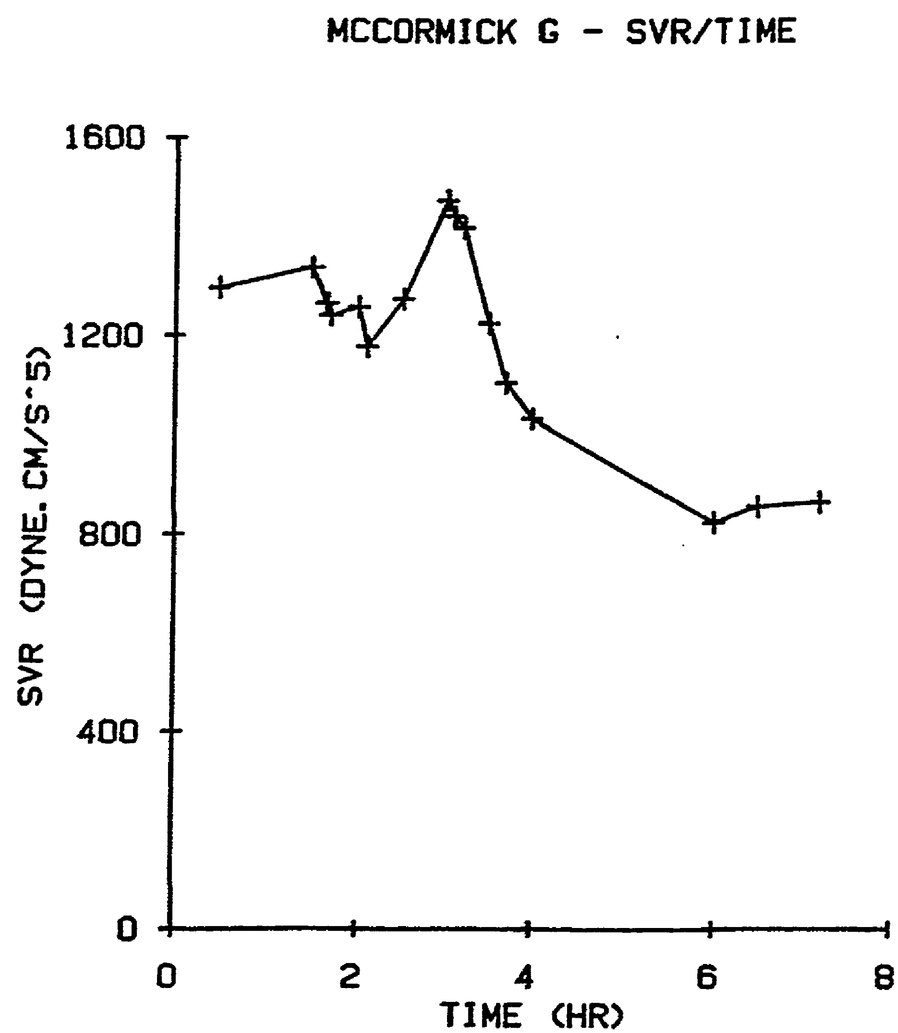

Figure 49 Alteration in systemic vascular resistance intra- and postoperatively.

vasodilator therapy.

A leak developed in the Dacron graft and the patient underwent an emergency operation to have this repaired. The patient returned to the intensive care unit, and the comparison of PAWP and CVP following the second operation is shown in figure 50. The patient had developed atrial fibrillation and a different correlations was found on this occasion but with still a wide variability of PAWP and CVP. The patient recovered from this complication but developed massive haematemesis subsequently and died.

The computer system tends to be used by the medical and nursing staff for the most severely ill patients. The case reports selected illustrate the problems in dealing with the large amounts of data which are made available and the features which may not be noted by standard monitoring.

The system has demonstrated how computers can be used routinely in an intensive care unit and operating theatres to assist in patient management by collecting data automatically and presenting this information in an easily understandable manner. Calculations which would normally consume a considerable amount of time are performed rapidly enough to be of value in the determination of the optimum therapy for critically ill patients. Nurses can enter comments and notes as events occur and have these accurately recalled and printed out to enable them to inform more accurately their colleagues who take over the care of the patient.

Data entered off-line using the keyboard or obtained on-line from monitoring equipment can be recalled rapidly and displayed in numerical or graphical form. At the same time, the information is stored in a manner which enables simple and easy retrieval and analysis to identify important correlations or to understand more clearly the sequence of events which a patient has undergone. The eventual aim is to not only collect and display the information but to attempt predict possible

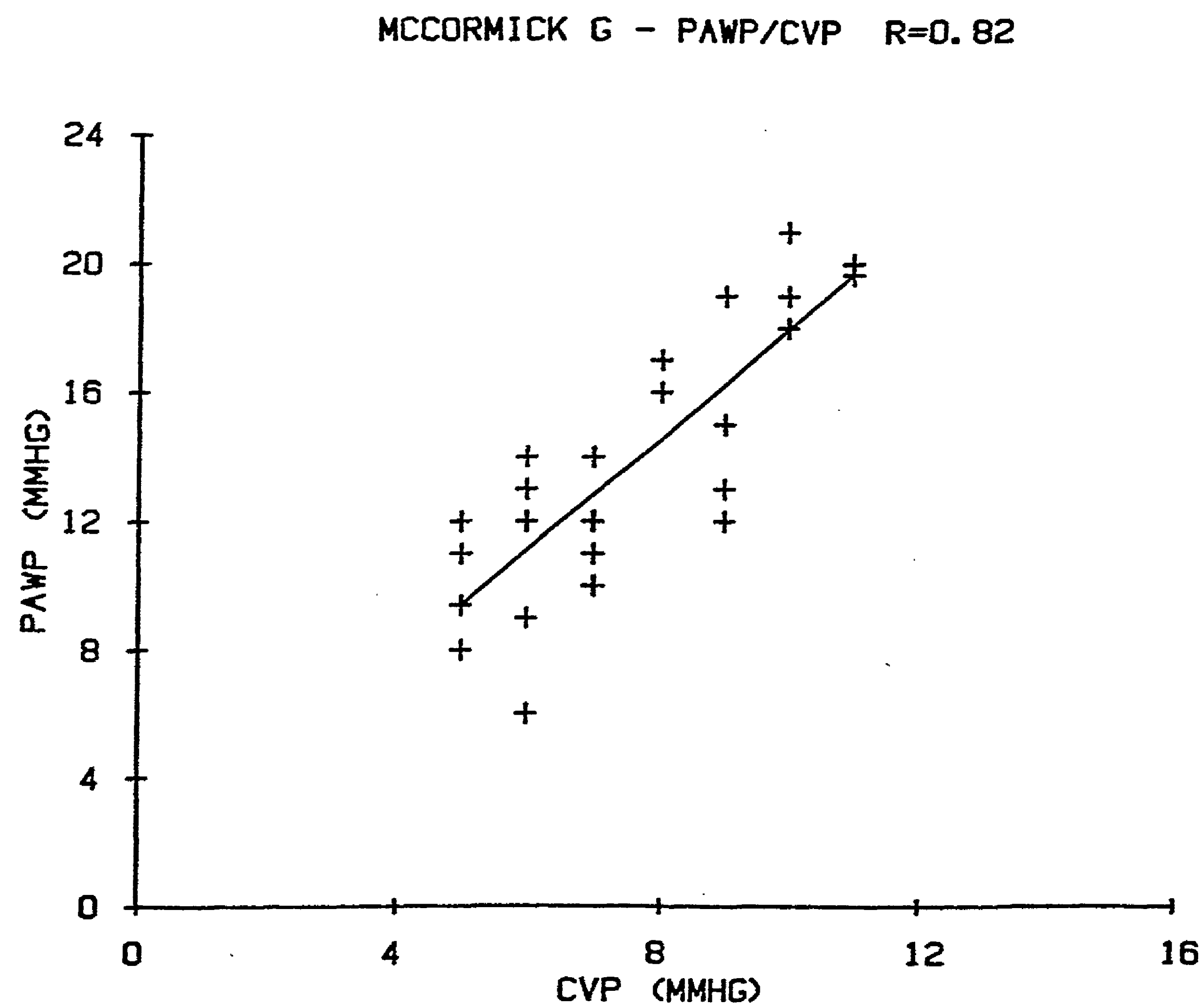

Figure 50 Comparison of CVP and PAWP following second operation after repair of leaking Dacron graft.

difficulties before they arise. This aim will only be realised when the data from all severely ill patients are collected and analysed routinely by computer.

While microcomputers have been shown to be of benefit for patient care it is perhaps in the area of experimental work that they will have their initial impact in many hospitals since the ability to collect data automatically and store this in a suitable form for analysis is such a valuable feature for research.

CHAPTER 4

MICROCOMPUTERS IN EXPERIMENTATION

MICROCOMPUTERS IN EXPERIMENTATION

INTRODUCTION

The final example of the application of microcomputers to anaesthesia describes a program in which the computer controls the apparatus used in the experiment and collects all required data. Subsequent analysis is also performed by the computer and the entire system is designed to minimise the time and effort required to undertake research.

The project to be described could not have been undertaken in its present form if a computer had not been incorporated into the design because of the precise timing of events which was required. Observer bias is removed from the process of data collection since this function is undertaken by the computer system. Analysis of data frequently requires many hundreds of measurements to be made from pen recording. This is time consuming and also liable to inaccuracies. The 8-bit ADC used for this project has a resolution of 1 part in 255 but 12-bit devices are readily available with a resolution of 1 part in 4095.

Many doctors working in district general hospitals will probably have original ideas of research projects which could usefully be undertaken. However, there is often little time available for such ventures. If data collection and analysis could be performed by a computer, such projects could become feasible.

CARBON DIOXIDE RESPONSE

The possibility of delayed respiratory depression following the use of extradural morphine for postoperative analgesia has been reported in several studies (13,70). A recent study in volunteers not undergoing surgery (56) showed differences in the degree and time course of respiratory depression following i.v. and extradural morphine as assessed by the ventilatory response to rebreathing CO_2.

The ventilatory response to a challenge of increasing inspired concentrations of CO_2 (69) has been widely used as an estimate of respiratory drive. However, in patients who have undergone abdominal surgery, the ventilatory response may not be the most appropriate variable to measure since it may be affected by imperfect analgesia or alterations in pulmonary mechanics independently of any effects on chemoreceptor sensitivity and respiratory drive.

Two measures of respiratory drive have been described which are isometric and therefore less influenced by factors which may alter the ventilatory response :-

1. the occlusion pressure developed at 100 ms (P 0.1; 97).
2. the maximum rate of change of isometric inspiratory pressure (50).

The first method required a shutter valve to be closed during the expiratory phase of respiration. The patient then inspired against the closed shutter and developed a negative inspiratory pressure which was measured 100 ms after the beginning of inspiration. Measurement of the inspiratory pressure shows a good correlation with more direct estimates of respiratory drive such as integration of phrenic nerve activity (21) and measurement of the diaphragmatic electromyogram (47). A good correlation with CO_2 tension has also been found (3, 99).

However, with the experimental technique used by Whitelaw and his colleagues (97), it was not possible to obtain P 0.1 for every breath during a test since the shutter was operated manually and could not be opened until the operator was satisfied that an acceptable measurement had been obtained. The subject was therefore acutely aware of the occlusion and began to gasp, which altered measurements of subsequent breaths (97, 45).

The second method measured the maximum rate of increase of inspiratory pressure when the inspiratory valve was loaded to prevent its opening until a negative pressure of 1 kPa had developed. The subject was not aware of the measurements being made and consequently this method allowed data to be collected breath by breath. However, moisture collecting on the valve was found to alter the pressure at which the valve opened (81). This could produce alterations in the maximum rate of pressure change recorded by their system independent of changes in respiratory drive.

The merits of these two techniques can be combined by measuring P 0.1 for each breath. This is performed by occluding the inspiratory airway for only the duration of time required to obtain the measurement. The method is based on the rebreathing technique where the subject rebreathes from a bag which initially contains 50% O2, 43% N2 and 7% CO2. The circuit is shown in figure 51, and the apparatus used, in figure 52.

The reservoir bag is suspended in a glass bottle and as the patient inspires the gas mixture from the bag, air is drawn into the glass bottle through the flow head. The decrease in pressure across the flow head is analysed by a pneumotachograph (VP5, Mercury Electronics) which was specially modified by the manufacturer to provide analogue outputs suitable for connection to the ADC used in the Apple II microcomputer. The outputs from the VP5 which are used by the computer are :-

1. negative airway pressure
2. tidal volume
3. output signal to indicate inspiration or expiration

In addition, the VP5 accepts an input signal from the computer to

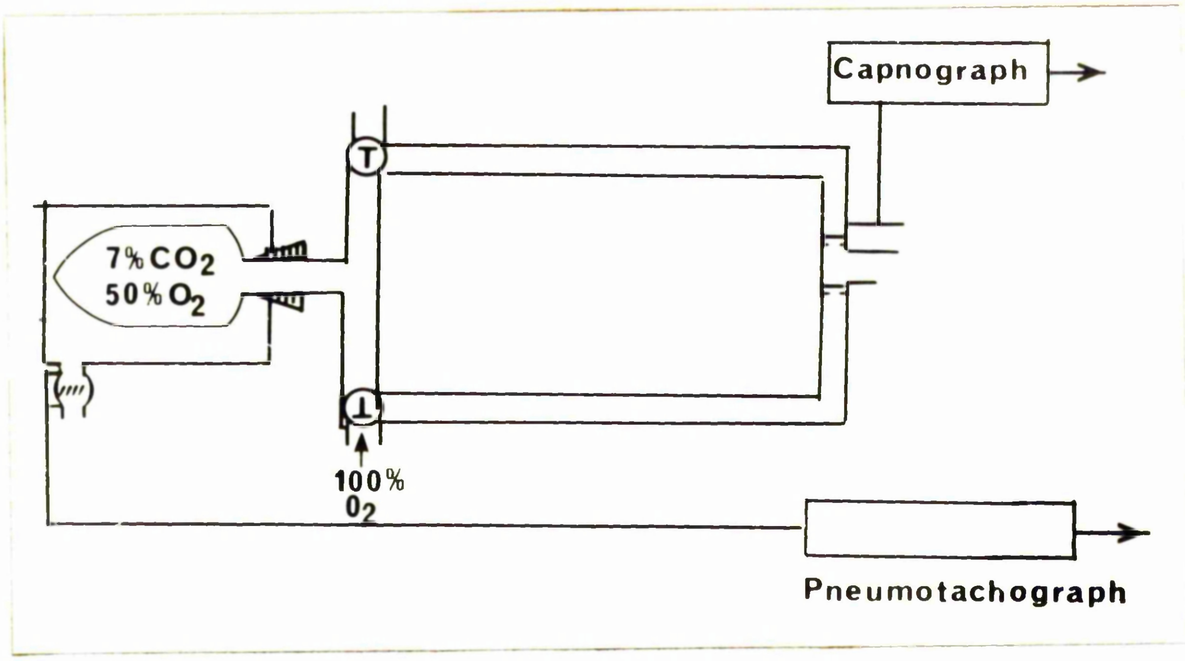

Figure 51 Circuit used for measurement of response to rebreathing CO2.

Figure 52 Apparatus use for CO2 response experiments.

operate the relay controlling the occlusion shutter.

A continuous duty solenoid shutter, which can be closed by a signal from the VP5, is placed at the proximal end of the inspiratory tubing. End-tidal concentration of CO2 is measured with a capnograph (Beckman) and the output also connected to the ADC. All collection of data and the operation of the occlusion shutter are controlled by the computer.

The start of inspiration is signaled by detecting a decrease in the pressure developed at the mouthpiece of less than -0.02 kPa. The computer measures the negative pressure developed 50 and 100 ms later and opens the occlusion shutter at 120 ms. Tidal volume is measured at the end of inspiration and end-tidal CO2 at the end of expiration. The data are stored on disc and subsequently analysed by the computer.

CARBON DIOXIDE RESPONSE SUITE

The overall organisation of the programs used for collection and analysis of the response to rebreathing of carbon dioxide is shown in figure 53. As with the other suites decribed previously, the HELLO program is run automatically when the computer is switched on. This displays a menu which allows the user to collect, print out or analyse data.

CO2SPACE

If the data collection option is selected, CO2SPACE is run. This measures the free space remaining on the disc and if it is less than 5% of the total, it will not allow the user to procede. A new disc containing the programs must be inserted and the HELLO program is re-run. When adequate space is available on the disc, the program CO2SETUP is loaded and executed.

CO2SETUP

The same routines are used as in the cardiovascular program suite to obtain from the file, NAMEFILE, the name of the patient which was last entered into the system. Usually, a different name has to be entered and this is undertaken by the routine from line 310 which has the standard checks for incorrect data entry. The data collection program CO2RESPONSE is then loaded and run.

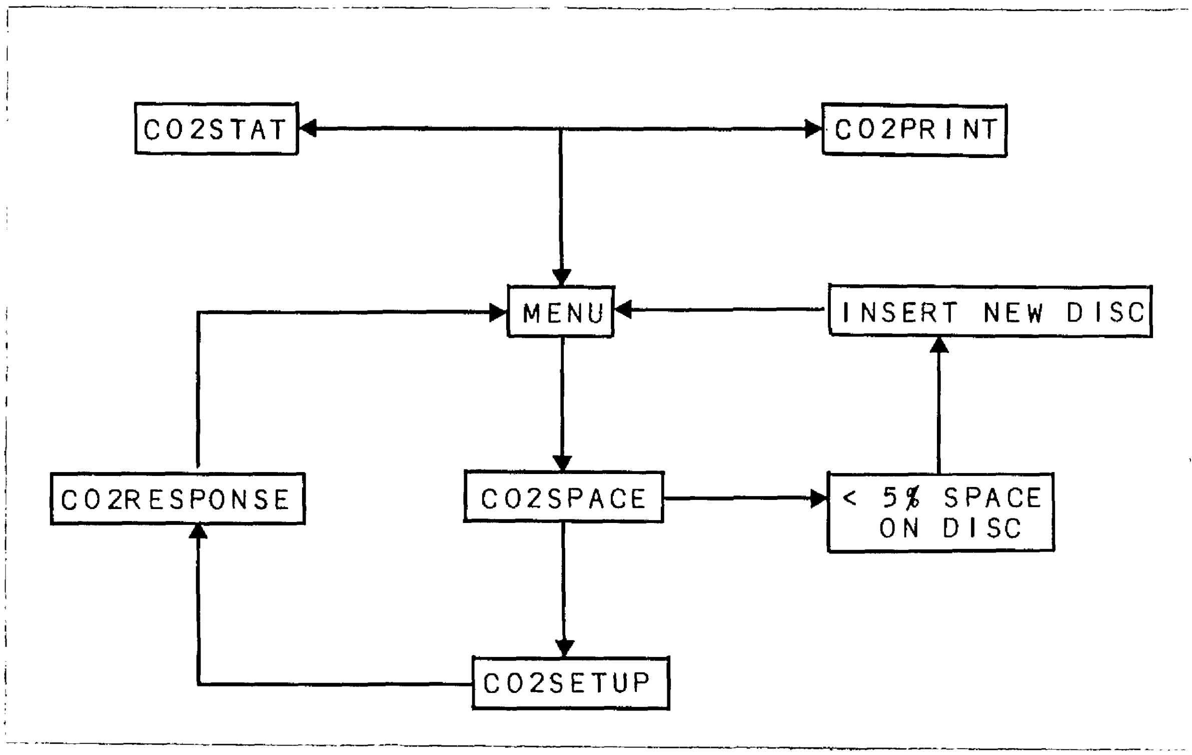

Figure 53 Organisation of CO2RESPONSE suite.

CO2RESPONSE

This is the program which controls the experiment and collects data. The functions performed are:-

1. measurement of end tidal CO2 concentration.
2. measurement of tidal volume and total time for each breath, and from these, calculation of the instantaneous minute volume, mean flow and respiratory rate for each breath.
3. closure of the solenoid operated shutter during expiration.
4. detection of the start of inspiration.
5. collection of the inspiratory pressure developed 50 and 100 ms after the beginning of inspiration.
6. opening of the shutter at 120 ms.

The organisation of CO2RESPONSE is shown in figure 54. Data are collected from the measuring instruments using the commercially available ADC (Interactive Structures AI-02) which was used also for collection of data from the Roche 2000 cardiovascular monitor and the Seimens Servo 900 B ventilator. Events are timed with the Mountain Hardware clock /calendar card or by a specially designed timer card each of which allow timing to an accuracy of 1 ms. Timing data are obtained from the cards by machine code programs and the information stored in memory to be retrieved later in BASIC.

The pneumotachograph analogue output of tidal volume is obtained using BASIC but all other signals are collected using machine code routines. Two machine code programs are available for data collection. The user indicates which timer card is present in the computer and the correct machine code program is loaded into memory. These are described in more detail later.

The shutter is controlled by the programmable switches in the games port of the computer . When mains power is applied to the computer, the games port switches become positive and the

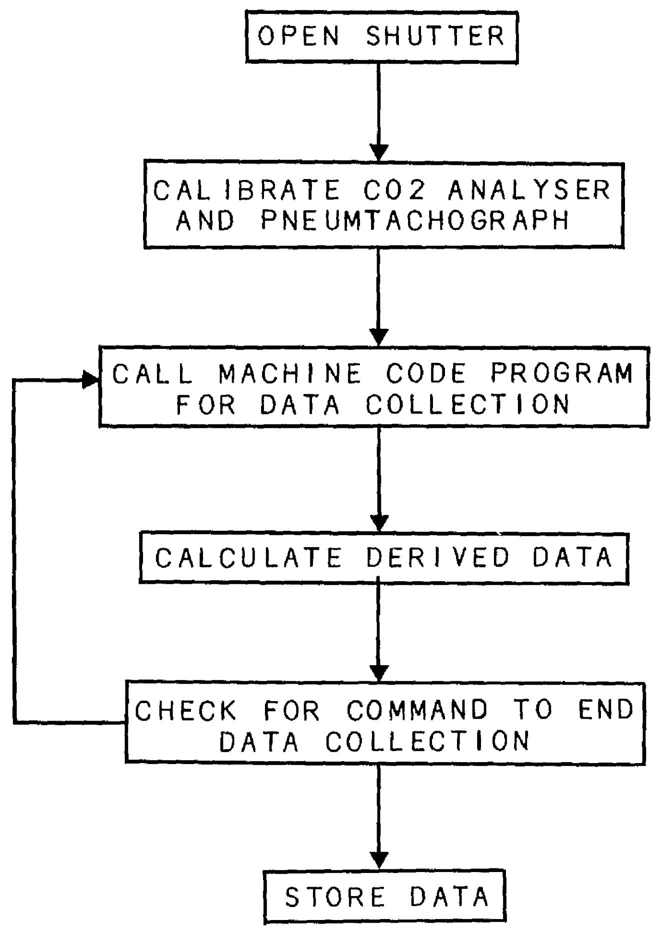

Figure 54 Organisation of CO2RESPONSE program.

shutter is closed. Therefore, immediately after loading the program and setting the lowest memory location available for storage of BASIC variables at 28000, the program passes to the subroutine at line 5000 which turns off the switches in the games port and opens the shutter. The variables required in the program are then set at the routine SET VARIABLES from line 520 and the arrays for storing data are dimensioned.

The barometric pressure (PB) is required to convert percentage concentration of CO2 into kPa and the default value of 760 mmhg is displayed along with instructions to allow this to be altered if necessary. The user is asked if he wishes to calibrate the system, and if so, the program passes to the subroutine CALIBRATE which is described below.

If this is not required, the last calibration values are obtained from the file CALVALUES. When no calibration values are stored in the file, a disc read error is detected. An appropriate message is displayed on the screen and the program passes to the calibration subroutine. When the calibration values have been satisfactorily acquired, the name of the patient undergoing the test is obtained from NAMEFILE.

The start of inspiration is detected by monitoring the negative pressure developed at the mouthpiece. When the pressure decreases below a preset threshold, inspiration is deemed to have begun. The default threshold used is -0.02 kPa but can be altered to any desired value with the SET PRES THRESHOLD routine at line 550. The threshold value in kPa is converted to the digital equivalent and then stored in location 27808 for later use by the machine code program.

The user is asked to identify the timing card used in the computer and the relevant machine code program is loaded into memory. Before the data collection procedes, the user is reminded to reset the pneumotachograph to zero all the signals.

The output from the pneumotachograph is examined to ensure that the programs enters the data collection routine at the end of expiration. The I/E signal on channel 11 is examined at line 750 and if this shows that the patient is inspiring, the program waits until the patient begins to expire. It then continues to monitor the I/E signal until the start of inspiration when data collection begins for the next respiratory cycle.

CALIBRATE

The program passes to this subroutine if calibration of the system is required either because this option was selected by the user or by default if no calibration values were obtained from the disc. The flow signal, pressure sensor and CO2 analyser are each calibrated in turn.

Volume

The user is instructed to attach a precision 1 litre syringe to the patient circuit and to move the plunger out and in. The pneumotachograph integrates the flow signal and provides an analogue output proportional to the volume measured. At the end of 'inspiration', the volume then equals the Vt for that 'breath'. The volume signal is connected to channel 7 of the ADC and a value is obtained and stored in the array VI(X). The I/E channel is examined to ensure that 'inspiration' is still proceeding and a further value obtained from the volume signal.

When the I/E signal indicates that inspiration has been completed, the array VI(X) is scanned and the maximum taken as the digital value of Vt. The digital value is multiplied by the calibration factor for Vt (SI) and compared with the true value of 1 litre. When the measured value is within 1% of 1 litre, the calibration factor is accepted and the user informed. If the measured value is outwith the acceptable range, the calibration factor SI is altered and a further series of measurements is made until a satisfactory calibration factor is obtained.

Pressure

The pressure sensor is calibrated by pressurising a manometer and connecting this to the pressure input of the VP5. The user is asked to enter into the computer the manometer pressure and 100 digital values are collected. These are averaged and the calibration factor for pressure (SP) calculated.

CO2 Analyser

The CO2 channel on the ADC is read and the digital value displaye on the screen. The user is asked to adjust the zero control on the analyser to give 0-1 on the screen. The calibration gas normally contains 7% CO2 but any concentration can be selected. The gas is passed through the Beckman infra-red CO2 analyser and the signal from the analyser connected to channel 3 of the ADC. The digital value corresponding to the output from the CO2 analyser is then displayed on the computer screen.

Ten per cent CO2 produces an output of 5V which, when applied to the ADC, produces a value of 255. Seven per cent should produce a value of 178.5. The user is therefore instructed to adjust the gain on the analyser so that the digital value displayed is 178-179. The correct digital value is displayed to the user whatever concentration of CO2 is used for calibration up to a maximum of 10%. The CO2 analyser is in effect being calibrated using the computer and the ADC as a digital voltmeter with a resolution of 1 in 255.

Once calibration has been completed and the new calibration values are stored in CALVALUES for later use, the program returns from the calibration subroutine to line 620. The patient name is obtained from NAMEFILE, the correct machine code program loaded and the I/E channel monitored to ensure that data collection begins at the start of inspiration.

DATA COLLECTION

Tidal volume is collected during the inspiratory phase from lines 200-260 and the expiratory data from lines 290-340. The variable, TE, is incremented and is used to reject the first breath to allow the timing data to be collected in the correct order. The volume signal from the pneumotachograph is collected and stored in the array, VT(IC). After each sample of the volume signal has been collected, the I/E channel is monitored to ensure that the patient is still inspiring and the program continues to obtain further values until the end of inspiration is detected.

If the flag, 'CLOCK$', has been set to 'ST', the Strathand timer card is in use and the time for the end of inspiration is stored by calling the machine code routine starting at location 775. The high byte of the inspiratory time is stored in location 27802 and the low byte in location 27803. The machine code program zeros the timer at the start of inspiration so that the durations of inspiration and the total respiratory cycle are simply the values recorded in milliseconds at the appropriate times.

When the Mountain Hardware clock is used, the time for the end of inspiration is obtained by calling the portion of the machine code routine starting at location 768. With the Mountain Hardware clock, the counters for less than 1 s are set to zero at the start of inspiration by the machine code program, but those for 1 s and greater are not zeroed.

The time for the start of inspiration is stored in memory by the machine code program and this time must therefore be read at the end of inspiration before the machine code program is called again and a new value obtained for the next breath. The six characters of the start time are concatenated into the variable T0$ at line 222 in the same manner as for the timings in the ROCHESERVO program. No data have been stored for the first breath taken by the subject and an illegal quantity error is generated if an attempt is made to read the memory locations.

The program therefore does not obtain a start time for the first breath and the data for the first breath are rejected.

The VT(IC) array is scanned to obtain the peak value of the volume signal which is equal to the tidal volume for that breath. This is followed by a delay of about 0.75 s which is inserted to ensure that the initial portion of expired gas which contains the dead space gas is not sampled since this will have a greater concentration of CO2 than the end tidal sample during the first part of the test. The signal to close the inspiratory occlusion shutter is then sent from the games port by the command POKE -16295,1.

If the Strathand timer is in use, control is passed to the machine code program at location 794; otherwise, control is passed to location 773. These data collection routines are identical except for the timing portions which are dependent on the timer card used. The digital value from the CO2 analyser is obtained and the maximum stored in location 27807. Between the collection of each CO2 value, the pressure signal from the mouthpiece is monitored to detect the changeover from expiration to inspiration.

When the mouthpiece pressure exceeds the preset threshold which was stored in location 27808 in BASIC after selection by the user, the time for the total duration of the respiratory cycle is stored. With the Strathand timer, the high byte of the time in ms in stored in location 27804 and the low byte in location 27805. The timer is then set to zero and the low byte scanned to detect 50 ms and 100 ms. Where the Mountain Hardware card is used, the TIME1 subroutine in the machine code subroutine is called and the six characters for the time at the end of the respiratory cycle are stored. The time bits less than 1 s are set to zero and scanned to detect 50 ms and 10 ms.

Fifty and one hundred milliseconds after the beginning of inspiration, the pressure in the mouthpiece is obtained and stored in locations 27806 and 27809. Immediately after the beginning of inspiration, the signal is sent to open the occusion shutter but because of the latency inherent in the solenoid, the shutter does not begin to open until 120 ms after this signal is sent. Control is then returned to BASIC, the breath number N incremented, and the values for the occusion pressure at 50 ms (P 0.05), 100 ms (P 0.1) and the end-tidal CO2 obtained. The machine code programs are described in detail later.

With the Strathand timer card, the inspiratory time in ms is obtained by PEEKing locations 27802 and 27803. The expiratory time is calculated by obtaining the total time from locations 27804 and 27805 and subtracting the inspiratory time. When the Mountain Hardware card has been used, the clock time for the end of inspiration and the end of expiration are obtained in the same manner as for the timings in ROCHESERVO and stored as T1$ and T2$. Three time strings are obtained for one breath from the Mountain Hardware clock :-

1. T0$ = start of inspiration
2. T1$ = end of inspiration
3. T2$ = end of expiration

The strings are converted to real numbers using the VAL command and the inspiratory and expiratory times calculated. The program passes to the CALCULATION routine where the data are used to derive the required values which are displayed on the screen.

The computer prompts the user to type '1' when the rebreathing test has been concluded. Before collecting data from a further breath, the program scans the keyboard to detect if the numeral '1' has been typed. If this has occurred, the data are displayed on the screen and then stored on disc. The values stored for each breath are :-

1. end-tidal CO2
2. instananeous minute volume
3. P 0.1 and P 0.05
4. respiratory rate
5. expiratory/inspiratory ratio

6. mean inspiratory flow rate

A menu is then displayed which enables further data to be obtained, the HELLO program to be run, or the session to be terminated. If another test is to be undertaken, CO2SETUP is loaded and run to allow a different name to be entered. This name will be used to label the file in which data from the patient will be stored, for example, Smith JA, Smith JB. If the HELLO program is run, the user again has the option of collecting further data, printing out the results of previous studies or of analysing results. When the print option is selected, the program CO2PRINT is run.

CO2PRINT

The user is instructed to place the disc containing the data in the drive and type the RETURN key when ready. He is then asked whether or not a printer is connected to slot 1 in the computer. If a printer is available, the variable PR is set to 1, otherwise PR is allocated the value zero. The name of the patient for whom the data are to be retrieved is entered and the file read. The standard ON ERR command is used if a disc error is detected and the user is informed that the file is unavailable.

When no printer is available, four results are displayed on the screen at once. Each succeeding block of four results is then displayed by typing any key. If a printer is available, the results are all printed out in succession. More results from a different experimental run or patient can be printed out or control can be returned to the HELLO program.

DATA ANALYSIS

The data are analysed for correlations between the end-tidal CO2 and the variables measured as estimates of respiratory drive. These are instantaneous minute volume (Vi), occlusion pressure at 50 and 100 ms (P 0.1 and P 0.05), respiratory rate, ratio of expiration to inspiration and mean inspiratory flow (MIF). The data files obtained by the CO2RESPONSE program are analysed by CO2STAT which is obtained from the menu displayed by the HELLO program.

CO2STAT

This program analyses the files obtained by CO2RESPONSE and calculates the correlation coefficient and the formula for the regression line. The file name to be analysed is entered and the files containing the Vi, P 0.1, P 0.05, rate, E/I ratio, MIF and associated end-tidal CO2 values are loaded. The entire data from disc are loaded into a three dimensional array N(X,Y,Z) where :-

N(0,Y,Z) = Vinst array
N(1,Y,Z) = P 0.1 array
N(2,Y,Z) = P 0.05 array
N(3,Y,Z) = respiratory rate array
N(4,Y,Z) = E/I ratio array
N((5,Y,Z) = MIF array.

The FOR-NEXT loop between lines 724 and 780 selects Vinst for analysis where J = 0, P 0.1 where J = 1, and continues until MIF when J = 5. The values to be entered into the correlation and regression calculations for each of the arrays are stored as X(A,0) and X(A,1) by the FOR-NEXT loop between lines 725 and 740. If either of the pair of values is equal to zero, it is eliminated from the calculation since it could not have been a valid measurement. This happened occasionally when a complete breath had not been taken by the patient but had been detected as a breath by the system. The variable 'K' is then incremented so that when the next value is obtained from the 'N' array, it is stored in the 'X' array at position I-K. In other words, the unacceptable value is overwritten by the new value.

The analysis of the correlation between end-tidal CO2 and the estimates of respiratory drive initially compares all values of CO2. It then calculates the correlations and regression equations of successively greater values of CO2. This is achieved by setting a threshold above which the CO2 and its corresponding estimated of respiratory drive are analysed. The variable SECO2 is set to 6 after the analysis of all the values of CO2 has been completed. When the 'X' array is extracted from the 'N' array, not only are the values which are stored in the 'X' array checked for being equal to zero, the CO2 value is checked to ensure that it is greater than the preset value, SECO2. If the CO2 value is less than SECO2, 'K' is incremented and that pair of data is not used in the statistical calculations.

The two dimensional 'X' array containing the data pairs is passed to the calculation routines at subroutines COR COEF and REGRESSION which return the required values. The calculated results are displayed for each selected partial pressure of end-tidal CO2 and the next set are obtained by typing any key. A hard copy print of the final results can be obtained if required.

CO2 RESPONSE MACHINE CODE PROGRAMS

Two machine code programs are available for use with the CO2RESPONSE program. The appropriate routine is loaded depending on the timer card inserted in the computer. The organisation of these routines is shown in figure 55. The subroutines which obtain the timings will be described first and then the other sections which are common to both programs.

MOUNTAIN HARDWARE CLOCK / CALENDAR

This timing card makes available the date and time and can be read using BASIC or machine code. The date and time are set using the SET TIME program described under the cardiovascular section. The subroutines for obtaining the timings for CO2RESPONSE are from lines 63 to the end of the program CO2+P50+P100CLOCK and are based on the routines supplied with the card.

Certain locations may be used by the clock. Therefore, the contents of location $45 are loaded into the 'A' register and all the 6502 registers together with the contents of locations $38 and $39 are saved on the stack. The card is located in slot 3 of the computer and is accessed by the commands :-

```
LDA #$C3
STA $39
JSR $C300
```

This series of commands instructs the ROM on the card to obtain the date and time and load the data into temporary storage in memory. The locations and registers saved previously are restored and the data read from the temporary store into locations which are subsequently read from BASIC.

The time information is stored in the form of ASC II characters and six characters are acquired for each time

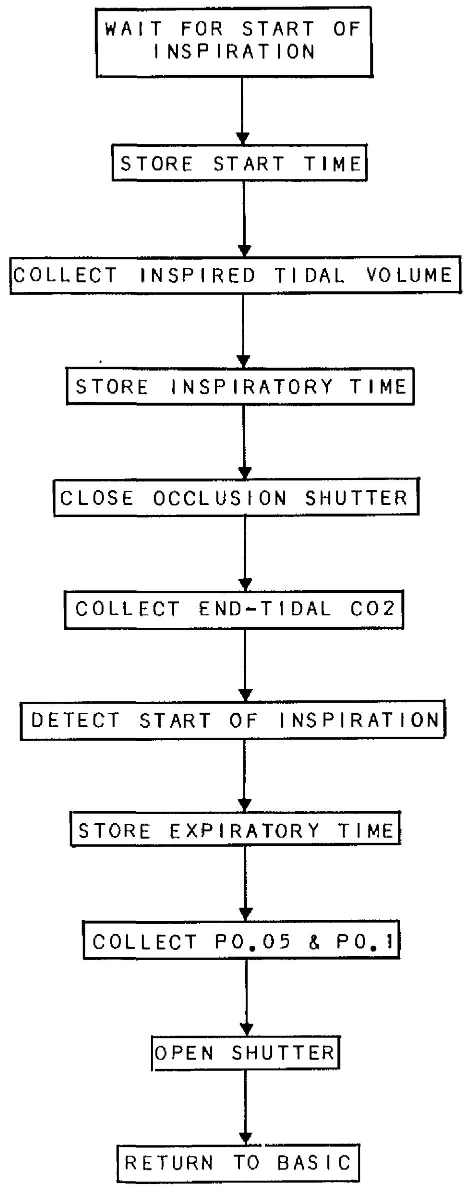

Figure 55 Organisation of CO2RESPONSE machine code program.

recorded; units and tens of seconds, a full stop and units, tens and hundreds of milliseconds. These are temporarily stored by the clock in locations \$282-\$287 and are relocated at subroutine LOADCLK into locations 27786-27791 for TIME0 which is the start of inspiration, the next six locations for TIME1 which is the end of inspiration and the next six for TIME2 which is the end of expiration.

The time bits for less than 1 s are stored as binary coded decimal and are set to zero when the clock is stopped by loading the 'A' register with the contents of location \$C0B6 at the beginning of inspiration. The clock is restarted by loading the 'A' register with the contents of \$C0B5 and the start time of the next inspiration is obtained by passing to the TIME0 subroutine at line 63. The units and tens of milliseconds elapsed since the start of inspiration are read repeatedly by accessing location \$C0B4 at subroutine READCLOCK until the required times have elapsed at which the two occlusion pressures should be measured. Since the time bits are binary coded decimal, the required contents of location \$C0B4 are \$50 and the next occurrence of \$00.

STRATHAND TIMER CARD

This card performs the much simpler task of incrementing two bytes of memory until 65 536 ms have elapsed at which time the counters return to zero and the process is repeated. The timer is started at line 2 by loading the 'A' register with the contents of location \$C0B2 and reset to zero by loading 'A' with \$C0B4. At the end of inspiration, the inspiratory time is obtained by stopping the timer and loading the high and low bytes of the counters into locations \$6C9A and \$6C9B. The time at the end of expiration is read in the same manner and the two bytes stored in locations \$6C9C and \$6C9C. The timer is then reset and the counters read until the required times have elapsed. The contents of location \$C0B4 should be \$32 and \$64 for 50 and

100ms.

Both machine code programs are called at the end of inspiration and the first part of the routine is to obtain and store the time for this event as described above. The subsequent portions are called during expiration and must perform the following functions :-

1. collect and store the end-tidal concentration of CO2
2. detect the beginning of inspiration and obtain the time for the end of expiration
3. wait 50 and 100 ms, and measure and store the occulsion pressures developed at the mouthpiece
4. open the occlusion shutter and return to BASIC

END-TIDAL CO2 / DETECT INSPIRATION

The final result for the measurement of end-tidal CO2 is stored in location 27807 which is loaded initially with zero. The program then loops at the subroutine INCHECK until the beginning of inspiration is detected.

INCHECK

Register 'Y' is loaded with 0 which is the channel number on the ADC connected to the negative pressure signal from the mouthpiece. The standard routine used in RESP.OBJ0 is again employed to obtain the digitised value from the channel selected on the ADC. This value is compared with the threshold pressure which was requested by the user during the initial part of the BASIC program and which was stored in location 27808. Above this pressure, inspiration is accepted as having begun and the program passes to the section labelled INSPN.

If the value recorded from the negative pressure channel is less than threshold, expiration is still progressing and the digital value for the signal from the CO2 analyser is obtained once again from channel 3 on the ADC. The digital value is compared with the previous value stored and, if greater, the new value is stored in location 27807 in place of the old value and the program returns to INCHECK.

END-EXPIRATION TIME / OCCLUSION PRESSURES

When the value of the negative pressure channel recorded at the mouthpiece exceeds the threhold set previously, inspiration has begun and the program passes to the routine INSPN.

INSPN

The first set of instructions record the time at the end of expiration by passing to subroutine TIME1. The shutter used has a latency of 120 ms in its response and the signal to open the shutter is therefore sent from the first switch output of the games port (pin 15) by accessing location $C058 at the start of inspiration.

The program passes to the routine READ50MS which continually reads the relevant clock or timer until 50 ms has elapsed at which time the negative pressure channel is measured at subroutine READPRES. The occlusion pressure at 50ms is stored in locataion 27809 to be retrieved from BASIC and the program passes to the routine READ100MS. This waits until 100 ms have elapsed from the beginning of inspiration, obtains the occlusion pressure from READPRES and stores this value in location 2786. Control is then returned to BASIC where the data stored in the different locations is extracted and the various calculations performed as described previously.

RESULTS

The original program used for data collection did not measure the occlusion pressure 50 ms after the start of inspiration and data using the latest program are not yet available. However, information from one patient are presented to illustrate the type of results which can be obtained using this system.

The P0.1 and instantaneous minute ventilation (Vi) were measured preoperatively and at 3.5 h and 5.0 h postoperatively following the administration of morphine intravenously and epidurally. The regression equations obtained from these measurements are shown in table VIII and are displayed in graphical form in figure 56. The graphs were arranged to superimpose the lines for Vi and P0.01 obtained preoperatively. At 3.5 h and 5.0 h postoperatively, a greater reduction in the slope of Vi occurred compared with P0.01. This suggests that the occulsion pressure, as an isometric measurement, may provide a more sensitive response to the challenge of increased arterial PCO_2 than Vi in postoperative patients.

This project is in progress at present and the results of the completed study is intended to be the subject of a future thesis by Dr J.C. Howie, lecturer in the University Department of Anaesthesia. However, it demonstrates the value and the method of incorporating a microcomputer into the experimental system. Standard, inexpensive components can be used to collect data automatically and to analyse the results easily. An equivalent analogue system is complicated to construct (34) and has the added disadvantage that subsequent modifications cannot be incorporated easily or analyses undertaken so readily.

	Preoperative	Postoperative	
		3.5h	5.0h
Vi	Y = 7.06X - 23.08	Y = 3.42X - 9.47	Y = 4.09X - 13.35
P0.1	Y = 0.10X - 0.42	Y = 0.07X - 0.41	Y = 0.06X - 0.27

Table VIII. Regression equations for Vi and P0.1 following challenge of rebreathing CO_2.

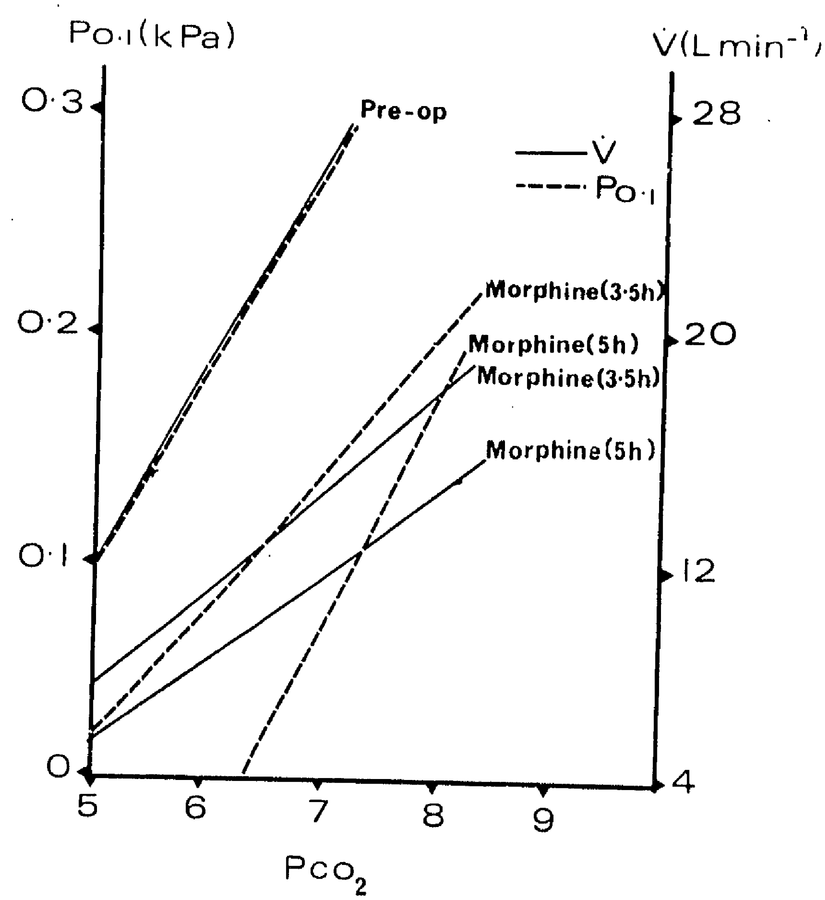

Figure 56 Response of P0.1 (broken line) and Vi (solid line) to epidural morphine.

CHAPTER 5

CONCLUSIONS

CONCLUSIONS

The aims of the project were:-

I To develop systems which fulfilled educational, clinical and research requirements.

II To determine if inexpensive microcomputers could replace large central computers for certain functions.

III To assess whether a clinician with no previous experience or training involving computers could develop programs which could serve a valuable function.

I To develop systems which fulfilled educational, clinical and research requirements.

The widespread interest shown by medical and nursing staff in the Royal Infirmary and elsewhere has demonstrated clearly that computers will be used increasingly in anaesthesia and intensive care. Three examples were selected to illustrate applications of microcomputers to anaesthesia and intensive care which were found to have been of value. These ranged from the simple structure of the off-line CAL and CASA programs to the complex on-line machine code control of the CO2RESPONSE program.

(1) CAL/CASA

Tuition by computer appears to be more efficient than the standard tutorial method. It also offers the potential benefit of providing instruction in district general hospitals which may be remote from large teaching centres and unable to provide adequate tuition for junior medical staff.

Shortage of software has been one of the most frequently noted problems associated with the use of CAL. The CALWRITER suite of programs should permit software to be developed more easily by those who are not computer experts. The large number of requests which have been received for copies of these programs suggests that many doctors, and others involved in education, are prepared to undertake this exercise.

Other problems have been lack of funds and of personel who are skilled in computing. Microcomputers are both inexpensive and simpler to use than previous computers and software developed in one centre can readily be used with a similar microcomputer elsewhere.

CASA has been shown to be acceptable to most of the anaesthetists, both selected and unselected, who took part in the evaluation. It therefore has the potential of being able to provide a system of medical audit which is acceptable to doctors and which would enable the public to be assured by the profession that their theoretical knowledge was acceptable since data collection and analysis are simple.

(2) Applications in the intensive care unit and operating theatre.

Data collection on-line has been successfully undertaken with improvement in the ability to manage critically ill patients. Data from monitoring equipment connected to such patients are now routinely collected, displayed and stored by the computer system. An increasing number of patients undergoing aortic bifurcation grafts have a pulmonary artery catheter inserted for monitoring intra- and postoperatively with data collected automatically by the computer. The entry of text comments has allowed much of the normal information recorded during anaesthesia to be stored and printed out in a well organised, logical manner at the end of the operation.

The Roche suite of programs require specialised interfacing to equipment but the design of the programs has been successfully transferred to a PDP 11/23 computer for use in a cardiac surgery operating theatres and intensive care unit. The FILE suite of programs are more easily transported to different units and have been given to several departments for independent evaluation.

(3) Use of microcomputers in experimentation.

The final example of the use of microcomputers has illustrated their ability to control events with a high degree of precision and to collect data automatically without observer bias. This particular project is still in its initial phase but the ability to alter programs to incorporate new features has been a major benefit compared with the greater complexity and rigidity of an analogue system. Data analysis by computer makes maximum use of the experimenter's time and, compared with the manual measurement of pen recordings, may be more accurate.

Many other applications are of importance although they have not been discussed. The text of this thesis was written by the author using the same type of Apple II microcomputer used for theother applications described. The University and NHS departments of Anaesthesia have four similar systems used for word processing. This has allowed maximum utilisation of scarce secreterial resources.

Another example is the development of techniques using light pens for nurses to select a standard comment or drug from a menu displayed on the screen. This technique have been developed and used for storage of anaesthetic and other medical records (fig. 57). because of the simplicity it offers for communication with the computer.

A further possible area of development is patient management using closed loop systems. Price and his colleagues (64) have shown that a computer-based closed loop sytem can control intracranial pressure more efficiently than medical or nursing staff.

Sheppard and colleagues (79) have reported the use of computer controlled infusion of sodium nitroprusside for maintaining arterial pressure at a predetermined value in more than 13 000 post cardiac by-pass patients. They also found that the computer system achieved better control than medical or nursing staff. Microcomputers are now sufficiently powerful to be used for such a purpose and more widespread application of these techniques can be expected.

II. Replacement of large centralised computers with inexpensive microcomputers.

The successful replacement of large centralised computers by small, inexpensive microcomputers for many functions has been demonstrated to be feasible and desirable. In particular, an attempt to compare the attitudes of students receiving CAL using a central computer with those using microcomputers was abandonned after 18 months because of continued failure of the central system.

Microcomputers are designed to be easy to use and, since they normally are required to perform one task at a time, their operating systems are simple to design and usually error free. Microcomputers are robust and have proven extremely reliable in routine use for teaching, in the operating theatre and intensive care unit.

Centralised computers have more complex operating systems which are more difficult to use and may be associated with more errors. A further advantage of microcomputers is that interfacing them to other equipment is usually simple because of the availability of a wide range of inexpensive interface cards.

The continuous increase in computing power and memory size of microcomputers will lead to more functions, which were previously the province of large systems, being undertaken by microcomputers. The continual decrease in the cost/performance ratio will frequently result in the microcomputer-based system costing less to purchase and operate than a centralised computer.

Network systems have been developed commercially and a further development of the Apple II computer applications in the intensive care unit and operating theatres would involve the use of a network such as the Corvus Omninet. This allows up to 64 separate computers to interface with Winchester disc drives of up to 80 megabytes capacity. Data is transferred to each satellite computer at a rate of 1.2 megabaud. This network system would enable all computers to access one single collection of programs as well as the information concerning other patients linked into the network and would provide much greater capacity for the long term storage of patient data. More rapid transfer of data and programs will allow greater flexibility to be achieved and programs to be improved.

The three applications of microcomputers described in this thesis would then be available to any satellite computer which was connected into the network without the need to insert the appropriate floppy discs. An initial selection menu would allow the user to receive instructions on the use of any of the programs and to select the program suite required. Such a system would enable tasks to be undertaken more simply and efficiently than with a large central computer. There would also be less chance of the breakdown of a single component causing the entire system to fail.

The only limitation would be the maximum memory size available for the analysis of large amounts of data. However, at least two of the new generation of 16-bit computers, which are available with 512 k bytes of memory, can be connected to the network and it would be possible to use these where large arrays of data were to be analysed.

Figure 57 Data entry using light pen.

III. Program development by non-computer experts

The improved performance of the hardware will have little effect unless adequate software is available. The author had no experience of computing before the start of the project and the programs described demonstrate a learning process from the simple CAL/CASA programs written in BASIC to the machine code routine for the CO2RESPONSE program.

A considerable effort was involved in the successful production of these programs and it is probably unrealistic to expect a doctor without an extremely high degree of interest and motivation to develop programs of such complexity.

However, it is likely that programs, such as CALWRITE, will be developed which will enable untrained users to produce programs for off-line use which are suitable for their own purposes. On-line programs will always be more difficult to develop but, where these are to be used in the ward or operating theatre, doctors and nurses with experience in producing and implementing programs should be involved. They must ensure that the final system fulfils the criteria described in chapter 3 of being simple to use and able to perform tasks better than can be performed without the computer. It is essential that the design of any computer-based system must closely involve those who will use the system since, what may appear an elegant engineering or programming solution, may be unacceptable when the system is used in the real situation.

The microcomputer is an invaluable tool which will be used increasingly for education and to improve the collection and management of patient data. It is important that medical staff become involved in computing projects to direct the exploitation of this new technology for the maximum benefit of their patients.

REFERENCES

1. Ahmed K, Ingram D, Dickinson CJ. Software for educational computing. Lancaster: MTP Press, 1981.

2. Alment EAJ. Competence to practice. London, Committee of Enquiry into competence to practice, 1976.

3. Altose MD, Kelsen SG, Stanley NN, Levinson RS, Cherniack NS, Fishman AP. Effects of hypercapnia on mouth pressure during airway occlusion in conscious man. J Appl Physiol 1976; 40: 338-343.

4. Anastasio EJ, Morgan JS. Factors inhibiting the use of computers in instruction. EDUCOM, Interuniversity Communications Council, Inc., Princeton: New Jersey, 1972.

5. Bailey JS, Wyatt R, Allwood JM. The nurse to computer interface In: Paul, JP, ed. Computing in Medicine, London: Macmillan, (in press).

6. Ballantine TVN, Proctor HJ, Broussard ND. The work of breathing: potential for clinical application and the results of studies performed on 100 normal males. Annal Surg 1970; 171: 590-594.

7. Bitzer MD, Bitzer DL. Teaching nursing by computers: an evaluative study. Comput Biol Med 1973; 3: 187-189.

8. Bloch R, Sweeney G, Ahmed K, Dickinson CJ, Ingram, D. MacDope: A simulation of drug disposition in the human body. Applications in clinical pharmacokinetics. Brit J Clin Pharmacol 1980; 10: 591-602.

9. Brown RL. Computerised nursing. Nursing Mirror 1976 Feb 12: 56.

10. Campbell D, Kenny GNC, Schmulian C, Davis P.D. Computer assisted self-assessment in anaesthesia: a preliminary study. Anaesthesia 1980; 35: 998-1002.

11. Casbergue J. Computer-assisted instruction in health professions education: guidelines for utilisation. In: DeLand EC, ed. Information Technology in Health Science Education. New York and London: Plenum Press, 1978; 63-87.

12. Christopher GR. The influence of a computer assisted learning experience upon the attitudes of school administrators. The Ohio State University PhD Thesis, 1969.

13. Christensen V. Respiratory depression after extradural morphine. Br J Anaesth 1980; 52, 841.

14. Cohn JD, Engler PE. Shunt effect of carboxyhemoglobin. Crit Care Med 1979; 7: 54-58.

15. Colley PS, Cheney FW. Sodium nitroprusside increases Qs/Qt in dogs with regional atelectasis. Anesthesiology 1977; 47: 338-341.

16. Davies IK. The reinforcement of learning. Times Educational Supplement 1963; Apr: 19.

17. Davis PD, Kenny GNC. Microcomputer control of a video cassette lecture. Med Ed 1980; 14: 196-198.

18. Davis PD, Kenny GNC, Campbell D. On-line analysis of respiratory function with a microcomputer. In: Paul JP, ed. Computing in Medicine, London: Macmillan, (in press).

19. Deutsch LS, Englelse, WAH, Van der Voorde F, Hugenholtz PG. The Unibed patient monitoring system: a new approach for a new technology. Med Instrum 1977; 11: 274-277.

20. Dickinson CJ. A computer model of human respiration. Lancaster: MTP Press, 1977.

21. Elridge FL. Relationship between respiratory nerve and muscle force output. J Appl Physiol 1973; 39: 567-574.

22. Feurzeig W, Munter P, Swets J, Breen M. Computer-aided teaching in medical diagnosis. J Med Educ 1964; 39: 746-754.

23. Foex P, Prys-Roberts C, Hahn CEW, Fenton M. Comparison of oxygen content of blood measured directly with values derived from measurements of oxygen tension. Br J Anaesth 1970; 42: 803-804P.

24. Forbes CD. Self-assessment - who needs it? Scot Med J 1979; 24: 185-186.

25. Gallagher TJ, Civetta JM, Kirby RR. Terminology update: optimum PEEP. Crit Care Med 1978; 6: 323-326.

26. Harden RM, Dunn WR, Murray TS, Rogers J, Stone C. Doctors accept a challenge: self-assessment exercises in continuing medical education. Br Med J 1979; 284: 652-653.

27. Harless WG, Drennon CG, Marxer JJ, Root JA, Miller GE. CASE: A computer-aided simulation of the clinical encounter. J Med Educ 1971; 46: 443-448.

28. Harless WG, Drennon CG, Marxer JJ, Root JA, Wilson LL, Miller GE. GENESYS - A generatimg system for the CASE natural language model. Comput Biol Med 1973; 3: 246-268.

29. Hemmer M, Suter PM. Treatment of cardiac and renal effects of PEEP with dopamine in patients with acute respiratory failure. Anesthesiology 1980; 50: 399-403.

30. Hess RD, Tenezakis MD. The computer as a socialising agent: Some socio-affective outcomes of CAI. AV Communication Review 1973; 21: 31-225.

31. Hilberman M, Kamm B, Tarter M, Osborn JJ. An evaluation of computer-based patient monitoring at Pacific Medical Center. Comp Biomed Res 1975; 8: 447-460.

32. Hilberman M, Kamm B, Lamy M, Dietrich HP, Martz K, Osborn JJ. An analysis of potential physiological predictors of respiratory adequacy following cardiac surgery. Thorac Cardiovasc Surg 1975; 71: 711-720.

33. Jardin F, Eveleigh MC, Gurdjian F, Delille F, Margairaz A. Venous admixture in human septic shock: comparative effects of blood volume expansion, dopamine infusion and isoproterenol infusion on mismatch of ventilation and pulmonary blood flow in peritonitis. Circulation 1979; 60: 155-159.

34. Jordan C. Automatic method for measuring mouth occlusion pressure response to carbon dioxide inhalation. Med Biol Eng Comput 1981; 19: 279-286.

35. Jonson B, Norstrom L, Olsson SG, Akerback D. Monitoring of ventilation and lung mechanics during automatic ventilation: a new device. Bull Physiopath Resp 1975; 11: 729-743.

36. Kenny GNC. Programmable calculator: a program for use in the intensive care unit. Br J Anaesth 1979; 51: 793-796.

37. Kenny GNC. Computer assisted learning in medical education. Proc 10th and 11th Annual Meetings of the National Association of Clinical Tutors. Update Journals. London: 1981; 39-41.

38. Kenny GNC. Which microcomputer? Br J Hosp Med 1982; 27: 74-78.

39. Kenny GNC, Campbell D. Intrapulmonary shunt. In: Spence AA, ed. Clinics in Critical Care Medicine: Respiratory Monitoring in Intensive Care. Edinburgh: Churchill Livingston, (in press).

40. Kenny GNC, Campbell D. Microcomputer assisted cardiovascular monitoring. In: Paul JP, ed. Computing in Medicine. London: Macmillan, (in press).

41. Kenny GNC, Davis PD. The use of a micro-computer in anaesthetic teaching. Anaesthesia 1979; 34: 583-585.

42. Kenny GNC, Davis PD. Computing techniques in critical care medicine. In: Ledingham IMcA, Hanning CD, eds. Recent advances in critical care medicine. Edinburgh: Churchill Livingstone, (in press).

43. Kenny GNC, Davis PD, Campbell D. Monitoring respiratory function with a microcomputer. Br J Anaesth 1979; 51: 996P.

44. Kenny GNC, Schmulian C. Computer assisted learning in the teaching of anaesthesia. Anaesthesia 1979; 34: 159-162.

45. Kryger M, McCullough E, Weil J. Aspects of technique which affect precision in mouth occlusion pressure measurements. Am Rev Respir Dis 1976; 113 Suppl: 246P.

46. Lewis FJ, Shimizu T, Scofield AL, Rosi PS. Analysis of respiration by an on-line digital computer system: clinical data following thoracoabdominal surgery. Ann Surg 1966; 164: 547-557.

47. Lopata M, Evanich MJ, Lourenco RV. Relationship between mouth occlusion pressure and electrical activity of the diaphram: effects of flow-resistive loading. Am Rev Respir Dis 1977; 116: 449-455.

48. Luskin BJ. An identification and examination of obstacles to the development of computer assisted instruction. University of California PhD Thesis, 1970.

49. Lynch JP, Mhyre JG, Dantzer DR. Influence of cardiac output on intrapulmonary shunt. J Appl Physiol 1979; 46: 315-321.

50. Mathews AW, Howell JBL. The rate of isometric inspiratory pressure development as a measure of responsiveness to carbon dioxide in man. Clin Sci Molec Med 1975; 49: 57-68.

51. Merola AJ, Pengov RE, Stokes BT. Computer-supported independent study in the basic medical sciences. In: DeLand EC, ed. Information Technology in Education. New York and London: Plenum Press, 17-43.

52. Miller J, Preston TD, Dann PE, Bailey JS, Tobin G. Charting v computers. Nursing Times 1978; Aug 24: 1423-1425.

53. Murray TS, Cupples RW, Barber JH, Dunn WR, Scott DB, Hannay DR. Teaching decision making to medical undergraduates by computer-assisted learning. Med Ed 1977; 11: 262-264.

54. Murray TS, Barber JH, Dunn WR. The potential of computer-assisted learning in medical education. J R Coll Physicians 1977; 11: 401-404.

55. National Heart, Lung and Blood Institute. Extracorporeal support for respiratory insufficiency. U.S. Dept. of Health, Education and Welfare, Public Health Service, National Institutes of Health, 1979.

56. Neilson CH, Camporesi EM, Bromage PR. CO2 sensitivity after epidural and IV morphine. Anesthesiology 1981; 55: A372.

57. Norwood DD, Hawkins RE, Gall JE. Information system benefits hospital improves patient care. Hospitals, J.A.H.A. 1976; 50: 79-83.

58. Osborn JJ, Beamont JO, Raison JCA, Russell J, Gerbode F. Patient monitoring by digital computer. Surgery 1969; 64: 1057-1070.

59. Peters RM, Hilberman M. Respiratory insufficiency: diagnosis and control of therapy. Surgery 1971; 70: 280-287.

60. Peters RM, Hilberman M, Hogan JS, Crawford DA. Objective indications for respiratory therapy in post-trauma and postoperative patients. Annal J Surg 1972; 124: 262-269.

61. Pollock R, Gregory NL, Shaw A. Improved technique for recording respiratory data. Med Biol Eng Comput 1977; 15: 541-547.

62. Prakash O, Meij S, Bos E, Frederiksh PA, Jonson B. Lung mechanics in patients undergoing mitral valve replacement: the value of monitoring of compliance and resistance. Crit Care Med 1978; 6: 370-372.

63. Prakash O, van der Borden B, Meij S. On-line presentation of lung mechanics during mechanical ventilation. In: Payne JP, ed. Proc Computers in critical care and pulmonary medicine. (in press).

64. Price DJ, Digdale RE, Mason J. The control of ICP using three asynchronous closed loops. In: Shulman K, ed. Intracranial pressure IV. Heidelberg: Springer-Verlag, 1980; 395-399.

65. Proctor HJ, Ballantine TVN, Broussard ND, Hirsh EA. An analysis of pulmonary function following penetrating pulmonary injury with recommandations for therapy. Surgery 1970; 68: 92-98.

66. Proctor HJ, Woolson R. Prediction of respiratory muscle fatigue by measurements of the work of breathing. Surg Gynaec Obstet 1973; 136: 367-370.

67. Raison JCA, Beaumont JO, Russell JAG, Osborn JJ, Gerbode F. Alarms in an intensive care unit: an interim compromise. Comput Biomed Res 1968; 1: 556-564.

68. Rampil IJ, Flemming DC. An inexpensive micro-computer trend monitoring system. Proc New Engl Bioeng Conf 1978; 6: 76-70.

69. Read DJC. A clinical method for assessing the ventilatory response to CO2. Australas Ann Med 1967; 16: 20-32.

70. Reiz S, Westberg M. Side effects of epidural morphine. Lancet 1980; 2: 203-204.

71. Rombardey CP. A study of selected Michigan elementary and secondary teachers' and Principals' attitudes toward computer assisted instruction. Michigan State Universty PhD Thesis, 19

72. Royal Commisson on the National Health Service. Report. Cmnd 7615. London: HMSO, 1979.

73. Ruiz BC, Tucker WK, Kirby RR. A program for calculation of intrapulmonary shunt, Blood-gas and acid-base values with a programmable calculator. Anesthesiology 1978; 42: 88-95.

74. Sarnat AJ, Quinn ML, Ty Smith N. A preliminary evaluation of computerised speech recognition for anesthesia. Anesthesiology 1981; 55 Suppl: A128.

75. Schmulian C, Kenny GNC, Campbell D. Use of microcomputers for self-assessment and continuing education in anaesthesia. Br Med J 1982; 284: 403-405.

76. Scottish Council for Postgraduate Medical Education. Maintaining standards in general practice. Edinburgh: SCPME, 1981.

77. Shabot MM, Shoemaker WC, State D. Rapid bedside computation of cardiovascular variables with a programmable calculator. Crit Care Med 1977; 5: 105-111.

78. Shaw CD. Aspects of audit. Br Med J 1980; 280: 1256-8, 1314-6, 1361-3, 1443-6, 1509-11.

79. Sheppard LC, Kouchoukos NT, Shotts JF, Wallace FD, Regulation of mean arterial pressure by computer control of vasoactive agents in postoperative patients. Computers in Cardiology 1975; IEEE 75CH1018-C: 91-94.

80. Sheppard LC. The computer in the care of critically ill patients. Proc IEEE 1979; 67: 1300-1306.

81. Siafakas N, Morris AJR, Prime FJ. The rate of change of mouth occlusion pressure during exercise. Clin Sci 1979; 56: 455-461.

82. Simpson MA. Medical education: a critical approach. London: Butterworths, 1972.

83. Starkweather J, Kamp M. A self-contained CAI machine for health science education. In: DeLand EC, ed. Information Technology in Health Science Education. New York and London: Plenum Press, 1978; 217-228.

84. Stone JG, Khambatta HJ, Matteo RS. Pulmonary shunting during anesthesia with deliberate hypotension. Anesthesiology 1976; 45: 508-515.

85. Suter PM, Fairley HB, Isenberg MD. Optimum end-expiratory airway pressure in patients with acute pulmonary failure. N Eng J Med 1975; 292: 284-289.

86. Tebbutt D. The case against computer literacy. Personal Computer World 1982; 5: 137.

87. Tompkins WJ, Webster JG. Signal processing - hardware versus software. In: Tompkins WJ, Webster JG eds. Design of Microcomputer-Based Medical Instrumentation. Englewood Cliffs: Prentice-Hall, 1981; p91-168.

88. Toong HD. Microprocessors. Sci Am 1977; 273: 146-161.

89. Turnley SZ, Blumenfeld W. On-line respiratory analysis using a digital calculator. Med Biol Eng 1973; 11: 275-285.

90. Tuttle JG. The historical development of computer capabilities which permitted the use of the computer as an educational medium in the United States from 1958 to 1968, with implications of trends. New York University PhD Thesis, 1970.

91. Voss G, Weidenhammer J, Renschler HE. Computer-assisted instruction in medicine: a German view. Proc. Conf New Directions in Educational Computing. Association for the Development of Computer-Based Instruction Systems, 1978; 63-72.

92. Votaw RG, Farquhar BB. Current trends in computer-based education in medicine. Educational Technology 1978; Apr, 54-56.

93. Walton HJ. Small group methods in medical education. ASME Medical education, Booklet No 1, 1973.

94. Wax SD, Webb WR, Osburn KC. Microprocessor-based pulmonary mechanics instrumentation. Proc Annu Conf Eng Med Biol 1978; 20: 231.

95. Wirth N, Jensen K. Pascal: user manual and report. New York: Springer-Verlag, 1975.

96. Wyatt R. Long term extracorporeal membrane oxygenation (ECMO). In: Longmore DB, ed. Towards safer cardiac surgery. Lancaster: MIP Press, 475-482.

97. Whitelaw WA, Dereene JP, Milic-Emili J. Occlusion pressure as a measure of respiratory center output in conscious man. Respir Physiol 1975; 23: 181-199.

98. Wilson RS. Monitoring the lung: mechanics and volume. Anesthesiology 1976; 45: 135-147.

99. Zubillaga G, Lopata M, Evanich MJ, Lourenco RV. Diaphragmatic EMG (EMGdi) response to isocapnic hypoxia in normal man: comparison to hyperoxic hypercapnia. Am Rev Respir Dis 1976; 113 Suppl: 217.

100. Zuck D. Use of microcomputers for self-assessment and continuing education. Br Med J 1982; 284: 1268.

GLASGOW UNIVERSITY LIBRARY

APPLICATION OF MICROCOMPUTERS
TO ANAESTHESIA AND INTENSIVE CARE

VOLUME II of II VOLUMES

GAVIN NC KENNY, BSc, MBChB, FFARCS
University Department of Anaesthesia
Glasgow Royal Infirmary

Thesis submitted to the University of Glasgow
for the degree of Doctor of Medicine

July, 1982

Thesis
6561
Copy 2

GLASGOW
UNIVERSITY
LIBRARY

CONTENTS - VOLUME II

INTRODUCTION

All programs were written in APPLESOFT BASIC except the machine code programs which were written using the APPLE 6502 editor/assembler. Program listings were prepared with the same word processor program used for the main text. The programs were listed to a text file on disc and edited with Apple Pie. The program files were transferred to a 5 megabyte Corvus disc drive from which they called each other as required.

As explained in the main text, when the second page of graphics is used, the program must be less than 16 k bytes in length. Extra REM statements have been included in many of the programs to assist in documentation, and in some cases, this has increased the program size above 16 k.

Some of these statements must be removed before the programs will perform satisfactorily and application should be made to the author for suitable copies of the discs.

```
1    REM **************
2    REM * CALOB-TEXT *
3    REM **************
4    POKE  - 16368,0: REM * CLEAR KEYBOARD BUFFER *
5    TEXT : DIM Q$(55) : DIM T$(18) : PRINT D$;"MONI,C,O":
6    HOME :I = 9: REM * THERE ARE 8 QUES IN PREV PROG *
7    GOSUB 9000
8    REM ***********************************************
9    REM * ALLOCATE T$(X) ARRAY FOR FIRST PAGE OF TEXT *
10   REM ***********************************************
11   T$(1) = "   YOU ARE REQUIRED TO ANAESTHETISE A   30 YEAR
     OLD PARA 1+1 AT 38 WEEKS         GESTATION IN WHOM A
     DIAGNOSIS OF
12   T$(2) = "                                        "
15   T$(3) = "   FOETAL DISTRESS IS PRESENT AND THE
     DECISION TO PERFORM CAESAREAN SECTION   HAS BEEN TAKEN.
     SHE HAS RECEIVED 100 MG PETHIDINE.  HER PULSE RATE IS
     120/MIN."
20   T$(4) = "HER ARTERIAL PRESSURE IS 105/70 MMHG.":T$(6) = "
     HER HAEMOGLOBIN CHECKED 1 WEEK        PREVIOUSLY AT THE
     ANTENATAL WAS 12 G%.
22   T$(7) = "THERE IS NO SIGN OF OVERT VAGINAL
     BLEEDING."
24   T$(8) = "                                        "
26   T$(5) = "                                        "
30   T$(9) = "   ON ADMISSION SHE WAS GIVEN 15ML      ANTACID
     ORALLY AND PRIOR TO TRANSFER TO THEATRE SHE RECEIVED
     ANOTHER 30ML
31   REM *************************
32   REM * PRINT FIRST TEXT PAGE *
33   REM *************************
35   FOR T = 1 TO 9: PRINT T$(T): NEXT
36   T = 9
40   GOSUB 90: GOTO 97
80   REM **************************************
85   REM * WAIT UNTIL '1' IS TYPED ON KEYBOARD *
87   REM **************************************
90   PRINT : PRINT "WHEN READY TYPE 1 ";
```

```
91   GET Z$: IF Z$ <  > "1" THEN 95
92   HOME : RETURN
94   REM ***************************************
95   REM * ALLOCATE MAIN TOPIC QUESTION TO G$ *
96   REM ***************************************
97   G$ = "THE MOST EFFECTIVE ANTACID IS ?
98   H$ = ""
99   REM ***************************************************
100  REM * SET 'F' = 1 TO INDICATE ONLY 1 REPLY REQUIRED *
101  REM ***************************************************
102  F = 1
106  REM **********************************
105  REM * ALLOCATE CORRECT SCORE TO 'B' *
106  REM **********************************
109  B = 5
120  REM *******************************************
130  REM * ALLOCATE QUESTION TO BE SCORED TO 'X$' *
135  REM *******************************************
140  X$ = "1=MIST. MAG. TRISILICATE B.P.C.         2=MAGNESIUM
     TRISILICATE                       3=ALUDROX
144  REM *******************************************
145  REM * ALLOCATE FEEDBACK STRINGS TO A$ AND B$ *
146  REM *******************************************
150  A$ = "MIST. MAG. TRISILICATE B.P.C. CONTAINS  SODIUM
     BICARBONATE AND IS MORE POTENT   THAN MAGNESIUM
     TRISILICATE IN RAISING PH
170  B$ = "                                             ALTHOUGH
     ALUDROX IS A MORE PALATABLE    ANTACID, IT IS LESS
     EFFICIENT IN        NEUTRALISING GASTRIC ACID
175  REM **********************************************
180  REM * GOTO 'SINGLE REPLY' TEXT/QUESTION HANDLER *
185  REM **********************************************
190  GOSUB 50001:B$ = ""
195  HOME : VTAB 8: HTAB 1: PRINT "THE NEXT SECTION WILL
     PRESENT SEVERAL   QUESTIONS TO BE SCORED.": PRINT : PRINT
     : PRINT "YOU WILL BE ASKED TO SCORE EACH QUESTIONIN TURN
     AND WILL BE GIVEN FEEDBACK AFTEREACH REPLY.
197  GOSUB 90
```

```
200  HOME
210  G$ = "WHICH OF THE FOLLOWING REDUCE THE DANGEROF
     REGURGITATION DURING INDUCTION OF    ANAESTHESIA ?
215  REM * ALLOCATE INDIVIDUAL QUESTION STRINGS TO X$(A) ARRAY *
216  REM *           CORRECT SCORES TO B(A) ARRAY                *
217  REM *           VERTICAL TAB POSITIONS TO V(A) ARRAY
218  REM *           FEEDBACK STRINGS TO A$(A)-C$(A) ARRAYS      *
220 X$(1) = "1. METOCLOPRAMIDE
225 B(1) = 2
230 A$(1) = "METOCLOPRAMIDE APPEARS TO BE INEFFECTIVEIN
    PATIENTS WHERE NARCOTICS HAVE BEEN   ADMINISTERED AND DOES
    NOT REDUCE THE    DANGER OF REGURGITATION.
235 V(1) = 5
240 X$(2) = "2. ATROPINE
245 B(2) = 2
250 A$(2) = "RECENT EVIDENCE SUGGESTS THAT ATROPINE  DOES NOT
    INCREASE THE TONE OF THE GASTROOESOPHAGEAL SPHINCTER.
255 V(2) = 7
260 X$(3) = "3. ALKALI
265 B(3) = 4
270 A$(3) = "ALKALINISATION OF THE GASTRIC CONTENTS  INCREASES
    THE TONE OF THE GASTRO-       OESOPHAGEAL SPHINCTER.
280 V(3) = 9
290 X$(4) = "4. A HISTORY OF PREVIOUS GASTRIC REFLUX    DURING
    PREGNANCY
295 B(4) = 1
298 V(4) = 11
300 A$(4) = "IN PATIENTS WHO SUFFER FROM HEARTBURN   DURING
    PREGNANCY THE INTRA-GASTRIC      PRESSURE IS ELEVATED TO
    THE EXTENT       "
310 B$(4) = "USUALLY ASSOCIATED WITH PREGNANCY.
                                        "
312 C$(4) = "HOWEVER, THE SPHINCTER TONE IS NOT      INCREASED
    AND THEREFORE THERE IS A      GREATER CHANCE OF
    REGURGITATION PAST THESPHINCTER.
320 V(5) = 14
325 B(5) = 1
330 X$(5) = "5. THE USE OF A DEPOLARISING AGENT SUCH    AS
```

```
   SUXAMETHONIUM
340 A$(5) = "SUXAMETHONIUM MAY IN SOME PATIENTS     PRODUCE A
   SHARP RISE IN INTRA-GASTRIC   PRESSURE.
355 J = 5
360  GOSUB 30000
365  FOR Z = 1 TO 5:A$(Z) = "":B$(Z) = "":C$(Z) = "": NEXT
370  HOME
400 G$ = "THE DURATION OF PRE-OXYGENATION REQUIREDIS ?
410 F = 1:B = 2:X$ = "1=ONE MIN   2=TWO MIN   3=THREE MIN"
415 A$ = "AT LEAST 3 MIN PRE-OXYGENATION IS        "
417 B$ = "REQUIRED TO ENSURE ADEQUATE WASHOUT OF "
418 C$ = " NITROGEN FROM ALVEOLI THEREBY "
419 D$ = "INCREASINGTHE ARTERIAL PO2
420  GOSUB 50001:F = 0
430 B$ = "":C$ = "":D$ = ""
440  HOME
450 G$ = "DURING APNOEA A PREGNANT WOMAN WILL HAVEA MORE RAPID
   FALL IN ARTERIAL PO2 THAN ANON-PREGNANT WOMAN.
460 F = 1:B = 5
470 X$ =  CHR$ (13) + "1=TRUE    2=FALSE
475 A$ = "AFTER PRE-OXYGENATION THE REDUCTION IN  ARTERIAL PO2
   IS SUBSTANTIALLY GREATER   AND FASTER IN PREGNANT WOMEN AT
   TERM    THAN IN NON-PREGNANT WOMEN.
480  GOSUB 50001:F = 0
550 G$ = "PRE-OXYGENATION SHOULD BE CARRIED OUT IN":H$ =
   "WHICH OF THE FOLLOWING POSITIONS ?
555 H$ = ""
560 X$(1) = "1. SUPINE
565 B(1) = 1:V(1) = 4
570 A$(1) = "IN THE SUPINE POSITION THE GRAVID UTERUSOBSTRUCTS
   THE INFERIOR VENA CAVA AND MAYDECREASE CARDIAC OUTPUT BY
   UP TO 50%
580 B(2) = 3:V(2) = 6
590 X$(2) = "2. RIGHT LATERAL TILT
600 A$(2) = "RIGHT LATERAL TILT IS PROBABLY LESS     EFFECTIVE
   THAN LEFT LATERAL TILT IN     RELIEVING CAVAL OBSTRUCTION.
610 B(3) = 5:V(3) = 8
615 X$(3) = "3. LEFT LATERAL TILT
```

```
617 A$(3) = "LEFT LATERAL TILT OF 10 DEGREES
    RELIEVESOBSTRUCTION OF INFERIOR VENA CAVA
640 J = 3: GOSUB 30000
650  HOME : PRINT : PRINT
655 H$ = ""
660  PRINT "THE PATIENT IS GIVEN ATROPINE 0.6MG AND
     THIOPENTONE 250MG FOLLOWED BY           SUXAMETHONIUM
     100MG.
670  GOSUB 90
680 G$ = "WHEN SHOULD CRICOID PRESSURE BE APPLIED (SELLICK'S
    MANOUVRE) ?
690 F = 1:B = 1
700 X$ = "1=AFTER SUXAMETHONIUM HAS TAKEN EFFECT  2=BEFORE
    INDUCTION                          3=FOLLOWING THIOPENTONE
710 A$ = "   APPLICATION OF CRICOID PRESSURE      BEFORE
    INDUCTION MAY PRODUCE POTENTIALLYDANGEROUS MOVEMENTS BY
    THE PATIENT      UNLESS EXPLANATION HAS BEEN GIVEN.
720 B$ = " CRICOID PRESSURE APPLIED FOLLOWING      INJECTION
    OF SUXAMETHONIUM WILL NOT     GUARD AGAINST SUDDEN RISE IN
    INTRA-     GASTRIC PRESSURE WHICH MAY OCCUR.
740 C$ = "  CORRECTLY APPLIED CRICOID PRESSURE CAN  PREVENT
    REGURGITATION WHEN THE INTRA-   GASTRIC PRESSURE IS AS
    HIGH AS 94 CMH2O.
760  GOSUB 50001:F = 0:B$ = "":C$ = ""
770 G$ = "   INTUBATION PROVES DIFFICULT WITH AN  8MM
    ENDOTRACHEAL TUBE (ETT).
780 H$ = "WHAT ACTION DO YOU TAKE ?
820 B(1) = 1:X$(1) = "1.GIVE A SECOND DOSE OF
    SUXXAMETHONIUM":V(1) = 6
830 A$(1) = "A SECOND DOSE OF SUXAMETHONIUM WILL     PROLONG
    APNOEA TO A POINT WHERE CEREBRALHYPOXIA WILL OCCUR UNLESS THE
    LUNG
840 B$(1) = "    CONTINUING ANAESTHESIA WITH SPONTANEOUS
    RESPIRATION IS A SAFER OPTION.
860 B(2) = 1:X$(2) = "2.GIVE A LONG ACTING RELAXANT SUCH AS
    PANCURONIUM":V(2) = 8
862 A$(2) = "A LONG-ACTING RELAXANT WILL PROLONG     APNOEA TO
    A POINT WHERE CEREBRAL HYPOXIAWILL OCCUR UNLESS THE LUNGS CAN
```

```
BE VENTILATED WITH OXYGEN."
865 B$(2) = "    CONTINUING ANAESTHESIA WITH SPONTANEOUS
    RESPIRATION IS A SAFER OPTION.
880 B(3) = 5:X$(3) = "3.ATTEMPT INTUBATION WITH A 7MM ETT AND
    INTRODUCER":V(3) = 11
882 A$(3) = "INTUBATION SHOULD BE ATTEMPTED WITH A   SMALLER
    ETT USING AN INTRODUCER TO      MAINTAIN CURVATURE.
884 B$(3) = "                       IT IS IMPORTANT TO MAINTAIN
    OXYGENATION WHILE ATTEMPTING TO INTUBATE.
886 J = 3: GOSUB 30000
887  GOSUB 33000
888 G$ = "IF THE 7MM ETT COULD NOT BE PASSED, WHATWOULD BE THE
    BEST COURSE OF ACTION ?
889 H$ = ""
890 B(1) = 3:V(1) = 4
895 X$(1) = "1.GIVE OXYGEN BY MASK AND ALLOW THE       PATIENT
    TO WAKEN.
900 A$(1) = "TO ALLOW THE PATIENT TO WAKEN IS A      POSSIBLE
    OPTION. THE MOTHER WOULD BE    SAFEGUARDED BUT THE FOETUS IS
    IN DANGER.
920 X$(2) = "2.VENTILATE THE LUNGS WITH OXYGEN BY      MASK
    AND THEN CONTINUE ANAESTHESIA      WITH THE PATIENT BREATHING
925 B(2) = 4:V(2) = 7
930 A$(2) = "IF INTUBATION FAILS ANAESTHESIA MUST BE
    MAINTAINED WITH A VOLATILE AGENT SUCH ASHALOTHANE, AND CRICOID
    PRESSURE METICULOUSLY MAINTAINED.
940 B$(2) = "                  "
942 C$(2) = "SPONTANEOUS RESPIRATION, WHEN IT RETURNSSHOULD BE
    CONTINUED.
945 J = 2
950  GOSUB 30000
960  GOSUB 33000
965  HOME
970 T$(11) = "   A 7MM ETT IS EVENTUALLY PASSED AND    THE
    PATIENT'S LUNGS VENTILATED WITH 100%OXYGEN TO ENSURE AN
    ADEQUATE ARTERIAL PO2.
980 T$(12) = "                                        "
990 T$(13) = "ANAESTHESIA IS THEN MAINTAINED WITH     OXYGEN
```

```
   50%, NITROUS OXIDE 50%, AND      HALOTHANE 0.5%."
1000 T$(14) = "                                           "
1010 T$(15) = "WHEN RECOVERY FROM SUXAMETHONIUM IS
     APPARENT, PANCURONIUM 5MG IS          ADMINISTERED."
1015  FOR T = 11 TO 15: PRINT T$(T): NEXT
1016 T = 15
1020  GOSUB 90
1030 G$ = "A LEAK IS NOTICED PAST THE ETT WHICH    CANNOT BE
     CORRECTED BY FURTHER INFLATIONOF THE CUFF.
1040 H$ = "WHAT ACTION DO YOU TAKE ?
1045 B(1) = 1:V(1) = 6
1050 X$(1) = "1.CHANGE THE ETT
1060 A$(1) = "INTUBATION HAS PROVED DIFFICULT. IF THE ETT IS
     REMOVED, IT MAY BE IMPOSSIBLE TO REPLACE IT.
1070 B(2) = 5:V(2) = 8
1080 X$(2) = "2.PACK AROUND THE ETT WITH A THROAT PACK
1090 A$(2) = "PACKING AROUND THE ETT MAY NOT ENSURE A GAS
     TIGHT SEAL BUT WILL HELP TO DECREASETHE LEAK.
1100 B(3) = 5:V(3) = 11
1110 X$(3) = "3.INCREASE THE GAS FLOW TO THE PATIENT    TO
     ACHIEVE AN ADEQUATE EXPIRED MINUTE   VOLUME
1120 A$(3) = "GAS FLOW SHOULD BE INCREASED AND EXPIREDMINUTE
     VOLUME MEASURED WITH A WRIGHT'S  RESPIROMETER.
1130 J = 3: GOSUB 30000: GOSUB 33000
1140  HOME
1150 G$ = "AFTER DELIVERY OF THE INFANT, WHICH     UTERINE
     STIMULANT SHOULD BE EMPLOYED ?
1155 H$ = ""
1157 B(1) = 3:V(1) = 4
1160 X$(1) = "1.ERGOMETRINE 0.5MG I.V.
1170 A$(1) = "ERGOMETRINE COULD BE USED BUT PRODUCES  BOTH
     ARTERIAL AND VENOUS SPASM."
1180 B(2) = 3:V(2) = 6
1190 X$(2) = "2.SYNTOMETRINE 1ML I.M.
1200 A$(2) = "SYNTOMETRINE IS AS EFFECTIVE AS
     SYNTOCINON BUT HAS ONLY SLIGHTLY LESS   VASOCONSTRICTOR
     ACTIVITY THAN ERGOMETRINE.
1210 B(3) = 5:V(3) = 8
```

```
1220 X$(3) = "3.SYNTOCINON 10 UNITS I.V. AND
     SYNTOCINON INFUSION
1230 A$(3) = "SYNTOCINON 10 UNITS I.V. AND SYNTOCINON INFUSION
     IS THE REGIME OF CHOICE.
1240 J = 3: GOSUB 30000: GOSUB 33000
1250  HOME : PRINT "   TWENTY PER CENT OF PATIENTS WITH
      ABRUPTIO PLACENTA DEVELOP A COAGULATION DEFECT.": PRINT
1260  PRINT "OF THESE PATIENTS, A QUARTER EXHIBIT    EXCESSIVE
      INTRAPARTUM  OR POSTPARTUM    BLEEDING.
1270  GOSUB 90
1272 G$ = "HOW WOULD YOU MANAGE THIS THEORETICAL   CLINICAL
     PROBLEM ?
1273 V(1) = 4:V(2) = 8:V(3) = 11:V(4) = 13:V(5) = 15
1274 B(1) = 5
1275 X$(1) = "1.CHECK BLOOD FOR FIBRIN DEGRADATION
     PRODUCTS (FDP'S) AND ARRANGE            COAGULATION SCREEN
1277 A$(1) = "THE BLOOD SHOULD BE CHECKED FOR FDP'S   AND A
     COAGULATION SCREEN PERFORMED TO   DETERMINE WHETHER DIFFUSE
     INTRAVASCULAR COAGULATION (DIC) IS OCCURING.          "
1278 B$(1) = "DIC IS A DYNAMIC PROCESS AND RESULTS    MUST BE
     INTERPRETED WITH CAUTION.
         "
1279 C$(1) = "RAPID TRANSFUSION AND DELIVERY MAY      PREVENT
     THE PROGRESSION OF THE BLEEDING DEFECT.
1285 B(2) = 5
1290 X$(2) = "2.REPLACE ESTIMATED BLOOD LOSS WITH       FRESH
     WHOLE BLOOD.
1300 A$(2) = "TRANSFUSION OF FRESH BLOOD SHOULD BE
     INSTIGATED AS SOON AS POSSIBLE SINCE    THE DEGREE OF BLOOD
     LOSS MAY NOT REFLECTTHE DEFICIT DUE TO RETROPLACENTAL
     HAEMORRAGE.
1310 B(3) = 5
1320 X$(3) = "3.CONSIDER ADMINISTRATION OF FIBRINOGEN
1330 A$(3) = "FIBRINOGEN SHOULD BE GIVEN IF SEVERE    BLEEDING
     OCCURS.
1335 B(4) = 5
1400 X$(4) = "4.MONITOR C.V.P.
1410 A$(4) = "CENTRAL VENOUS PRESSURE SHOULD BE
```

```
    MONITORED AS A GUIDE TO FLUID          REPLACEMENT.
1420 J = 4:JX = 0: GOSUB 30000: GOSUB 33000
1422 V(1) = 4:V(2) = 6:V(3) = 8
1425 B(1) = 5
1430 X$(1) = "5.MONITOR HOURLY URINE OUTPUT
1440 A$(1) = "HOURLY URINE VOLUME MUST BE MEASURED TO DETECT
    ANY RENAL IMPAIRMENT FOLLLOWING  HYPOPERFUSION OR AS A
    CONSEQUENCE OF    INTRAVASCULAR COAGULATION.
1450 B(2) = 1
1460 X$(2) = "6.ADMINISTER EPSILON AMINO CAPROIC ACID"
1470 A$(2) = "EPSILON AMINO CAPROIC ACID SHOULD NOT
    BEADMINISTERED SINCE IT MAY AGGRAVATE DIC.
1480 B(3) = 5
1490 X$(3) = "7.CHECK BLOOD GASES AND ARRANGE CHEST     X-RAY
    LATER
1500 A$(3) = "BLOOD GAS ANALYSIS AND CHEST X-RAY MUST BE
    PERFORMED TO DETECT WHETHER THE LUNGSHAVE BEEN AFFECTED BY
    D.I.C.
1510 J = 3:JX = 4: GOSUB 30000: GOSUB 33000:JX = 0
1520  HOME : PRINT : PRINT : PRINT : PRINT "ALTHOUGH THIS
     PATIENT DID NOT ASPIRATE  GASTRIC CONTENTS, THIS IS A MAJOR
     COMPLICATION OF OBSTETRIC ANAESTHESIA.
1530  GOSUB 90
1540 G$ = "WHAT ACTION SHOULD YOU TAKE IF          ASPIRATION
    OCCURS ?
1542 V(1) = 4:V(2) = 6:V(3) = 8:V(4) = 10:V(5) = 13
1545 B(1) = 5
1550 X$(1) = "1.TILT TABLE TO HEAD-DOWN POSITION
1560 A$(1) = "RAPID TILTING OF THE TABLE WILL ASSIST  THE
    GRAVITATIONAL DRAINAGE OF GASTRIC   CONTENTS."
1570 B(2) = 5
1580 X$(2) = "2.APPLY SUCTION TO THE PHARYNX
1590 A$(2) = "SUCTION IS ESSENTIAL TO FACILITATE      REMOVAL
    OF GASTRIC CONTENTS FROM AIRWAY.
1600 B(3) = 5
1610 X$(3) = "3.RAPIDLY PASS ETT
1620 A$(3) = "WHENEVER POSSIBLE AN ETT MUST BE PASSED WITHOUT
    DELAY TO PROTECT THE AIRWAY AND PREVENT FURTHER ASPIRATION
```

```
     OF GASTRIC   CONTENTS SHOULD THIS OCCUR.
1640 B(4) = 1
1650 X$(4) = "4.INCREASE OXYGEN TO 100% AND INFLATE     THE
     LUNGS IMMEDIATELY"
1660 A$(4) = "INJUDICIOUS EARLY VENTILATION WITH      OXYGEN
     BEFORE SUCTION THROUGH ETT MAY   CAUSE THE GASTRIC ASPIRATE
     TO PASS      FURTHER DOWN THE RESPIRATORY PATHWAYS."
1670 B(5) = 4
1680 X$(5) = "5.ADMINISTER METHYLPREDNISOLONE 1G T.I.D  FOR 48
     HOURS"
1690 A$(5) = "ALTHOUGH CONTROVERSIAL STEROIDS IN LARGEDOSES
     ARE ADVISABLE TO ATTEMPT TO REDUCEPULMONARY DAMAGE."
1700 J = 5:JX = 0: GOSUB 30000: GOSUB 33000
1701 B(1) = 1:V(1) = 4:V(2) = 6:V(3) = 8:V(4) = 10:V(5) =
     12:V(6) = 14
1702 X$(1) = " 6.ADMINISTER INTRATRACHEAL STEROIDS
1704 A$(1) = "INTRATRACHEAL STEROIDS HAVE NO
     ADVANTAGE OVER SYSTEMIC ADMINISTRATION."
1705 B$(1) = "                                        THEY
     MAY ALSO CAUSE PULMONARY DAMAGE."
1706 B(2) = 1
1710 X$(2) = " 7.PERFORM BRONCHIAL LAVAGE"
1720 A$(2) = "THERE IS LITTLE EVIDENCE THAT BRONCHIAL LAVAGE
     IS OF BENEFIT.
         "
1730 B$(2) = "IT MAY DISSEMINATE THE GASTRIC CONTENTS DOWN THE
     AIRWAYS AND DEPLETE PULMONARY  SURFACTANT."
1740 B(3) = 4
1750 X$(3) = " 8.ADMINISTER ANTIBIOTICS"
1760 A$(3) = "ADMINISTRATION OF ANTIBIOTICS IS NOT
     IMMEDIATELY NECESSARY BUT SINCE        INFECTION FREQUENTLY
     SUPERVENES AN      APPROPRIATE BROAD SPECTRUM ANTIBIOTIC
     SHOULD BE GIVEN."
1770 B(4) = 5
1780 X$(4) = " 9.MONITOR PO2 FREQUENTLY
1790 A$(4) = "BLOOD GAS ANALYSIS SHOULD BE FREQUENTLY
     PERFORMED TO MONITOR GAS EXCHANGE.
1800 B(5) = 4
```

```
1810 X$(5) = "10.ARRANGE CHEST X-RAY
1820 A$(5) = "CHEST X-RAY SHOULD BE CARRIED OUT
     FOLLOWING RECOVERY FROM ANAESTHESIA TO  DEMONSTRATE WHETHER
     GROSS PULMONARY     CHANGES HAVE OCCURRED."
1830 B(6) = 5
1840 X$(6) = "11.CONSIDER TRANSFER OF PATIENT TO
     RESPIRATORY INTENSIVE CARE UNIT"
1850 A$(6) = "SUCH PATIENTS SHOULD BE TREATED IN
     RESPIRATORY INTENSIVE CARE UNITS SINCE  THEY REQUIRE CAREFUL
     MONITORING AND MAY REQUIRE INTERMITTENT POSITIVE PRESSURE
      VENTILATION."
1860 J = 6:JX = 5: GOSUB 30000: GOSUB 33000:JX = 0
1870  HOME : VTAB 10: HTAB 1: PRINT "YOUR SCORE IS ";C%
1875  PRINT :P =  INT (C% / (C% + W%) * 100 + .5)
1880  PRINT "OUT OF ";C% + W%" = ";P;"%"
1885 Q$(I) =  STR$ (P)
1890  PRINT
1900 D$ =  CHR$ (4)
1905  GOSUB 90
2000  HOME : VTAB 5: PRINT "HAVE YOU USED THIS OBSTETRIC
      PROGRAMME ": PRINT : PRINT "BEFORE ?   Y/N"
2010  VTAB 10: PRINT "PLEASE TYPE 'Y' IF YOU HAVE": PRINT :
      PRINT "TYPE 'N' IF NOT"
2020  VTAB 20: GET A$: IF A$ = "N" OR A$ = "Y" THEN Q$(I + 1)
      = A$: GOTO 19000
2030  GOTO 2020
```

```
8985  REM **************************
8990  REM * EXPLAIN SCORING SYSTEMS*
8995  REM **************************
9000  HOME
9020  VTAB 8
9040  PRINT : PRINT "IN THIS PROGRAMME THE SCORING SYSTEM IS-"
9045  PRINT
9050  PRINT "1=MUST DO OR CONSIDER ABSOLUTELY CORRECT"
9070  PRINT "2=A POSSIBLE COURSE OF ACTION             OR
      POSSIBLY CORRECT"
9080  PRINT : PRINT "3=SHOULD NOT DO OR SHOULD CONSIDER WRONG"
9100  PRINT : PRINT "E.G. GIVING A SALINE EMETIC TO AN
      UNCONSCIOUS PATIENT WOULD BE SCORED-3"
9110  PRINT : PRINT "50MG PETHIDINE AS ANALGESIA FOR A 30%
      BURNED PATIENT OF 70 KG WOULD BE        SCORED-2"
9120  GOSUB 90: RETURN
18085  REM ************************
18090  REM * SAVE ANSWERS TO DISC *
18095  REM ************************
19000  D$ =  CHR$ (4)
19005  PRINT D$
19010  PRINT D$;"OPEN ASSESS FILE COUNT"
19015  PRINT D$;"READ ASSESS FILE COUNT"
19020  INPUT X
19025  X = X + 1: REM * UPDATE FILE COUNT ONLY AFTER DATA ALL
       ENTERED *
19030  PRINT D$;"CLOSE ASSESS FILE COUNT"
20000  D$ =  CHR$ (4)
20010  PRINT D$;"OPEN ASSESS"X",L4"
20015  REM ** TAKE FROM 9- ONLY **
20020  FOR A = 9 TO 51
20030  PRINT D$;"WRITE ASSESS"X",R";A
20040  PRINT Q$(A)
20050  NEXT
20060  PRINT D$;"CLOSE ASSESS"X
20067  HOME
20068  VTAB 22
20070  PRINT "THE NEXT PROGRAMME IS NOW LOADING"
```

```
20080  PRINT D$;"RUN TONSILS"
29960  REM ********************************************
29970  REM * HANDLING ROUTINE FOR MORE THAN 1 QUESTION *
29980  REM ********************************************
30000  N = 1: IF I > 23 THEN 30002
30001  HOME : VTAB 10: PRINT "SCORE ALL ";J;" QUESTIONS IN THE
       NEXT": PRINT : PRINT "SECTION.": FOR XI = 1 TO 3000: NEXT
30002  HOME : PRINT G$;H$: PRINT: REM * PRINT MAIN TOPIC
       QUESTION *
30005  FOR Z = 1 TO J: PRINT : PRINT X$(Z): NEXT Z: REM * PRINT
       INDIVIDUAL QUESTIONS *
30006  HTAB 1: VTAB 20: PRINT "SCORE QUESTION ->";: FLASH :
       PRINT N + JX: NORMAL
30007  VTAB 22: HTAB 1: PRINT "TO RECALL SCORING SYSTEM, TYPE
       8": PRINT : PRINT "TO RECALL PREVIOUS TEXT, TYPE 9";
30010  VTAB V(N) + 1: HTAB (1): PRINT "  "
30012  VTAB V(N) + 1: HTAB (1)
30020  GOSUB 30100: RETURN
30030  REM ***********************
30040  REM * GET STUDENT'S REPLY *
30050  REM ***********************
30100  GET X$: IF X$ = "6" OR X$ = "7" THEN 30100
30101  IF X$ = "8" THEN  GOSUB 9038: POP : GOTO 30002
30102  IF X$ = "9" THEN  GOSUB 60000: POP : GOTO 30002
30104  IF X$ > "3" THEN 30100
30105  FLASH : PRINT X$: NORMAL
30107  HTAB 1: VTAB 20: FOR II = 1 TO 40: PRINT " ";: NEXT
30108  HTAB 1: VTAB 24: FOR II = 1 TO 38: PRINT " ";: NEXT
30110  HTAB 1: VTAB 22: PRINT "TO CHANGE ANSWER TYPE 1
       OTHERWISE TYPE 2
30115  GET Z$: IF Z$ < "1" OR Z$ > "2" THEN 30115
30120  IF Z$ = "1" THEN  VTAB 22: HTAB 1: CALL-868: GOTO 30006
30122  IF Z$ = "3" THEN  GOSUB 60000: GOTO 30002
30125  HOME :X% =  VAL (X$)
30127  REM ************************************
30128  REM * RE-ORGANISE SCORES TO NEW SYSTEM *
30129  REM ************************************
30130  IF B(N) = 5 OR B(N) = 4 THEN B(N) = 1: GOTO 30160
```

```
30140  IF B(N) = 1 OR B(N) = 2 THEN B(N) = 3: GOTO 30160
30150  IF B(N) = 3 THEN B(N) = 2
30160  IF B(N) = X% THEN C% = C% + 1: PRINT "CORRECT -":FF =
       1: GOTO 30180
30164  REM *****************
30165  REM * GIVE FEEDBACK *
30166  REM *****************
30170  W% = W% + 1: PRINT "INCORRECT -":FF = 0
30180  PRINT : PRINT A$(N);B$(N);C$(N);D$(N);E$(N): IF FF = 1
       THEN Q$(I) = "C"
30182  IF FF = 0 THEN Q$(I) = "W"
30184  I = I + 1
30186  GOSUB 90
30187  IF N = J THEN  RETURN: REM * LAST QUESTION ANSWERED *
30190  N = N + 1: GOTO 30002
32007  Z = 1
33000  FOR X = 1 TO J:A$(X) = "":B$(X) = "":C$(X) = "":D$(X) =
       "":E$(X) = "":F$(X) = "":G$(X) = "": NEXT : RETURN
39000  REM ****************************************
29970  REM * HANDLING ROUTINE FOR ONLY 1 QUESTION *
39020  REM ****************************************
40000  HOME : PRINT : PRINT G$;H$: VAB 10: HTAB 1: PRINT
       "ONLY ONE REPLY IS REQUIRED HERE"
40005  VTAB 12: HTAB 1: PRINT "TYPE IN THE NUMBER OF THE
       CORRECT ANSWER "
40010  VTAB 22: HTAB 1: PRINT "TO RECALL SCORING SYSTEM, TYPE
       8": PRINT : PRINT "TO RECALL PREVIOUS TEXT, TYPE 9";
40020  RETURN
50000  FF = 0: GOSUB 50100: GOTO 50010
50001  GOSUB 40000: VTAB 5: HTAB 1
50010  X% = 0: PRINT X$;: HTAB 40: GOSUB 52000: GOTO 52020
50020  IF X% = 1 THEN  GOTO 50000
50030  HOME : PRINT : PRINT : PRINT
50031  REM ************************************
50032  REM * RE-ORGANISE SCORES TO NEW SYSTEM *
50033  REM ************************************
50034  IF B = 5 OR B = 4 THEN B = 1: GOTO 50037
50035  IF B = 3 THEN B = 2: GOTO 50037
```

```
50036  IF B = 1 OR B = 2 THEN B = 3
50038  IF B = X% THEN 50045
50040  GOTO 50050
50042  REM *****************
50043  REM * GIVE FEEDBACK *
50042  REM *****************
50045  PRINT "CORRECT -":FF = 1: GOTO 50060
50050  W% = W% + 1: PRINT "INCORRECT -":FF = 0
50060  PRINT : PRINT A$;B$;C$;D$;E$: IF FF = 1 THEN Q$(I) =
       "C"
50062  IF FF = 0 THEN Q$(I) = "W"
50064  I = I + 1
50070  PRINT : PRINT "WHEN READY  TYPE 1"
50080  GET Z$: IF Z$ < > "1" THEN 50080
50090  RETURN
50094  REM **********************************************
50095  REM * PRINT MAIN TOPIC QUESTION FOR SINGLE REPLY *
50096  REM **********************************************
50100  HOME : PRINT : PRINT G$;H$
50200  PRINT : PRINT "1=DEFINITE  YES
50210  PRINT "2=PROBABLE YES
50220  PRINT "3=COULD DO/CONSIDER
50240  PRINT : PRINT : RETURN
50242  PRINT "4=PROBABLE NO
50244  PRINT "5=DEFINITE NO
50252  PRINT : RETURN
51170  REM *******************************************
51180  REM * GET STUDENT'S REPLY FOR SINGLE QUESTION *
51190  REM *******************************************
52000  GET Z$: IF Z$ > "3" AND Z$ < "8" THEN 52000
52001  IF Z$ < "1" THEN 52000
52002  IF Z$ > "9" THEN 52000
52003  IF Z$ = "9" THEN  GOSUB 60000: POP : GOTO 50001
52004  IF Z$ = "8" THEN  GOSUB 9038: POP : GOTO 50001
52005  X% =  VAL (Z$)
52007  VTAB 22: HTAB 1: FOR II = 1 TO 39: PRINT " ";: NEXT
52008  VTAB 24: HTAB 1: FOR II = 1 TO 39: PRINT " ";: NEXT
```

```
52009  REM *************************
52010  REM * ALLOW TO CHANGE SCORE *
52012  REM *************************
52015  VTAB 10: HTAB 1: FOR II = 1 TO 40: PRINT " ";: NEXT :
       VTAB 10: HTAB 1: PRINT "THE ANSWER YOU SELECTED IS ";:
       FLASH : PRINT X%: NORMAL : RETURN
52020  PRINT : PRINT "IF YOU WISH TO CHANGE YOUR ANSWER TYPE
       1": PRINT "-------IF NOT TYPE 2        ";
52030  GET Z$: IF Z$ < "1" OR Z$ > "3" THEN 52030
52040  IF Z$ = "1" THEN 52060
52045  IF Z$ = "3" THEN  HOME : GOSUB 60000: GOTO 52060
52050  GOTO 50030
52055  HOME
52060  IF F = 1 THEN  GOTO 50001
52070  GOTO 50000
59970  REM ***********************
59980  REM * PRINT PREVIOUS TEXT *
59990  REM ***********************
60000  HOME
60002  FOR M = 1 TO 9: PRINT T$(M): NEXT :M = M - 1: GOSUB 90:
       IF M = T THEN  RETURN
60005  FOR M = 11 TO 15: PRINT T$(M): NEXT :M = M - 1: GOSUB
       90: IF M = T THEN  RETURN
60020  RETURN
```

```
4    REM **************
5    REM * CASA-GRAPH *
6    REM **************
10   D$ =  CHR$ (4):N = 0: DIM Q$(118),CL(100),SC(100)
20   POKE 216,0: PRINT D$;"NOMONI,C,O"
30   GOTO 110
45   REM *****************************
40   REM * CASE CONVERTER SUBROUTINE *
48   REM *****************************
50   FOR J = 1 TO  LEN (A$):L = 32:L$ =  MID$ (A$,J,1): IF L$ =
     "@" OR L$ = "&" THEN J = J + 1:L = 0 - 64 * (L$ = "&")
60   I =  ASC ( MID$ (A$,J)): IF I <  ASC ("A") OR I >  ASC
     ("Z") THEN L = 0
70   PRINT  CHR$ (I + L);: NEXT : PRINT : RETURN
95   SS = 160 / HY: REM * SCALING FOR Y-AXIS *
110   TEXT : CALL  - 936
130   PRINT D$;"BLOAD CHARGEN A$6000 L$100"
140   PRINT D$;"BLOAD CHARTAB A$6800 L$400"
150   Z = 49: REM * TOTAL SCORE FOR OBST PROG *
154   REM ***********************************
155   REM * OBTAIN NAME AND NUMBER OF FILES *
156   REM ***********************************
160   HOME : PRINT "TYPE IN FILE NAME E.G. 'YORKASSESS'": PRINT :
      PRINT " THEN <RETURN> ";: INPUT N$
165   PRINT : PRINT : PRINT "TYPE IN THE NUMBER OF FILES
      <RETURN> ";: INPUT NF
166   REM **************************************
167   REM * CHANGE SCALING OF AXES IF REQUIRED *
168   REM **************************************
170   HOME : PRINT : PRINT "PRESENT Y-AXIS": PRINT : PRINT
      "0-8"
172   PRINT : PRINT "TO CHANGE TYPE LOW-Y THEN HIGH-Y": PRINT
      "OTHERWISE TYPE 99 ";: INPUT LY: IF LY = 99 THEN LY = 0:
      GOTO      190
175   INPUT HY
190   IF HY = 0 THEN HY = 8
192   PRINT : PRINT : PRINT "PRESENT NO. OF DIVISIONS = 8":
      PRINT : PRINT "TO CHANGE TYPE NO. OTHERWISE TYPE 99 ";:
```

```
     INPUT NY: IF NY = 99 THEN NY = 8
195  SS = 160 / HY: REM * SCALING FACTOR FOR Y-AXIS *
200  HOME : PRINT : PRINT "PRESENT X-AXIS": PRINT : PRINT
     "50-100"
202  PRINT : PRINT "TO CHANGE TYPE LOW-X THEN HIGH-X": PRINT
     "OTHERWISE TYPE 99";: INPUT LX: IF LX = 99 THEN LX = 50:HX =
     100: GOTO 205
204  INPUT HX
205  PRINT : PRINT : PRINT "PRESENT NO. OF DIVISIONS = 10":
     PRINT : PRINT "TO CHANGE TYPE NO. OTHERWISE TYPE 99 ";:
     INPUT NX: IF NX = 99 THEN NX = 10
206  NC = HX - LX
207  FC = LX: IF FC = 0 THEN FC = 1
210  FOR I = 1 TO NF
214  REM **************************************
215  REM * READ FILES AND OBTAIN TOTAL SCORES *
216  REM **************************************
220  PRINT D$;"OPEN"N$;I",L4"
221  PRINT D$;"READ"N$;I",R";50: INPUT A$: IF A$ = "Y" THEN
     280
230  PRINT D$;"READ"N$;I",R";Z
240  INPUT A$: IF A$ = "11" THEN 280
250  A =  VAL (A$)
260  N = N + 1:A1 = A1 + A:A2 = A2 + (A * A)
270  CL(A) = CL(A) + SS
280  PRINT D$;"CLOSE"N$;I
290  NEXT I
292  HOME : PRINT "ANOTHER DISC? ";: GET A$: IF A$ = "Y" THEN
     160
293  IF A$ <  > "N" THEN 292
294  PRINT
300  MEAN = A1 / N:SD =  SQR (((N * A2) - (A1 * A1)) / (N - 1)
     ^ 2):SE = SD /  SQR (N)
310  X$ = "@SCORE (PER CENT)"
320  Y$ = "@NUMBER"
330  CW = 200 / NC: REM * CELL WIDTH *
332  GOTO 360
335  PRINT : PRINT "CELL WIDTH=";CW: PRINT "TO CHANGE TYPE NO.
```

```
    OTHERWISE TYPE 99";: INPUT ZW: IF ZW = 99 THEN ZW = CW
340 REM ********************
350 REM * GRAPHICS ROUTINE *
355 REM ********************
360 HGR2 : PRINT D$;"PR#0": PRINT D$;"IN#0": POKE 54,0: POKE
    55,96: REM * INITIALISE HIRES CHARACTER MODE *
370 HCOLOR= 7
380 HPLOT 79,0 TO 79,160 TO 279,160
390 FOR V = 0 TO NY:Y = 160 - 160 * V / NY: HPLOT 74,Y TO
    78,Y: NEXT
400 FOR V = 0 TO NX:X = 79 + V * 200 / NX: HPLOT X,161 TO
    X,165: NEXT
410 FOR V = 0 TO NY STEP 2:Y = 160 - 160 * V / NY:A$ =  STR$
    (LY + V * (HY - LY) / NY): VTAB (Y / 8 + 1): HTAB (11 -  LEN
    (A$)): GOSUB 50: HPLOT 74,Y TO 79,Y: NEXT
420 FOR V = 0 TO NX STEP 2: VTAB 22:X = 79 + V * 200 / NX:A$
    =  STR$ (LX + V * (HX - LX) / NX):HT = 13 + (X - 80) * 28 /
    200 -  LEN (A$) / 2: IF HT +  LEN (A$) > 40 THEN HT = 41 -
    LEN (A$)
430 HTAB HT: GOSUB 50: NEXT
440 VTAB 3: HTAB 1:A$ = ""
450 FOR V = 1 TO  LEN (Y$): IF  MID$ (Y$,V,1) = " " THEN
    GOSUB 50:A$ = "": GOTO 470
460 A$ = A$ +  MID$ (Y$,V,1)
470 NEXT : GOSUB 50
480 VTAB 24: HTAB 41 -  LEN (X$):A$ = X$: GOSUB 50
490 HCOLOR= 1
494 REM ****************
495 REM * PLOT RESULTS *
496 REM ****************
500 FOR I = FC TO FC + NC - 1
510 FOR J = 1 TO CW:X = 79 + (I - FC) * CW + J: HPLOT X,159
    TO X,159 - CL(I): NEXT J: NEXT I
550 VTAB 1: HTAB 30:A$ = "MEAN=" +  STR$ ( INT (10 * MEAN +
    .5) / 10): GOSUB 50
560 HTAB 30:A$ = "S.D.=" +  STR$ ( INT (10 * SD + .5) / 10):
    GOSUB 50
570 HTAB 30:A$ = "S.E.=" +  STR$ ( INT (10 * SE + .5) / 10):
```

```
     GOSUB 50
575  HTAB 30:A$ = "N=" +  STR$ (N): GOSUB 50
580  GET A$: PRINT : GOSUB 30000: PRINT D$;"PR#0": STOP
29000  REM ****************************
29010  REM * PRINT GRAPH ON SILENTYPE *
29020  REM ****************************
30000  VTAB 2: PRINT : PRINT D$;"PR#1": PRINT : PRINT " ":
       POKE  - 12528,7: POKE  - 12526,83: POKE  - 12525,64:
       POKE  - 12529,255: POKE  - 12524,0: PRINT  CHR$ (17):
       PRINT D$;"PR#0":A$ = "": RETURN
```

```
6   REM *****************
5   REM * CASA-ANALYSIS *
8   REM *****************
10  DIM V(33),T(33,4),Q$(200,8)
15  HOME : INPUT "PRINTER 0/1 ";PR
20  HOME : PRINT "PUT DISC IN DRIVE"
80  PRINT : INPUT "TYPE FILE NAME ";N$
90  PRINT : INPUT "TYPE 1ST FILE NO. ";P1
100  PRINT : INPUT "TYPE LAST FILE NO. ";P2
110  GOSUB 1700: REM ** SET UP P LOOK-UP TABLE **
120  D$ =  CHR$ (4):N = 0
130  PRINT D$;"NOMONI,C,O"
140  PRINT D$
150  PRINT D$
160  FOR I = P1 TO P2
170  HOME : PRINT I
180  F = I - ZZ
190  PRINT D$;"OPEN";N$;F",L4
195  REM * K=UNACCEPTABLE FOR ANALYSIS *
200  PRINT D$;"READ";N$;F",R";0
202  INPUT A$: IF A$ = "N" THEN K = K + 1: GOTO 270: REM * NOT
     ANAESTHETIST *
206  PRINT D$;"READ";N$;F",R";50
208  INPUT A$: IF A$ = "Y" THEN K = K + 1: GOTO 270: REM *
     USED PROG BEFORE *
209  FOR J = 1 TO 7
210  PRINT D$;"READ";N$;F",R";J
220  INPUT Q$(I - K,J)
230  NEXT J
240  J = 49: REM * OBST TOTAL SCORE *
242  PRINT D$;"READ";N$;F",R";J
250  INPUT Q$(I - K,8)
270  PRINT D$;"CLOSE"
280  NEXT I
290  HOME : PRINT "ANALYSE ANOTHER DISC Y/N ";
300  GET A$: PRINT A$: IF A$ = "N" THEN P2 = P2 - K: GOTO 470
310  IF A$ <  > "Y" THEN 300
320  PRINT : INPUT "TYPE FILE NAME ";N$
```

```
330  PRINT : INPUT "TYPE 1ST FILE NO. ";P3
340  PRINT : INPUT "TYPE 2ND FILE NO. ";P4
350  NF = P4 - P3 + 1
360  ZZ = P2:P1 = P2 + 1:P2 = P1 + NF - 1
370  GOTO 140
470  Z = 8: REM * OBST SCORE= 8TH ELEMENT IN ARRAY *
475  HOME : PRINT "FOR MEAN ANALYSIS TYPE 1"
480  PRINT : PRINT "T TEST TYPE 2"
490  GET A$: PRINT A$
500  PRINT D$;"PR#";PR
510  A =  VAL (A$): ON A GOSUB 700,1190
515  GOTO 470
699  REM ******************************
700  REM * MEAN/SD/SEM/SUMX^2 ANALYSIS *
701  REM ******************************
705  GOTO 800
1152  PRINT "QUES ";QUES;" SUB-GROUP ";K$: PRINT "MEAN =
      ";MEAN: PRINT "SD = ";SD: PRINT "SE = ";SE: PRINT "N = ";N
707  REM ****************************************************
708  REM * SELECT ALTERNATIVE RECORD NOS FOR DIFFERENT PROGRAMS *
709  REM ****************************************************
710  PRINT : PRINT : PRINT : IF Z = 8 THEN  PRINT "OBSTETRIC
      ANALYSIS"
720  PRINT : PRINT : PRINT : IF Z = 102 THEN  PRINT "BURNS
     ANALYSIS (ABSOLUTELY CORRECT": PRINT : PRINT : PRINT
730  PRINT : PRINT : PRINT : IF Z = 103 THEN  PRINT "BURNS
     ANALYSIS (NEARLY+ABSOLUTELY)": PRINT : PRINT : PRINT
740  PRINT : PRINT : PRINT : IF Z = 117 THEN  PRINT "SHOCK
     ANALYSIS (ABSOLUTELY CORRECT": PRINT : PRINT : PRINT
750  PRINT : PRINT : PRINT : IF Z = 118 THEN  PRINT "SHOCK
     ANALYSIS (NEARLY+ABSOLUTELY)": PRINT : PRINT : PRINT
760  QUES = 1:NN = 9770  GOSUB 780: GOTO 790
780  FOR I = 1 TO 129: PRINT "-";: NEXT : PRINT : PRINT : RETURN
784  REM *****************
785  REM * PRINT HEADINGS *
786  REM *****************
790  PRINT
800  PRINT "UNDERGRADUATE EDUCATION"
```

```
810  GOSUB 1120
820  QUES = 2:NN = 9
830  PRINT
840  GOSUB 780
850  PRINT "POST-GRADUATE EDUCATION"
860  GOSUB 1120
870  QUES = 3:NN = 9
880  PRINT
890  GOSUB 780
900  PRINT "GEOGRAPHICAL AREA EMPLOYED"
910  GOSUB 1120
920  QUES = 4:NN = 8
930  PRINT
940  GOSUB 780
950  PRINT "GRADE OF EMPLOYMENT"
960  GOSUB 1120
970  QUES = 5:NN = 4
980  GOSUB 780
990  PRINT "EXPERT/NON-EXPERT"
1000  GOSUB 1120
1010  QUES = 6:NN = 9
1020  PRINT
1030  GOSUB 780
1040  PRINT "YEARS PRACTISING ANAESTHESIA"
1050  GOSUB 1120
1060  QUES = 7:NN = 9
1070  PRINT
1080  GOSUB 780
1090  PRINT "YEARS SINCE HIGHER QUALIFICATION OBTAINED"
1100  GOSUB 1120
1110  RETURN: REM * TO MENU AT LINE 470 *
1119  REM ***********************************
1120  REM * MEAN/SD/SEM/SUMX^2 CALCULATIONS *
1121  REM ***********************************
1122  FOR J = 1 TO NN:K$ =  STR$ (J): REM * NN = NO OF SUBGROUPS
      IN EACH SECTION *
1124  N = 0:A1 = 0:A2 = 0:MEAN = 0:SD = 0:SE = 0
1126  FOR I = 1 TO P2
```

```
1128  IF Q$(I,QUES) <  > K$ THEN 570: REM * NOT IN SUBGROUP *
1130  A =  VAL (Q$(I,Z)): PRINT A,
1132  N = N + 1:A1 = A1 + A:A2 = A2 + (A * A)
1134  NEXT I
1136  IF N < 2 THEN 1146
1138  X1(QUES, VAL (K$)) = A1:X2(QUES, VAL (K$)) = A2
1140  MEAN = A1 / N:SD =  SQR (((N * A2) - (A1 * A1)) / (N - 1)
      ^ 2):SE = SD /  SQR (N)
1142  MX(QUES, VAL (K$)) = MEAN
1144  NX(QUES, VAL (K$)) = N
1146  IF N = 0 THEN 1180
1148  PRINT
1150  PRINT
1154  PRINT "SUM OF X = ";A1,"SUM OF X^2 = ";A2
1156  PRINT : PRINT
1180  NEXT J:RETURN
1189  REM ******************
1190  REM * T-TEST ANALYSIS *
1191  REM ******************
1200  FOR QUES = 1 TO 7: REM * 7 MAIN GROUPS OF PARTICIPANTS *
1210  FOR K = 1 TO 9:K$ =  STR$ (K): REM * MAX OF 9 SUBGROUPS
      IN EACH MAIN GROUP *
1220  FOR I = 1 TO P2: REM * TOTAL NO OF FILES *
1230  IF Q$(I,QUES) <  > K$ THEN 1260: REM * NOT IN SUBGROUP *
1240  A =  VAL (Q$(I,Z))
1250  N = N + 1:A1 = A1 + A:A2 = A2 + (A * A)
1260  NEXT I
1270  IF N < 2 THEN 1330
1280  X1(QUES, VAL (K$)) = A1:X2(QUES, VAL (K$)) = A2
1290  MEAN = A1 / N:SD =  SQR (((N * A2) - (A1 * A1)) / (N - 1)
      ^ 2):SE = SD /  SQR (N)
1300  MX(QUES, VAL (K$)) = MEAN
1310  NX(QUES, VAL (K$)) = N
1320  N = 0:A1 = 0:A2 = 0
1330  NEXT K
1340  NEXT QUES
1350  PRINT : PRINT
```

```
1354  REM *****************
1355  REM * PRINT RESULTS *
1356  REM *****************
1360  PRINT D$;"PR#";PR
1370  FOR I = 1 TO 7: FOR J = 1 TO 9
1380  IF NX(I,J) < 2 THEN 1460
1390  NX = NX(I,J)
1400  MX = MX(I,J):X1 = X1(I,J):X2 = X2(I,J)
1410  FOR QUES = 1 TO 7: FOR K = 1 TO 9
1420  IF NX(QUES,K) < 2 THEN 1450
1430  MY = MX(QUES,K):NY = NX(QUES,K):Y1 = X1(QUES,K):Y2 =
      X2(QUES,K)
1440  GOSUB 1490
1450  NEXT K: NEXT QUES
1460  NEXT J,I
1470  FOR I = 1 TO 25: PRINT : NEXT
1480  RETURN
1485  REM ***********************
1490  REM * T-TEST CALCULATIONS *
1495  REM ***********************
1500  ZX = X2 - X1 ^ 2 / NX
1510  ZY = Y2 - Y1 ^ 2 / NY
1512  IF ZX + ZY <  = 0 THEN 1610
1515  IF MX = MY THEN 1610
1520  T = (MX - MY) /  SQR ((ZX + ZY) / (NX - 1 + NY - 1) * (1
      / NX + 1 / NY))
1530  T =  ABS (T)
1540  IF T < 1.96 THEN 1610
1545  REM *****************
1550  REM * P CALCULATION *
1555  REM *****************
1560  DF = NX - 1 + NY - 1
1570  GOSUB 1620
1580  IF P$ < ".001" THEN 1610
1590  PRINT "QUES ";I;" SUB-GROUP ";J,"QUES ";QUES;" SUB-GROUP
      ";K,"P = ";P$
1600  PRINT : PRINT
1610  RETURN
```

```
1614  REM **************************
1615  REM * CHECK FOR SIGNIFICANCE *
1616  REM **************************
1620  FOR E = 1 TO 33
1630  IF DF <  = V(E) THEN 1650: REM * SIGNIFICANT RESULT *
1640  NEXT : RETURN
1650  IF T > T(E,4) THEN P$ = ".001": RETURN
1660  IF T > T(E,3) THEN P$ = ".01": RETURN
1670  IF T > T(E,2) THEN P$ = ".02": RETURN
1680  IF T > T(E,1) THEN P$ = ".05": RETURN
1690  P$ = "N.S.": RETURN
1695  REM *******************
1700  REM * P LOOK-UP TABLE *
1705  REM *******************
1710  FOR I = 1 TO 30:V(I) = I: NEXT
1720  V(31) = 40:V(32) = 60:V(33) = 120
1730  FOR I = 1 TO 33: FOR J = 1 TO 4: READ T(I,J): NEXT J,I:
      RETURN
1740  DATA 12.706,31.821,63.657,633.619,4.303,6.965,9.925,
      31.598,3.182,4.541,5.841,12.924
1750  DATA   2.776,3.747,4.604,8.61,2.571,3.365,4.032,6.869,
      2.447,3.143,3.707,5.959
1760  DATA  2.365,2.998,3.499,5.408,2.306,2.896,3.355,5.041,
      2.262,2.821,3.25,4.781,2.228,2.764,3.169,4.587,2.201,
      2.718,3.106,4.
1770  DATA  2.197,2.681,3.055,4.318,2.16,2.65,3.012,4.221,
      2.145,2.624,2.977,4.14,2.131,2.602,2.947,4.073,2.12,2.583,
      2.921,4.015
1780  DATA  2.101,2.552,2.878,3.922,2.093,2.539,2.861,3.883,
      2.086,2.528,2.845,3.85,2.08,2.518,2.831,3.819,2.074,2.508,
      2.819,3.7
1790  DATA  2.064,2.492,2.797,3.745,2.06,2.485,2.787,3.725,
      2.056,2.479,2.779,3.707,2.052,2.473,2.771,3.69,2.048,2.467,
      2.763,3.6
1800  DATA  2.042,2.457,2.75,3.646,2.021,2.423,2.704,3.551,
      2,2.39,2.66,3.46,1.98,2.358,2.617,3.373,1.96,2.326,2.576,
      3.291
```

```
1810  DATA  1.68,2.01,2.4,1.66,1.98,2.36,1.65,1.96,2.33,1.64,
      1.96,2.33
1820  RETURN
```

```
1   REM ******************
2   REM * HELLO-CALWRITER *
3   REM ******************
4   POKE 50944,5
5   D$ =  CHR$ (4)
6   TEXT : GOTO 10
7   HTAB (40 -  LEN (Z$)) / 2: PRINT Z$: RETURN
10  HOME :Z$ = "GLASGOW ROYAL INFIRMARY": GOSUB 7: PRINT :Z$ =
    "CAL/CASA DEVELOPMENT PROGRAMS": GOSUB 7
15  PRINT : PRINT
20  PRINT : PRINT : PRINT "1. WRITE / EDIT CAL PROGRAMME"
30  PRINT : PRINT "2. RUN TEACHING PROGRAMME
34  PRINT : PRINT "3. INSTRUCTIONS": PRINT : PRINT : PRINT
35  GET A$
40  IF A$ = "1" THEN  PRINT A$: PRINT D$"RUN CALWRITE"
50  IF A$ = "2" THEN  PRINT A$: PRINT D$;"RUN CALREAD"
60  IF A$ = "3" THEN  PRINT A$: PRINT D$;"RUN CALINSTRUCT"
70  GOTO 35
```

```
2       REM ***************
3       REM * CALINSTRUCT *
4       REM ***************
5       D$ =  CHR$ (4)
6       TEXT : GOTO 10
7       HTAB (40 -  LEN (Z$)) / 2: PRINT Z$: RETURN
10      GOSUB 1000
20      HOME : VTAB 20:Z$ = "LOADING START PROGRAM": GOSUB 7:
        PRINT : PRINT D$;"RUN HELLO"
60      IF A$ = "3" THEN  GOSUB 1000: HOME : GOTO 10
770     VTAB 23: GOSUB 1190
1000    TEXT : GOTO 1020
1010    HTAB (40 -  LEN (Z$)) / 2: PRINT Z$: RETURN
1020    D$ =  CHR$ (4)
1030    HOME : PRINT "THIS SUITE OF PROGRAMMES ALLOWS THE
        GENERATION AND RUNNING OF COMPUTER      ASSISTED LEARNING
        PACKAGES."
1050    PRINT
1060    PRINT : PRINT "THERE ARE THREE MAIN PROGRAMMES IN THIS
        SUITE:-
1070    PRINT : PRINT "CALINSTRUCT.....YOU ARE USING THIS
                PROGRAMME AT PRESENT.
1080    PRINT : PRINT "CALWRITE........THIS ALLOWS YOU TO
                PREPARE FILES WHICH WILL                BE READ
                BY 'CALREAD'.
1090    PRINT : PRINT "CALREAD.........ALLOWS THESE FILES TO BE
                PRESENTED FOR TEACHING                  OR
                SELF-ASSESSMENT.
1100    PRINT : PRINT : PRINT : GOSUB 1190: HOME
1101    PRINT
1102    PRINT "THERE ARE 3 OTHER PROGRAMMES OF LESSER
             IMPORTANCE WHICH ASSIST IN EDITING.": PRINT
1103    PRINT
1104    PRINT : PRINT "CALPRINT........ALLOWS THE FILES TO BE
                PRINTED FOR EXAMINATION.
1106    PRINT : PRINT "CALFILE-EDIT....ALLOWS THE FILE NAMES TO
                BE ALTERED."
1107    PRINT
```

```
1108  PRINT : PRINT "CALFILE-SORT....LISTS THE FILE NAMES
               ALPHABETICALLY."
1109  PRINT : PRINT : PRINT : GOSUB 1190: HOME
1110  PRINT : PRINT "THE AUTHOR SHOULD WRITE OUT THE
      TEACHINGPROGRAMME IN THE FORM OF PAGES:-
1120  PRINT : PRINT "ONE PAGE CONSISTS OF A FRAME OF TEXT
      SUCH AS A SHORT CASE HISTORY OR THE     RESULTS OF LABORATORY
      INVESTIGATIONS.
1130  PRINT : PRINT "THE TEXT FRAME IS THEN FOLLOWED BY A
      MAIN TOPIC QUESTION TO WHICH THE LATER  INDIVIDUAL QUESTIONS
      WILL REFER.
1140  PRINT : PRINT "THE INDIVIDUAL QUESTIONS WHICH REFER TO
      THE MAIN TOPIC QUESTION ARE THEN        FOLLOWED BY THE
      CORRECT SCORES FOR EACH QUESTION AND BY SUITABLE FEEDBACK.
1150  PRINT : PRINT : GOSUB 1190:
1160  HOME : PRINT "THERE ARE 5 SEPARATE FRAMES IN EACH PAGE":
      PRINT : PRINT : PRINT "1. TEXT FRAME": PRINT : PRINT "2. TOPIC
      QUESTION FRAME"
1170  PRINT : PRINT "3. INDIVIDUAL QUESTION FRAME": PRINT :
      PRINT "4. CORRECT SCORES FRAME": PRINT : PRINT "5. FEEDBACK
      FRAME"
1180  PRINT : PRINT : PRINT : GOSUB 1190: GOTO 1210
1190  HTAB 9: PRINT "WHEN READY TYPE 1 -->";: GET A$: IF A$ <
      > "1" THEN 1190
1200  RETURN
1210  HOME : PRINT "THE FORM OF LAYOUT IS THEREFORE:-"
1220  PRINT :Z$ = "PAGE 1": GOSUB 1010: PRINT
1230  PRINT "TEXT": PRINT "XXX XXXX XX XXXXXX XXX.  XXX XX
      XXXXX XXX XX.  XXXXX  XXX."
1240  PRINT : PRINT "MAIN TOPIC QUESTION": PRINT "XX XXXXX
      XX?"
1250  PRINT : PRINT : PRINT "INDIVIDUAL QUESTIONS AND SCORES":
      PRINT "1. XXX XXXXX ---------------5"
1260  PRINT "2. XXXX XXX XXX ------------1"
1270  PRINT : PRINT : PRINT "FEEDBACK": PRINT : PRINT "XXXXXXX
      XX XX XX XX"
1280  PRINT : PRINT : GOSUB 1190
1290  HOME : PRINT "A MAXIMUM OF 20 PAGES CAN BE ENTERED
```

```
     FORANY ONE PROGRAMME, BUT EACH PAGE DOES   NOT HAVE TO DISPLAY
     A TEXT FRAME OR     QUESTION FRAME."
1300 PRINT
1310 PRINT : PRINT "THEREFORE, IF TWO TEXT FRAMES ARE
     NEEDEDTO FOLLOW EACH OTHER, THIS CAN BE       ARRANGED."
1320 PRINT
1330 PRINT : PRINT "EACH SECTION OF AN INDIVIDUAL PAGE WILL
     NOW BE DESCRIBED AND THE REQUIRED       INSTRUCTIONS
     EXPLAINED.": PRINT : PRINT : PRINT : PRINT
1340 GOSUB 1190
1350 HOME :Z$ = "TEXT FRAME": GOSUB 1010: PRINT : PRINT :
     PRINT "A MAXIMUM OF  8 TEXT STRINGS CAN BE     ENTERED ON ANY
     ONE PAGE.  A LINE FEED ISAUTOMATICALLY INSERTED BETWEEN EACH
     TEXT STRING.
1360 PRINT : PRINT "NO MORE THAN 5 LINES OF TEXT SHOULD BE
     TYPED INTO A SINGLE STRING AND AT LEAST 3 LINES SHOULD BE LEFT
     AT THE END OF THETEXT FRAME.
1370 PRINT : PRINT "THE BASIC INTERPRETER USED TO WRITE THE
     PROGRAMME DOES NOT ALLOW THE USE OF     COMMAS (,) OR COLONS
     (:) IN 'INPUT'     COMMANDS AND THESE MUST NOT BE USED FOR
     PUNCTUATION IN ANY OF THE FRAMES."
1380 PRINT : PRINT "THE WORD 'END' SHOULD BE TYPED TO
     INDICATE THAT THE ENTRY OF THE TEXT     FRAME HAS BEEN
     COMPLETED."
1390 PRINT : GOSUB 1190
1400 HOME :Z$ = "TEXT FRAME - 2": GOSUB 1010: PRINT : PRINT
     PRINT "IF NO TEXT ENTRY IS REQUIRED THEN THE   WORD 'NONE'
     SHOULD BE TYPED.
1410 PRINT : PRINT "AFTER EACH TEXT STRING AND EACH COMMAND
     WORD THE <RETURN> KEY MUST BE PRESSED TOINDICATE TO THE
     COMPUTER THAT YOU HAVE  COMPLETED YOUR ENTRY.
1420 PRINT : PRINT "AFTER ENTRY OF ALL THE TEXT STRINGS YOU
     WILL BE SHOWN THEM AS THEY WILL APPEAR  IN THE CAL PROGRAMME.
     ANY CORRECTIONS  CAN BE MADE BY TYPING THE LETTER 'C'
     FORC(ORRECT."
1430 PRINT : PRINT "IF THE TEXT FRAME IS CORRECT THEN TYPE
     'F' FOR F(ORWARD AND THE TOPIC QUESTION FRAME WILL BE
     DISPLAYED."
```

```
1440  PRINT : PRINT : PRINT : GOSUB 1190
1450  HOME :Z$ = "TOPIC QUESTION FRAME": GOSUB 1010: PRINT :
      PRINT : PRINT "THE TOPIC QUESTION IS TYPED IN AND CAN  BE
      CORRECTED IN THE SAME WAY AS THE TEXTSTRINGS.
1460  PRINT : PRINT "TYPING 'B' FOR B(ACK DISPLAYS THE TEXT
      FRAME AGAIN.  THIS CAN BE RE-EDITED IF  REQUIRED."
1470  PRINT : PRINT "TYPING 'F' FOR F(ORWARD DISPLAYS THE
      INDIVIDUAL QUESTION FRAME.
1480  PRINT : PRINT : PRINT : GOSUB 1190
1490  HOME :Z$ = "INDIVIDUAL QUESTION FRAME": GOSUB 1010:
      PRINT : PRINT : PRINT "A MAXIMUM OF 6 INDIVIDUAL QUESTIONS CAN
      BE ENTERED IN EACH FRAME.  IF LESS THAN 6 ARE REQUIRED THEN
      THE WORD 'END '";
1500  PRINT "     SHOULD AGAIN BE USED TO INDICATE THAT NOMORE
      QUESTIONS ARE REQUIRED."
1510  PRINT : PRINT "INDIVIDUAL QUESTIONS CAN BE CORRECTED
      BY TYPING 'C' AND THE PREVIOUS FRAMES   VIEWED BY TYPING 'B'.
      TYPING 'F' WILL  LEAD ON TO THE 'CORRECT SCORES' FRAME."
1520  PRINT : PRINT : PRINT : GOSUB 1190
1530  HOME :Z$ = "CORRECT SCORES FRAME": GOSUB 1010: PRINT :
      PRINT : PRINT "THE CORRECT SCORE FOR EACH INDIVIDUAL
      QUESTION SHOULD BE TYPED IN AS REQUESTEDAND ANY ALTERATION
      REQUIRES ALL THE     SCORES TO BE RE-ENTERED.
1540  PRINT : PRINT "BACKWARD OR FORWARD MOVEMENT THROUGH
      THEFRAMES IS CONTROLLED WITH THE 'B' AND   'F' KEYS AS BEFORE.
1550  PRINT : PRINT "THE 'F' KEY WILL ALLOW THE FEEDBACK FOR
      EACH INDIVIDUAL QUESTION TO BE ENTERED.
1560  PRINT : PRINT : PRINT : GOSUB 1190
1570  HOME :Z$ = "FEEDBACK FRAME": GOSUB 1010: PRINT : PRINT :
      PRINT "A MAXIMUM OF 5 FEEDBACK STRINGS CAN BE  TYPED IN FOR
      EACH INDIVIDUAL QUESTION.  EACH FEEDBACK STRING  SHOULD BE NO
      MORE"
1580  PRINT "THAN 5 LINES IN LENGTH AND SHOULD NOT   CONTAIN
      ANY COMMAS OR SEMI-COLONS.
1590  PRINT : PRINT "THE FEEDBACK FOR EACH INDIVIDUAL ANSWER
      SHOULD BE CORRECTED BEFORE MOVING TO THEFEEDBACK FOR THE NEXT
      QUESTION.
1600  PRINT : PRINT "HOWEVER, AS WITH THE OTHER FRAMES, THE
```

```
     'B' AND 'F' KEYS ALLOW MOVEMENT THROUGH THE FRAMES AND ANY
     FRAME CAN BE        CORRECTED BY MOVING TO IT AND TYPING
     'C'FOR 'C(ORRECT'.
1610 PRINT : PRINT : GOSUB 1190
1620 HOME : PRINT "SUMMARIES OF THESE INSTRUCTIONS ARE
     DISPLAYED EITHER JUST BEFORE THE FRAME  TO BE COMPLETED OR AT
     THE LAST LINE OF  THE FRAME.
1625 PRINT
1630 PRINT : PRINT "THE FRAMES OF EACH PAGE CAN THEREFORE
     BEEASILY CORRECTED FROM WITHIN THAT PAGE.
1635 PRINT
1640 PRINT : PRINT "TO CORRECT OUTWITH THE PAGE BEING
     ENTERED, THE 'E' KEY SHOULD BE PRESSED  WHEN IT APPEARS WITH
     THE OTHER PROMPT   LETTERS.
1645 PRINT
1650 PRINT : PRINT "THE PAGE NUMBER TO BE CORRECTED IS THEN
     ENTERED.  THE FRAMES CAN BE DISPLAYED ASBEFORE USING THE 'B'
     AND 'F' KEYS AND   CORECTIONS MADE USING THE 'C' KEY."
1655 PRINT : PRINT : PRINT : GOSUB 1190: HOME
1660 PRINT : PRINT "IF IT IS NECCESSARY TO EDIT A PREVIOUS
     PROGRAMME, THEN THE EDITOR SHOULD BE    SELECTED FROM THE
     FIRST MENU OF THE": PRINT "PROGRAMME 'CALWRITE'.
1665 PRINT
1670 PRINT : PRINT "THIS ALLOWS FILES WHICH HAVE BEEN
     DEVELOPED PREVIOUSLY TO BE LOADED INTO  MEMORY.
1675 PRINT
1680 PRINT : PRINT "INDIVIDUAL PAGES CAN THEN BE SELECTED
     AND FRAMES WITHIN THESE PAGES EXAMINED  AND CORRECTED IF
     NECCESSARY.
1685 PRINT
1690 PRINT : PRINT "EXTRA PAGES CAN BE ADDED TO THE END OF
     THE FILE OR INSERTED AT ANY PLACE."
1770 VTAB 24: GOSUB 1190
1780 RETURN
```

```
2 REM ************
3 REM * CALWRITE *
4 REM ************
5 TEXT : SPEED= 255: GOTO 10
6 POKE - 16368,0
7 CALL - 868: HTAB (40 - LEN (Z$)) / 2: PRINT Z$;: RETURN
10 HOME :Z$ = "GLASGOW ROYAL INFIRMARY": GOSUB 70: PRINT ::Z$
   = "CAL/CASA DEVELOPMENT PROGRAMS": GOSUB 70: FOR I = 1 TO
   2000: NEXT
11 DIM T$(30,10),X$(30,6),B%(30,6),G$(30)
12 D$ = CHR$ (4)
13 PRINT D$;"NOMONI,C,O"
14 DIM NT%(30),NX%(30),TE$(30,6,5)
15 REM **************
16 REM * FIRST MENU *
17 REM **************
18 VTAB 10
19 PRINT "1. ENTER NEW PROGRAMME": PRINT : PRINT : PRINT "2.
   EDIT PREVIOUS PROGRAMME"
20 PRINT : PRINT : PRINT "3. EDIT LIST OF PROGRAMME NAMES"
21 PRINT : PRINT : PRINT "4. RETURN TO START-UP PROGRAMME
22 VTAB 23: HTAB 1: PRINT "TYPE NO. REQUIRED ";: GET A$: IF
   A$ = "1" THEN SELECT$ = "NEW": GOTO 28
23 IF A$ = "2" THEN SE$ = "OLD": GOTO 28
24 IF A$ = "3" THEN PRINT : PRINT D$;"RUN CALFILE-EDIT"
25 IF A$ = "4" THEN PRINT : PRINT D$;"RUN HELLO"
26 GOTO 22
28 VTAB 24: CALL - 868: ONERR GOTO 38
29 PRINT D$: REM * GET DATA FILE NAMES *
30 X$ = "CALFILELEN": PRINT D$: PRINT D$"OPEN";X$: PRINT
   D$;"READ";X$: INPUT FI: PRINT D$"CLOSE"
31 X$ = "CALFILE": PRINT D$;"OPEN";X$: PRINT D$;"READ";X$: FOR
   I = 1 TO FI: INPUT F$(I): NEXT : PRINT D$;"CLOSE"
32 HOME :Z$ = "TEACHING PROGRAMMES": GOSUB 70: PRINT : PRINT
34 FOR I = 1 TO FI: PRINT I;". ";F$(I): PRINT : NEXT
36 GOTO 40
37 REM * CALFILE READ ERROR TRAP *
38 PRINT D$;"CLOSE":SE$ = "NEW"
```

```
39  REM * ENTER NEW PROGRAM NAME *
40  VTAB 22: HTAB 1: PRINT "TYPE THE NAME OF THE ";SE$;"
    PROGRAMME --->";: INPUT NA$
41  POKE 216,0
42  IF NA$ = "" THEN 40
45 A =  ASC ( LEFT$ (NA$,1)): IF A < 48 OR A > 57 THEN 47
46  HOME : VTAB 10: PRINT "PROGRAMME NAME CANNOT BEGIN WITH A
    DIGIT": FOR I = 1 TO 3000: NEXT : POKE 216,0: GOTO 32
47 I = 0
48 I = I + 1: IF NA$ = F$(I) THEN 55
49  IF I < FI THEN 48
50 X$ = "CALFILELEN": PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$:
   PRINT FI + 1: PRINT D$;"CLOSE"
52 X$ = "CALFILE": IF FI > 0 THEN  PRINT D$;"APPEND";X$: GOTO
    54
53  PRINT D$;"OPEN";X$
54  PRINT D$;"WRITE";X$: PRINT NA$
55  PRINT D$;"CLOSE": IF SELECT$ = "OLD" THEN 1100: REM * GET
    PREV DATA FOR EDITING *
56  GOTO 100: REM * OPEN NEW DATA FILE *
57  REM **********************
58  REM * GENERAL SUBROUTINES *
59  REM **********************
60  VTAB 24: HTAB 1: PRINT "B(ACK  F(ORWARD  C(ORRECT  Q(UIT
    ";
62  GET A$: IF A$ = "Q" THEN 65
63  REM IFA$ = "E" THEN 150
64  RETURN
65  HOME : VTAB 10: PRINT "DO YOU WANT TO STOP DEVELOPMENT OF
    THIS": PRINT : PRINT : PRINT "PROGRAMME AND RETURN TO THE
    MENU? Y
66  GOTO 60
70  HTAB (40 -  LEN (Z$)) / 2: PRINT Z$
72  RETURN
90  HTAB 9: PRINT "WHEN READY TYPE 'F' -->";: GET A$: IF A$ <
    > "F" THEN 90: RETURN
92  RETURN
97  REM ***********************
```

```
98  REM * MAIN PROGRAM ROUTINE *
99  REM ************************
100  FOR I = 1 TO 5:F(I) = 0: NEXT
105 PAGE = PAGE + 1: IF PA = 30 THEN  HOME : VTAB 10: PRINT
    "THIS IS NOW THE 30TH AND THEREFORE THE": PRINT : PRINT "LAST
    PAGE
106  IF PA > 30 THEN  HOME : VTAB 10: PRINT "YOU HAVE ALREADY
     ENTERED 30 PAGES": PRINT : PRINT "YOU MAY WISH TO RUN THE
     EDITOR
107  GOSUB 200
108 TX = TX + NX%(PAGE)
110  GOSUB 140: IF A$ = "N" THEN  PRINT D$: GOSUB 5000:SE$ =
     "OLD": GOTO 1000
120  IF A$ = "Y" THEN 100
137  POKE 216,0: GOTO 1000
140  HOME : VTAB 10: HTAB 11: PRINT "MORE PAGES?  Y/N  ";: GET
     A$: IF A$ = "Y" OR A$ = "N" THEN  RETURN
142  GOTO 140
146  REM ********************
147  REM * EDITOR SUBROUTINE *
148  REM ********************
150  HOME :Z$ = "PROGRAMME EDITOR": GOSUB 70
151  VTAB 7: PRINT "TYPE PAGE NUMBER TO BE EDITED ": PRINT :
     PRINT
152  PRINT "TYPE 'R' TO RETURN TO PROGRAMME ": PRINT
     "DEVELOPMENT": PRINT
153  PRINT : PRINT "TYPE 'P' FOR PRINT-OUT  ";
154  INPUT A$: IF A$ = "R" THEN  RETURN
155  IF A$ = "P" THEN  GOSUB 6900: GOTO 1000
156  IF  VAL (A$) = 0 OR  VAL (A$) > 30 THEN  GOSUB 2005: GOTO
     150
160  RPAGE =  VAL (A$)
161  IF EDIT$ = "Y" AND PAGE =  > RP THEN 165
162  IF PAGE - 1 < RP THEN  GOSUB 164: GOTO 150
163  GOTO 165
164  VTAB 20: HTAB 1: PRINT "PAGE "; VAL (A$);" HAS NOT BEEN
     COMPLETED YET": FOR I = 1 TO 3000: NEXT : RETURN
165  TPAGE = PAGE:PAGE = RPAGE: FOR I = 1 TO 4:TF(I) = F(I):
```

```
    NEXT :TF(5) = F(5)
166 IF SELECT$ = "OLD" THEN X = NX%(PA)
167 FOR I = 1 TO 4:F(I) = 1: NEXT :F(5) = 6: GOSUB 200
175 PAGE = TPAGE: FOR I = 1 TO 4:F(I) = TF(I): NEXT :F(5) =
    TF(5)
180 HOME : VTAB 10: HTAB 12: PRINT "MORE EDITING? Y/N ";: GET
    A$: IF A$ = "Y" THEN 150
181 IF SELECT$ = "OLD" THEN  RETURN
185 POP : GOTO 200
197 REM *******************
198 REM * TEXT FRAME INPUT *
199 REM *******************
200 HOME : HTAB 11: PRINT "TEXT ENTRY PAGE ";PA: VTAB 8:Z$ =
    "TYPE THE TEXT STRINGS AS REQUIRED": GOSUB 70: VTAB 12: PRINT
    :Z$
201 PRINT : PRINT :Z$ = "TYPE 'NONE' IF NO TEXT IS REQUIRED":
    GOSUB 70: PRINT : PRINT :Z$ = "TYPE 'M' NOW FOR FREE MEMORY":
202 GOSUB 60: IF A$ < > "M" THEN 206
203 VTAB 19: HTAB 1: CALL  - 868:Z$ = "CHECKING FREE MEMORY":
    GOSUB 70:FX =  FRE (0)
204 VTAB 19: HTAB 6: PRINT "FREE MEMORY = ";FX;" BYTES     ":
    GOTO 202
206 IF F(1) = 1 THEN T = NT%(PA): GOTO 250
207 F(1) = 1
208 HOME : PRINT "TEXT PAGE ";PA: PRINT
209 T = 1
210 GOSUB 8000: GOSUB 8022:T$(PA,T) = AA$: IF T$(PA,T) =
    "END" THEN T$(PA,T) = "":T = T - 1: GOTO 240
220 IF T =  > 10 THEN 240
230 T = T + 1: GOTO 210
240 NT%(PA) = T: IF T = 0 THEN 300
250 HOME : FOR I = 1 TO NT%(PA): PRINT T$(PA,I): PRINT : NEXT
260 GOSUB 60: IF A$ = "B" THEN 200
262 IF A$ = "C" THEN 270
264 IF A$ = "F" AND F(2) = 0 THEN 300
265 IF A$ = "F" AND F(2) = 1 THEN 315
266 GOTO 260
267 REM ****************
```

```
268  REM * CORRECT TEXT *
269  REM ****************
270  HOME : FOR J = 1 TO NT%(PA): PRINT J: PRINT T$(PA,J):
     PRINT : NEXT : PRINT J
272  VTAB 24: HTAB 1: PRINT "TYPE NO. FOR CORRECTION ";: GET
     A$: IF A$ =  STR$ (T + 1) THEN T = T + 1: IF T > 10 THEN T =
     =10: GOTO272
273  IF A$ < "1" OR A$ >  STR$ (T) THEN 272
274  HOME : PRINT : PRINT T$(PA, VAL (A$)): VTAB 24: PRINT
     "A(LTER  D(ELETE  L(EAVE  STRING NO. ";A$;" ";: GET Z$: IF Z$
     = "D" T275  IF Z$ = "L" THEN 250
275  IF Z$ = "L" THEN 250
276  HOME : PRINT : PRINT T$(PA, VAL (A$))
278  VTAB 24: HTAB 1: PRINT "TYPE IN THE CORRECTED STRING
     ";: VTAB 1: HTAB 40: GOTO 290
280  IF  VAL (A$) =  > T THEN 286
282  FOR I =  VAL (A$) TO T:T$(PA,I) = T$(PA,I + 1): IF I + 1
     = T THEN 286
284  NEXT
286 T$(PA,T) = "":T = T - 1:NT%(PAGE) = T
288  GOTO 250
290  GOSUB 8000: GOSUB 8022:T$(PA, VAL (A$)) = AA$:NT%(PAGE) =
     T: GOTO 250
297  REM ********************
298  REM * TOPIC FRAME INPUT *
299  REM ********************
300  HOME : PRINT "TOPIC QUESTION FOR PAGE ";PA
301 F(2) = 1
305  VTAB 5: PRINT "TYPE 'NONE' IF NO QUESTION/FEEDBACK IS":
     PRINT : PRINT "REQUIRED FOR THIS FRAME"
310  PRINT : PRINT : HTAB 40: GOSUB 8000: GOSUB 8022:G$(PA) =
     AA$
315  HOME : PRINT G$(PA)
320  GOSUB 60: IF A$ = "B" THEN 250
322  IF A$ = "C" THEN 350
323  IF A$ <  > "F" THEN 320
326  IF A$ = "F" AND G$(PA) = "NONE" THEN  RETURN
328  IF A$ = "F" AND F(3) = 0 THEN 400
```

```
330  IF A$ = "F" AND F(3) = 1 THEN 450
350  HOME : PRINT : PRINT G$(PA): VTAB 1: HTAB 40: GOSUB 8000:
     GOSUB 8022:G$(PA) = AA$: GOTO 315
397  REM ***********************
398  REM * QUESTION FRAME INPUT *
399  REM ***********************
400  HOME : PRINT "QUESTION ENTRY PAGE ";PA: VTAB 10:Z$ =
     "TYPE THE STRINGS AS REQUIRED": GOSUB 70: PRINT :Z$ = "TYPE
     'END' TO F401  PRINT : PRINT :Z$ = "MAXIMUM OF 6 QUESTIONS":
     FLASH : GOSUB 70: NORMAL
401  PRINT : PRINT :Z$ = "MAXIMUM OF 6 QUESTIONS": FLASH :
     GOSUB 70: NORMAL
402 F(3) = 1
404  VTAB 24: GOSUB 90
405  HOME :X = 1: PRINT "PAGE ";PA;" QUES ENTRY"
410  VTAB  PEEK (37) + 2: PRINT  CHR$ (64 + X);".": VTAB  PEEK
     (37) - 1
415  HTAB 40: INPUT X$(PA,X): IF X$(PA,X) = "END" THEN X = X -
     1: GOTO 440
420 X = X + 1: IF X > 6 THEN X = 6: GOTO 440
430  GOTO 410
440 NX%(PAGE) = X
450  HOME : PRINT G$(PA): PRINT : PRINT : FOR I = 1 TO X:
     PRINT X$(PA,I): PRINT : NEXT
460  GOSUB 60: IF A$ = "C" THEN 470
462  IF A$ = "B" THEN 315
464  IF A$ = "F" AND F(4) = 1 THEN 518
465  IF A$ = "F" AND F(4) = 0 THEN 500
466  GOTO 460
470  HOME : FOR I = 1 TO X: PRINT I: PRINT X$(PA,I): PRINT :
     NEXT : PRINT I
472  VTAB 24: HTAB 1: PRINT "TYPE NO. FOR CORRECTION ";: GET
     A$: IF A$ =  STR$ (X + 1) THEN X = X + 1: IF X > 6 THEN X = 6:
     GOTO473  IF A$ < "1" OR A$ >  STR$ (X) THEN 472
473  IF A$ < "1" OR A$ >  STR$ (X) THEN 472
474  HOME : PRINT : PRINT X$(PA, VAL (A$)): VTAB 24: PRINT
     "A(LTER  D(ELETE  L(EAVE  STRING NO. ";A$;" ";: GET Z$: IF Z$
     = "D" T475  IF Z$ = "L" THEN 450
```

```
475  IF Z$ = "L" THEN 450
476  HOME : PRINT : PRINT X$(PA, VAL (A$))
478  VTAB 24: HTAB 1: PRINT "TYPE IN THE CORRECTED STRING
     ";: VTAB 1: HTAB 40: GOTO 490
480  IF  VAL (A$) =  > X THEN 486
482  FOR I =  VAL (A$) TO X:X$(PA,I) = X$(PA,I + 1): IF I + 1
     = X THEN 486
484  NEXT
486 X$(PA,X) = "":X = X - 1:NX%(PAGE) = X
488  GOTO 450
490  INPUT X$(PA, VAL (A$)):NX%(PAGE) = X: GOTO 450
497  REM **********************
498  REM * ANSWER FRAME INPUT *
499  REM **********************
500  HOME : PRINT "CORRECT ANSWERS ENTRY PAGE ";PA
501 F(4) = 1
505  PRINT : PRINT
510  FOR I = 1 TO X: HTAB 1: PRINT X$(PA,I): PRINT : PRINT
     "CORRECT ANSWER = ";
515  GET A$: IF A$ < "1" OR A$ > "5" THEN 515
517 B%(PA,I) =  VAL (A$): PRINT A$: PRINT : PRINT : NEXT
518  HOME : FOR I = 1 TO X: PRINT X$(PA,I): PRINT "ANSWER =
     ";B%(PA,I): PRINT : NEXT
520  GOSUB 60: IF A$ = "C" THEN 500
522  IF A$ = "B" THEN 450
524  IF A$ = "F" AND F(5) > 0 THEN I = 1: GOTO 640
525  IF A$ = "F" THEN I = 0: GOTO 600
526  GOTO 520
597  REM ************************
598  REM * FEEDBACK FRAME INPUT *
599  REM ************************
600  HOME :I = I + 1: PRINT "FEEDBACK FOR QUES ";I;" PAGE";PA
605  VTAB 4: REM  * I=0 AT 525 *
610  IF F(5) < I THEN F(5) = I
615 J = 0
620 J = J + 1: GOSUB 8000: GOSUB 8022:TE$(PA,I,J) = AA$: IF
    TE$(PA,I,J) = "END" THEN TE$(PA,I,J) = "":J = J - 1: GOTO 640
625  IF J =  > 5 THEN 640
```

```
630  GOTO 620
640  HOME : VTAB 4: FOR K = 1 TO 5: IF TE$(PA,I,K) = "END"
     THEN 644
642  PRINT TE$(PA,I,K): PRINT
644  NEXT
646  VTAB 23: PRINT "QUES ";I;
650  GOSUB 60: IF A$ < > "C" THEN 760
660  HOME : FOR K = 1 TO 5: PRINT K: PRINT TE$(PA,I,K): PRINT
     : NEXT : PRINT K
663  VTAB 24: HTAB 1: PRINT "TYPE NO. FOR CORRECTION ";: GET
     A$: IF A$ < "1" OR A$ > "5" THEN 663
664  HOME : PRINT : PRINT TE$(PA,I, VAL (A$)): VTAB 24: PRINT
     "A(LTER  D(ELETE  L(EAVE  STRING NO. ";A$;" ";: GET Z$: IF Z$
     = D THEN 680
665  IF Z$ = "L" THEN 640
666  HOME : PRINT : PRINT TE$(PA,I, VAL (A$))
667  VTAB 24: HTAB 1: PRINT "TYPE IN THE CORRECTED STRING
     ";: VTAB 1: HTAB 40: GOTO 690
668  GOTO 690
680  IF  VAL (A$) =  > 5 THEN 686
682  FOR II =  VAL (A$) TO 4:TE$(PA,I,II) = TE$(PA,I,II + 1):
     IF II + 1 = 5 THEN 686
684  NEXT
685  GOTO 640
686 TE$(PA,I,5) = "": GOTO 640
690  GOSUB 8000: GOSUB 8022:TE$(PA,I, VAL (A$)) = AA$: GOTO
     640
760  IF A$ = "B" THEN I = I - 1: IF I < 1 THEN 518
770  IF A$ = "B" THEN 640
780  IF I =  > X THEN 870
790  IF A$ = "F" AND F(5) > I THEN I = I + 1: GOTO 640
800 J = 0: GOTO 600: REM * PROBLEM HERE ?640 *
815 V =  PEEK (37)
870  HOME : VTAB 10: PRINT "FINISHED EDITING PAGE ";PA" ?  Y/N
     ";: GET A$: IF A$ = "Y" THEN  RETURN
900  GOTO 640
```

```
1000  REM *****************
1001  REM * EDIT OLD FILE *
1002  REM *****************
1010  HOME :Z$ = "EDIT " + NA$: GOSUB 7: PRINT : PRINT : PRINT

1015  VTAB 5: PRINT "1. CORRECT / VIEW / PRINT PAGES": PRINT
1020  PRINT "2. ADD PAGES TO THE END OF THE FILE
1025  PRINT : PRINT "3. INSERT PAGES TO THE FILE
1026  PRINT : PRINT "4. DELETE PAGES FROM THE FILE
1027  PRINT : PRINT "5. SAVE FILE TO DISC
1028  PRINT : PRINT "6. RENAME PROGRAMME"
1029  PRINT : PRINT "7. RETURN TO MAIN MENU
1030  VTAB 20: HTAB 1: GET A$: IF A$ = "1" THEN EDIT$ = "Y":
      GOTO 1076
1031  IF A$ = "2" THEN EDIT$ = "": GOTO 1050
1032  IF A$ = "3" THEN  GOSUB 2000: GOTO 1090
1033  IF A$ = "4" THEN  GOSUB 1500: GOTO 1090
1034  IF A$ = "5" THEN  GOSUB 5000: GOTO 1000
1035  IF A$ = "6" THEN  HOME : VTAB 10: PRINT "TYPE IN THE NEW
      NAME ";: INPUT N1$: PRINT D$;"RENAME";NA$ + " CAL DATA,";N1$ +
      " CAL DATA ": GOTO1000
1036  IF A$ = "7" THEN  RUN
1038  GOTO 1030
1040  HOME : VTAB 10: HTAB 1: IF PA = 30 THEN  PRINT "YOU
      ALREADY HAVE 30 PAGES IN THIS PROG": FOR II = 1 TO 3000: NEXT
      : GOTO 11042  VTAB 10: HTAB 1: PRINT "ENTER THE PAGE NO. AFTER
      WHICH THE       NEW PAGE SHOULD BE INSERTED ";: INPUT A$
1042  VTAB 10: HTAB 1: PRINT "ENTER THE PAGE NO. AFTER WHICH
      THE       NEW PAGE SHOULD BE INSERTED ";: INPUT A$
1043 A1 =  VAL (A$)
1044  IF A1 < 1 OR A1 > PA - 1 THEN 1042
1050  REM
1076  IF EDIT$ < > "Y" THEN SELECT$ = "NEW": GOTO 100: REM *
      CHANGE SINCE ARE ADDING PAGES NOT EDITING *
1080  GOSUB 150
1085 EDIT$ = ""
1090  HOME : VTAB 10: HTAB 1: PRINT "SAVE THE EDITED FILE TO
      DISC? Y/N ";: GET A$: IF A$ = "N" THEN 1000
```

```
1092  IF A$ = "Y" THEN  GOSUB 5000: GOTO 1000
1094  GOTO 1090
1100  REM *************
1110  REM * READ DISC *
1120  REM *************
1130  ONERR  GOTO 1133
1132  GOTO 1140
1133  POKE 33,33:X =  PEEK (222): IF X < > 5 THEN  IF X < >
      6 THEN 1140
1134  VTAB 22: CALL  - 868:Z$ = "PROGRAMME UNAVAILABLE": GOSUB
      7: FOR I = 1 TO 3000: NEXT
1136  POKE 216,0: RUN
1140 D$ =  CHR$ (4)
1142  VTAB 2: PRINT D$
1150  VTAB 22: CALL  - 868:Z$ = "'" + NA$ + "' IS BEING
      LOADED": GOSUB 70: CALL  - 868: VTAB 2
1160 X$ = NA$ + " CAL DATA"
1170  PRINT D$;"OPEN";X$: PRINT D$;"READ";X$
1180  INPUT PAGE
1190  FOR I = 1 TO PA: INPUT NT%(I): INPUT NX%(I): NEXT
1200  FOR I = 1 TO PA: FOR J = 1 TO NT%(I)
1210  INPUT T$(I,J)
1220  IF T$(I,J) = "" THEN T$(I,J) =  CHR$ (13)
1230  IF T$(I,J) = "END" THEN J = NT%(I)
1240  NEXT J,I
1250  FOR I = 1 TO PA: INPUT G$(I)
1260  IF G$(I) = "E" THEN G$(I) = ""
1270  NEXT I
1280  FOR I = 1 TO PA: FOR J = 1 TO NX%(I)
1290  INPUT X$(I,J)
1300  NEXT J,I
1310  FOR I = 1 TO PA: FOR J = 1 TO NX%(I): INPUT B%(I,J):
      NEXT J,I
1320  FOR I = 1 TO PA: FOR J = 1 TO NX%(I)
1330  FOR K = 1 TO 5: INPUT TE$(I,J,K): IF TE$(I,J,K) = "E" OR
      TE$(I,J,K) = "END" THEN TE$(I,J,K) = ""
1340  NEXT K,J,I
1350  INPUT TX
```

```
1360  PRINT D$;"CLOSE"
1380  POKE 216,0
1390  GOTO 1000
1499  REM ***************
1500  REM * DELETE PAGE *
1510  REM ***************
1520  HOME : VTAB 10: PRINT "ENTER THE PAGE NUMBER TO BE
      DELETED ";: INPUT A$
1525 A1 =  VAL (A$)
1530  IF A1 < 1 OR A1 > 30 THEN  GOSUB 2005: RETURN
1540  IF PAGE < A1 THEN  GOSUB 164: RETURN
1550  VTAB 20:Z$ = "RE-ARRANGING DATA FILES": GOSUB 7
1570  FOR II = A1 TO PA: FOR J = 1 TO NT%(J):T$(II,J) = T$(II
      + 1,J): NEXT J
1575 NT%(II) = NT%(II + 1):NX%(II) = NX%(II + 1)
1577 G$(II) = G$(II + 1)
1580  FOR J = 1 TO NX%(J):X$(II,J) = X$(II + 1,J)
1585 B%(II,J) = B%(II + 1,J)
1590 NT%(II) = NT%(II + 1):NX%(II) = NX%(II + 1)
1600  FOR K = 1 TO 5:TE$(II,J,K) = TE$(II + 1,J,K): NEXT K
1610  NEXT J,II
1630  FOR J = 1 TO NT%(PA):T$(PA,J) = "": NEXT
1640 PA = PA - 1
1650  RETURN
1997  REM ***************
1998  REM * INSERT PAGE *
1999  REM ***************
2000  HOME : VTAB 10: HTAB 1: IF PA < 30 THEN 2010
2002  GOSUB 2005: GOTO 2000
2005  VTAB 20: HTAB 1: PRINT "MAXIMUM NO. OF PAGES =30 :
      MINIMUM =1 ": FOR II = 1 TO 3000: NEXT : RETURN
2010  VTAB 10: PRINT "ENTER THE PAGE NO. AFTER WHICH THE":
      PRINT : PRINT "NEW PAGE SHOULD BE INSERTED   ";: INPUT A$:A1 =
      VAL (2011  IF A1 < 0 OR A1 > 30 THEN  GOSUB 2005: POP : GOTO
      1000
2011  IF A1 < 0 OR A1 > 30 THEN  GOSUB 2005: POP : GOTO 1000
2012  IF PAGE < A1 THEN  GOSUB 164: GOTO 1000
2014  VTAB 20:Z$ = "RE-ARRANGING DATA FILES": GOSUB 7
```

```
2015  FOR I = PA + 1 TO 1 STEP  - 1:A = 0: IF I >  VAL (A$) +
      1 THEN A = 1
2020 NT%(I) = NT%(I - A):NX%(I) = NX%(I - A):G$(I) = G$(I - A)
2030  FOR J = 1 TO 10:T$(I,J) = T$(I - A,J): NEXT J
2040  FOR J = 1 TO 6:X$(I,J) = X$(I - A,J)
2050 B%(I,J) = B%(I - A,J)
2060  FOR K = 1 TO 5:TE$(I,J,K) = TE$(I - A,J,K): NEXT K
2070  NEXT J,I
2110  A =  VAL (A$) + 1: FOR J = 1 TO NT%(A):T$(A,J) = "": NEXT
      : FOR J = 1 TO NX%(A):X$(A,J) = "":B%(A,J) = 0: FOR K = 1 TO
      5: TE$(A,J,K) = "":NEXTK,J
2120 TPAGE = PAGE + 1:PA =  VAL (A$) + 1
2130  FOR I = 1 TO 5:F(I) = 0: NEXT : GOSUB 200
2140 PAGE = TP: RETURN
4999  REM *****************
5000  REM * WRITE TO DISC *
5001  REM *****************
5002  POKE 216,0
5003 Z$ = "SELECT SCORING SYSTEM":
5005  HOME : GOSUB 70: VTAB 5: PRINT "5 = 1-5 SCORING SYSTEM":
      PRINT : PRINT "3 = 1-3 SCORING SYSTEM"
5007  VTAB 10: HTAB 1:: PRINT "TYPE IN THE SYSTEM REQUIRED ";:
      GET A$
5008  IF A$ < > "3" THEN  IF A$ < > "5" THEN 5007
5009  IF A$ = "5" THEN SCOAR = 5
5010  IF A$ = "3" THEN SC = 3
5015  VTAB 15: HTAB 1: PRINT "1= ONLY EXACTLY CORRECT ANSWERS
      ACCEPTED": PRINT "2= SCORE EXACT AS 2 POINTS, +/- ONE AS 1"
5020  VTAB 22: HTAB 1:: PRINT "TYPE IN THE SYSTEM REQUIRED ";:
      GET A$: IF A$ = "1" THEN EX = 1: GOTO 5026
5022  IF A$ = "2" THEN EX = 0: GOTO 5026
5024  GOTO 5020
5026 D$ =  CHR$ (4): PRINT D$
5028 X$ = NA$ + " CAL DATA"
5030  PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$
5032  PRINT PAGE
5034  FOR I = 1 TO PA: PRINT NT%(I): PRINT NX%(I): NEXT
5040  FOR I = 1 TO PA: FOR J = 1 TO NT%(I): IF T$(I,J) = ""
```

```
     THEN  PRINT "E": GOTO 5055
5050  PRINT T$(I,J)
5052  IF T$(I,J) = "END" THEN J = 20
5055  NEXT J,I
5060  FOR I = 1 TO PA: IF G$(I) = "" THEN G$(I) = "E"
5070  PRINT G$(I): NEXT
5080  FOR I = 1 TO PA: FOR J = 1 TO NX%(I)
5090  PRINT X$(I,J)
5095  NEXT J,I
5100  FOR I = 1 TO PA: FOR J = 1 TO NX%(I): PRINT B%(I,J):
      NEXT J,I
5120  FOR I = 1 TO PA: FOR J = 1 TO NX%(I)
5130  FOR K = 1 TO 5: IF TE$(I,J,K) = "" THEN  PRINT "E": GOTO
      5150
5140  PRINT TE$(I,J,K)
5150  NEXT K,J,I
5160 TX = 0
5170  FOR I = 1 TO PA:TX = TX + NX%(I): NEXT
5180  PRINT TX: PRINT SCOAR
5190  PRINT EX
5250  PRINT D$;"CLOSE"
6000  HOME : VTAB 10: PRINT "TOTAL NO. OF QUESTIONS = ";TX
6010  RETURN
6896  REM *******************
6898  REM * PRINT SUBROUTINE *
6899  REM ********************
6900  HOME
6910  VTAB 7: HTAB 1: PRINT "PRINT OUT OF PROGRAMME? Y/N ";:
      GET A$: IF A$ <  > "Y" THEN  IF A$ <  > "N" THEN 6910
6920  IF A$ = "N" THEN  RETURN
7000  HOME : PRINT "PRINTER IN SLOT1 ?  Y/N ";: GET A$: IF A$
      <  > "Y" THEN  IF A$ <  > "N" THEN 7000
7010  IF A$ = "Y" THEN PR = 1: GOTO 7050
7020 PR = 0: VTAB 10: HTAB 1: PRINT "TYPE  SCREEN PRINT SPEED
      (100-255) ";: INPUT SP
7030  IF SP < 100 OR SP > 255 THEN 7020
7040  SPEED= SP
7050  VTAB 20: HTAB 1: PRINT "TYPE START PAGE ";: INPUT SP: IF
```

```
     SP > PA - 0 THEN 7050
7060  PRINT D$;"PR#";PR: PRINT : POKE 1785,40: FOR NP = SP TO
      PAGE: HOME : PRINT "PAGE ";NP: PRINT : GOSUB 7090: NEXT :
      PRINT D$;"PR#0"; SPEED = 255
7070  FOR I = 1 TO 3000: NEXT
7080  RETURN
7090  FOR J = 1 TO NT%(NP): PRINT T$(NP,J): PRINT : NEXT J
7100  PRINT : PRINT G$(NP): PRINT : PRINT
7110  FOR J = 1 TO NX%(NP): PRINT X$(NP,J): PRINT
      "SCORE....";B%(NP,J): PRINT : NEXT J
7120  PRINT : PRINT : FOR J = 1 TO NX%(NP): PRINT "FEEDBACK
      QUESTION";J: PRINT : FOR K = 1 TO 5: IF TE$(NP,J,K) = "" THEN
      7140
7130  PRINT TE$(NP,J,K): PRINT
7140  NEXT K: IF PR = 0 THEN  FOR II = 1 TO 1000: NEXT
7150  PRINT : NEXT J
7160  PRINT : FOR II = 1 TO 80: PRINT ".";: NEXT : PRINT :
      PRINT
7170  RETURN
7999  REM ****************
8000  REM * JUSTIFY LINE *
8001  REM ****************
8010  HTAB 40
8015 V1 =  PEEK (37) + 2
8017  VTAB 24: HTAB 1: CALL  - 868: PRINT "AUTOMATIC WORD WRAP
      ?  Y/N ";: GET Z$: HTAB 1: CALL  - 868: IF Z$ = "Y" THEN
      JUST$ = "Y": GOTO 8020
8018  IF Z$ < > "N" THEN 8017
8019 JUST$ = ""
8020  VTAB 24: HTAB 1: PRINT "     TYPE IN THE CORRECTED
      STRING  ";: VTAB V1 - 1: RETURN
8022  HTAB 40:V1 =  PEEK (37) + 2
8023  INPUT AA$
8025  IF JU$ < > "Y" THEN  RETURN
8027  VTAB 24:Z$ = "ORGANISING LINE": GOSUB 7
8028  VTAB V1
8030 A1$ = ""
8040  FOR LL = 40 TO 240 STEP 40: IF  LEN (AA$) < = LL THEN
```

```
     8150
8050 LE =  LEN (AA$)
8060  IF  MID$ (AA$,LL + 1,1) = " " THEN A1$ =  LEFT$
      (AA$,LL): FOR JJ = LL + 1 TO LE:A1$ = A1$ +  MID$ (AA$,JJ +
      1,1): NEXT :AA$ = A1$: GOTO 8150
8070  REM * NOT SPACE THEREFORE WORD AT END OF LINE *
8080  IF  MID$ (AA$,LL,1) = " " THEN 8150
8090  FOR JJ = LL TO 1 STEP  - 1:CO = CO + 1: IF  MID$
      (AA$,JJ,1) = " " THEN 8110
8100  NEXT
8110 A1$ =  RIGHT$ (AA$,LE - LL - 1 + CO)
8120 AA$ =  LEFT$ (AA$,JJ - 1)
8130  FOR II = 1 TO CO:AA$ = AA$ + " ": NEXT
8140 AA$ = AA$ + A1$: GOTO 8150
8150 CO = 0: NEXT LL:V2 =  PEEK (37) + 1: FOR II = V1 TO V2 +
     3: FOR JJ = 1 TO 40: PRINT " ";: NEXT JJ,II: IF V1 > 24 THEN
     V1 = 24
8160  VTAB V1: HTAB 1: PRINT AA$: RETURN
```

```
2    REM ***********
3    REM * CALREAD *
3    REM ***********
5    HOME : TEXT
6    POKE  - 16368,0: REM * CLEAR KEYBOARD BUFFER *
12   DIM T$(30,20),MX$(30,6),MB%(30,6),G$(30)
15   D$ =  CHR$ (4)
16   PRINT D$;"NOMONI,C,O"
20   DIM NT%(30),NX%(30),TE$(30,6,5),Q$(99)
24   REM ***************************
25   REM * OBTAIN AVAILABLE TOPICS *
26   REM ***************************
28   ONERR  GOTO 170
30   D$ =  CHR$ (4):X$ = "CALFILELEN": PRINT D$;"OPEN";X$: PRINT
     D$;"READ";X$: INPUT FILE
32   PRINT D$;"CLOSE":X$ = "CALFILE": PRINT D$;"OPEN"X$: PRINT
     D$;"READ"X$: FOR I = 1 TO FI: INPUT F$(I): NEXT : PRINT
     D$;"CLOSE"34  HOME :Z$ = "PROGRAMME SELECTION": GOSUB 70: VTAB
     4: HTAB 1: FOR I = 1 TO FI: PRINT I;". ";F$(I): PRINT : NEXT
34   HOME :Z$ = "PROGRAMME SELECTION": GOSUB 70: VTAB 4: HTAB
     1: FOR I = 1 TO FI: PRINT I;". ";F$(I): PRINT : NEXT
35   PRINT I;". ";"QUIT"
36   POKE 216,0
40   VTAB 24: HTAB 1: PRINT "TYPE THE NO. OF THE SELECTED
     PROGRAM ";: GET A$:A1 =  VAL (A$): IF A1 =  > 1 AND A1 <  = FI
     THEN 42
41   IF A1 <  > I THEN 40
42   NA$ = F$(A1): IF A1 = I THEN  PRINT D$: PRINT D$;"RUN
     HELLO"
43   VTAB 24: HTAB 1: CALL  - 868:Z$ = "'" + F$(A1) + "' IS NOW
     LOADING": GOSUB 70
44   VTAB 2
45   GOSUB 5000: REM * GET DATA FROM DISC *
50   GOTO 100
70   HTAB (40 -  LEN (Z$)) / 2: PRINT Z$;
72   RETURN
90   VTAB 24: HTAB 9: PRINT "WHEN READY TYPE 1 -->";: GET A$:
     IF A$ <  > "1" THEN 90
```

```
92  RETURN
97  REM ******************
98  REM * MAIN PROGRAMME *
99  REM ******************
100  IF NA$ = "RELAXANTS" THEN  GOTO 107
102  IF SC = 3 THEN  GOSUB 8000: GOTO 107
105  GOSUB 9000
107  MPAGE = 0:NS = 1
110  FOR NP = 1 TO PAGE: GOSUB 200: REM * PAGE HANDLING ROUTINE *
120  NEXT
130  GOTO 6000: REM * GET FINAL SCORE *
170  HOME : POKE 216,0: PRINT "NO PROGRAMMES AVAILABLE": END
```

```
174  REM ******************
175  REM * READ DISC ERROR *
176  REM ******************
180  POKE 216,0
190  HOME : PRINT "THE PROGRAMME YOU SELECTED IS NOT YET":
     PRINT : PRINT "AVAILABLE.  PLEASE SELECT ANOTHER.": FOR I = 1
     TO 3000: NEXT : GOTO 34
197  REM ******************
198  REM * PRINT TEXT PAGE *
199  REM ******************
200  MPAGE = MP + 1
205  IF  LEFT$ (T$(NP,1),4) = "NONE" THEN 300
210  HOME : FOR II = 1 TO NT%(NP): IF T$(NP,II) = "NONE" OR
     T$(NP,II) = "" OR T$(NP,II) = "END" THEN 240
230  PRINT T$(NP,II): PRINT
240  NEXT
290  VTAB 24: GOSUB 90
300  IF G$(NP) = "NONE" THEN  RETURN
305  G$ = G$(NP):J = NX%(NP)
310  FOR II = 1 TO NX%(NP):
320  B(II) = MB%(NP,II):X$(II) = MX$(NP,II)
330  NEXT
360  GOSUB 30000
370  RETURN
311  PRINT D$;"NOMONI,C,O"
4997  REM ************************
4998  REM * OBTAIN DATA FROM DISC *
4999  REM ************************
5000  ON ERR  GOTO 180
5010  PRINT D$
5090  X$ = NA$ + " CAL DATA"
5100  PRINT D$;"OPEN";X$: PRINT D$;"READ";X$
5110  INPUT PAGE
5120  FOR I = 1 TO PA: INPUT NT%(I): INPUT NX%(I): NEXT
5130  FOR I = 1 TO PA: FOR J = 1 TO NT%(I)
5140  INPUT T$(I,J)
5150  IF T$(I,J) = "" THEN T$(I,J) =  CHR$ (13)
5160  IF T$(I,J) = "END" THEN J = NT%(I)
```

```
5170  NEXT J,I
5180  FOR I = 1 TO PA: INPUT G$(I)
5190  IF G$(I) = "E" THEN G$(I) = ""
5200  NEXT I
5210  FOR I = 1 TO PA: FOR J = 1 TO NX%(I)
5220  INPUT MX$(I,J)
5230  NEXT J,I
5240  FOR I = 1 TO PA: FOR J = 1 TO NX%(I): INPUT MB%(I,J):
      NEXT J,I
5250  FOR I = 1 TO PA: FOR J = 1 TO NX%(I)
5260  FOR K = 1 TO 5: INPUT TE$(I,J,K): IF TE$(I,J,K) = "E" OR
      TE$(I,J,K) = "END" THEN TE$(I,J,K) = ""
5270  NEXT K,J,I
5280  INPUT TX
5285  INPUT SCOAR
5287  INPUT EX
5290  PRINT D$;"CLOSE"
5295  POKE 216,0
5300  RETURN
5997  REM ******************
5998  REM * CALCULATE SCORE *
5999  REM ******************
6000  HOME : VTAB 10: HTAB 1: PRINT "YOUR SCORE IS ";C%
6010  PRINT :P =  INT ((C% / TC * 50) + .5)
6020  PRINT "OUT OF ";TC * 2" = ";P;"%"
6030  Q$(NS) =  STR$ (P)
6040  VTAB 24: HTAB 1: GOSUB 90: GOTO 19000
6050  HOME : PRINT "ARE YOU AN ANAESTHETIST ?": PRINT : PRINT
      "1=YES     9=NO    ";: GET A$: IF A$ = "1" THEN Q$(0) = "A":
      GOTO 6080
6055  Q$(0) = "N"
6060  IF A$ <  > "9" THEN 6050
6070  Q$(0) = "N"
6080  GOTO 19000
7997  REM **********************
7998  REM * 1-3 SCORE EXPLAINED *
7999  REM **********************
8000  HOME : PRINT : PRINT "IN THIS PROGRAMME THE SCORING
```

```
     SYSTEM IS-"
8010 PRINT
8020 PRINT "1=MUST DO OR CONSIDER ABSOLUTELY CORRECT"
8030 PRINT "2=A POSSIBLE COURSE OF ACTION              OR
     POSSIBLY CORRECT"
8040 PRINT : PRINT "3=SHOULD NOT DO OR SHOULD CONSIDER WRONG"
8050 PRINT : PRINT "E.G. GIVING A SALINE EMETIC TO AN
     UNCONSCIOUS PATIENT WOULD BE SCORED-3"
8060 PRINT : PRINT "50MG PETHIDINE AS ANALGESIA FOR A 30%
     BURNED PATIENT OF 70 KG WOULD BE        SCORED-2"
8070 GOSUB 90: RETURN
9000 IF SC = 5 THEN 9038
9010 IF SC = 3 THEN 8000
9027 REM *************
9028 REM * 1-5 SCORE *
9029 REM *************
9038 HOME
9040 PRINT : PRINT "IN THIS PROGRAMME THE SCORING SYSTEM IS-"
9045 PRINT
9050 PRINT "1=MUST DO OR CONSIDER ABSOLUTELY CORRECT"
9060 PRINT "2=SHOULD DO OR CONSIDER CORRECT": PRINT
9070 PRINT "3=A POSSIBLE COURSE OF ACTION              OR
     POSSIBLY CORRECT"
9080 PRINT : PRINT "4=SHOULD NOT DO OR SHOULD CONSIDER WRONG"
9090 PRINT "5=MUST NOT DO OR CONSIDER ABSOLUTELY      WRONG
9100 PRINT : PRINT "E.G. GIVING A SALINE EMETIC TO AN
     UNCONSCIOUS PATIENT WOULD BE SCORED-5
9110 PRINT : PRINT "50MG PETHIDINE AS ANALGESIA FOR A 30%
     BURNED PATIENT OF 70 KG WOULD BE        SCORED-4
9120 GOSUB 90: RETURN
18997 REM ************************
18998 REM * WRITE SCORES TO DISC *
18999 REM ************************
19000 D$ =  CHR$ (4)
19004 PRINT D$
19005 PRINT D$;"NOMONI,C,O"
19007 X$ = NA$ + " SCORE COUNT"
19008 ONERR  GOTO 19022
```

```
19010  PRINT D$;"OPEN";X$
19015  PRINT D$;"READ";X$
19020  INPUT X: GOTO 19030
19022  POKE 216,0: PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$:
       PRINT 0
19030  PRINT D$;"CLOSE"
19040  X = X + 1
19050  PRINT D$;"OPEN";X$
19060  PRINT D$;"WRITE";X$
19070  PRINT X
19080  PRINT D$;"CLOSE"
20000  X$ = NA$ + " SCORE"
20010  PRINT D$;"OPEN";X$;X",L4"
20015  REM ** TAKE FROM 1 **
20016  PRINT D$;"WRITE";X$;X",R";0
20017  PRINT Q$(0)
20020  FOR A = 1 TO NS
20030  PRINT D$;"WRITE";X$;X",R";A
20040  PRINT Q$(A)
20050  NEXT
20060  PRINT D$;"CLOSE"
20067  HOME : VTAB 10: PRINT "   THANK YOU FOR YOUR
       CO-OPERATION"
20068  VTAB 23: HTAB 1: PRINT "THE INTRODUCTORY PROGRAMME IS
       LOADING"
20069  FOR I = 1 TO 3000: NEXT
20070  RUN
29997  REM *************************
29998  REM * TEXT/QUESTION HANDLER *
29999  REM *************************
30000  N = 1: IF I < 74 OR I > 79 THEN 30002
30001  HOME : VTAB 10: PRINT "SCORE ALL ";J;" QUESTIONS IN THE
       NEXT": PRINT : PRINT "SECTION.": FOR XI = 1 TO 2000: NEXT
30002  HOME : PRINT G$;H$: PRINT: REM * DISPLAY MAIN TOPIC
       TOPIC QUESTION *
30005  FOR Z = 1 TO J: PRINT : PRINT X$(Z): NEXT Z: REM * DISPLAY
       INDIVIDUAL QUESTIONS *
30006  HTAB 1: VTAB 22: PRINT "SCORE QUESTION ->"; CHR$ (64 +
```

```
      N + JX)
30007 VTAB 23: HTAB 1: PRINT "TO RECALL SCORING SYSTEM, TYPE
      8": PRINT "TO RECALL PREVIOUS TEXT, TYPE 9";
30009 GOTO 30020
30010 VTAB V(N) + 1: HTAB (1): PRINT "  "
30012 VTAB V(N) + 1: HTAB (1)
30020 GOSUB 30100: RETURN
30024 REM **************************
30025 REM * OBTAIN STUDENT'S SCORE *
30026 REM **************************
30100 VTAB 22: HTAB 20: GET X$: IF X$ = "6" OR X$ = "7" THEN
      30100
30101 IF X$ = "8" THEN  GOSUB 9000: POP : GOTO 30002
30102 IF X$ = "9" THEN  GOSUB 60000: POP : GOTO 30002
30103 IF X$ < "1" OR X$ >  STR$ (SCOAR) THEN 30100
30104 REM *************************
30105 REM * ALLOW TO CHANGE SCORE *
30106 REM *************************
30107 HTAB 1: VTAB 22: PRINT "THE SCORE YOU HAVE GIVEN
      QUESTION "; CHR$ (64 + N + JX);" IS ";: FLASH : PRINT X$:
      NORMAL
30108 HTAB 1: VTAB 24: FOR II = 1 TO 38: PRINT " ";: NEXT
30110 HTAB 1: VTAB 23: PRINT "TO CHANGE ANSWER TYPE 1
      OTHERWISE TYPE 2";
30115 GET Z$: IF Z$ < "1" OR Z$ > "2" THEN 30115
30120 IF Z$ = "1" THEN  VTAB 20: HTAB 1: FOR II = 1 TO 40:
      PRINT " ";: NEXT : VTAB 22: HTAB 1: CALL-868: GOTO 30006
30122 IF Z$ = "3" THEN  GOSUB 60000: GOTO 30002: REM * DISPLAY
      PREVIOUS TEXT *
30125 HOME :X% =  VAL (X$)
30129 REM *************************************
30130 REM * ASSESS SCORE AND DISPLAY FEEDBACK *
30131 REM *************************************
30140 IF B(N) = X% THEN  PRINT "GOOD!-YOU SCORE 2":FF = 2:
      GOTO 30180
30150 IF EX = 1 THEN 30170
30160 IF B(N) = X% + 1 OR B(N) = X% - 1 THEN  PRINT "ALMOST
      CORRECT-YOU SCORE 1":FF = 1: GOTO 30180
```

```
30170  W% = W% + 1: PRINT "INCORRECT-":FF = 0
30180  PRINT "CORRECT ANSWER WAS ";B(N): PRINT : FOR K = 1 TO
       5: IF TE$(NP,N,K) = "" THEN 30182
30181  PRINT : PRINT TE$(NP,N,K): REM * DISPLAY FEEDBACK STRINGS *
30182  NEXT :Q$(NS) =  STR$ (FF)
30183  C% = C% + FF:TC = TC + 1
30184  NS = NS + 1
30186  PRINT : PRINT : GOSUB 90
30187  IF N = J THEN  RETURN
30190  N = N + 1: GOTO 30002
32007  Z = 1
60000  REM *************************
60010  REM * DISPLAY PREVIOUS TEXT *
60020  REM *************************
60030  FOR II = 1 TO MP: HOME : FOR JJ = 1 TO NT%(II)
60035  IF T$(II,JJ) = "NONE" THEN 60065
60040  PRINT T$(II,JJ): PRINT
60050  NEXT JJ
60060  VTAB 24: GOSUB 90
60065  NEXT II
60070  RETURN
```

```
19  REM ****************
20  REM * CALFILE-EDIT *
21  REM ****************
30  D$ =  CHR$ (4): GOTO 100
100  HOME :Z$ = "ADD / ALTER NAME FILE": PRINT Z$: FOR I = 1
     TO  LEN (Z$): PRINT "=";: NEXT
102  PRINT
105  VTAB 5: PRINT "1. ADD NAME TO FILE
110  VTAB 7: PRINT "2. ALTER / DELETE NAME ON FILE
120  VTAB 9: PRINT "3. RETURN TO MAIN PROGRAM
125  VTAB 15: PRINT "SELECT REQUIRED NUMBER  ";
130  GET A$: PRINT A$: IF A$ < "1" OR A$ > "3" THEN 100
140  IF A$ = "1" THEN 1200
150  IF A$ = "3" THEN  PRINT D$;"RUN CALFILE-SORT"
1000  REM **************
1005  REM * LIST NAMES *
1008  REM **************
1010  ONERR  GOTO 1014
1012  GOTO 1018
1014  HOME :Z$ = "NO NAMES ON LIST": VTAB 10: HTAB (40 -  LEN
      (Z$)) / 2: PRINT Z$: FOR I = 1 TO 2000: NEXT
1016  POKE 216,0: GOTO 100
1018  NA$ = "CALFILELEN": PRINT D$;"OPEN";NA$
1019  PRINT D$;"READ";NA$: INPUT NI
1020  PRINT D$;"CLOSE";NA$
1021  NA$ = "CALFILE"
1022  PRINT D$;"OPEN";NA$
1023  PRINT D$;"READ";NA$
1024  FOR I = 1 TO NI: INPUT Z$(I): NEXT
1026  PRINT D$;"CLOSE";NA$
1030  POKE 216,0
1035  HOME : PRINT "  NAME": PRINT "  ===="
1038  PRINT
1040  FOR J = 1 TO NI: PRINT : PRINT J;". ";Z$(J): NEXT
1050  VTAB 22: HTAB 1: PRINT "SELECT PROGRAMME NO. FOR
      CORRECTION"
1060  VTAB 24: HTAB 1: PRINT "TYPE 'M' TO RETURN MENU ";
1065  GET A$: IF A$ = "M" THEN 100
```

```
1070  IF A$ < "1" OR A$ >  STR$ (NI) THEN 1065
1080  GOTO 5000
1200  REM ************
1200  REM * ADD NAME *
1200  REM ************
1210  HOME : VTAB 1: HTAB 1: PRINT "TYPE PROGRAMME NAME"::
      PRINT : PRINT "THEN TYPE <RETURN>   ": VTAB 5: INPUT NA$
1280  Z$ = "CALFILELEN"
1285  X$ = ""
1287  ONERR  GOTO 1320
1290  PRINT D$;"OPEN";Z$
1300  PRINT D$;"READ";Z$
1305  INPUT NI
1310  PRINT D$;"CLOSE";Z$
1320  NI = NI + 1
1325  PRINT D$;"OPEN";Z$
1330  PRINT D$;"WRITE";Z$
1332  PRINT NI
1334  PRINT D$;"CLOSE";Z$
1336  POKE 216,0
1338  Z$ = "CALFILE"
1339  IF NI > 1 THEN  PRINT D$;"APPEND";Z$: GOTO 1350
1340  PRINT D$;"OPEN";Z$
1350  PRINT D$;"WRITE";Z$
1355  PRINT NA$
1360  PRINT D$;"CLOSE";Z$
1370  GOTO 100
5000  VTAB 22: HTAB 1: FOR I = 1 TO 38: PRINT " ";: NEXT
5010  VTAB 24: HTAB 1: PRINT "TYPE 'D' TO DELETE PROGRAMME
      NUMBER "; VAL (A$);" ";: GET Z$: IF Z$ < > "D" THEN 100
5020  II =  VAL (A$)
5030  FOR JJ = II TO NI:Z$(JJ) = Z$(JJ + 1): NEXT :Z$(NI) =
      "":NI = NI - 1
5040  VTAB 24: HTAB 1: PRINT "    WRITE CORRECTED FILE TO
      DISC? Y/N ";: GET Z$: IF Z$ = "N" THEN 100
5050  IF Z$ < > "Y" THEN 5040
5052  PRINT D$
5053  ONERR  GOTO 5055
```

```
5054  PRINT D$;"DELETE";Z$( VAL (A$)) + " CAL DATA"
5055  PRINT D$
5060  Z$ = "CALFILELEN"
5070  PRINT D$;"OPEN";Z$: PRINT D$;"WRITE";Z$: PRINT NI: PRINT
      D$;"CLOSE"
5080  Z$ = "CALFILE"
5085  PRINT D$;"DELETE";Z$
5090  PRINT D$;"OPEN";Z$: PRINT D$;"WRITE";Z$: FOR I = 1 TO
      NI: PRINT Z$(I): NEXT : PRINT D$;"CLOSE"
6000  GOTO 100
```

```
4    REM ****************
5    REM * CALFILE-SORT *
6    REM ****************
10   D$ =  CHR$ (4)
30   DIM SM(3 + 1,90),TM(3 + 1,90)
40   DIM A(90 + 1)
84   HOME
85   VTAB 15: HTAB 10: PRINT "OBTAINING DATA FILE "
87   Z$ = "CALFILELEN"
90   PRINT D$;"OPEN";Z$
95   PRINT D$;"READ";Z$
100  INPUT NI
110  PRINT D$;"CLOSE";Z$
112  Z$ = "CALFILE"
115  PRINT D$;"OPEN";Z$
120  PRINT D$;"READ";Z$
130  FOR I = 1 TO NI: INPUT Z$(I): NEXT
180  PRINT D$;"CLOSE";Z$
190  GOSUB 15000
270  HOME
280  VTAB 15: HTAB 10: PRINT "WRITING REARRANGED FILES"
290  PRINT D$;"OPEN";Z$
300  PRINT D$;"WRITE"Z$
310  FOR I = 1 TO NI: PRINT Z$(I): NEXT
360  PRINT D$;"CLOSE"
370  TEXT
371  PRINT : PRINT D$;"RUN CALWRITE"
14990  REM *******************
15000  REM * SORT SUBROUTINE *
15005  REM *******************
15010  HOME : VTAB 15: HTAB 15: PRINT "SORTING"
15050  N = NI
15060  REM * SORT *
15070  F = 0:I = 0
15080  IF Z$(I) <  = Z$(I + 1) THEN 15130
15090  T$ = Z$(I + 1)
15100  Z$(I + 1) = Z$(I)
15110  Z$(I) = T$
```

```
15120  F = 1
15130  I = I + 1: IF I <  = N THEN 15080
15140  IF F = 1 THEN 15070
15150  FOR I = 2 TO NI + 1:Z$(I - 1) = Z$(I): NEXT
15160  RETURN
```

SECTION 2

ROCHE SUITE

```
2   REM ***************
3   REM * HELLO-ROCHE *
4   REM ***************
5   POKE 50944,5
7   TEXT
10  D$ =  CHR$ (4)
20  X$ = "NEXTPROG"
21  GOTO 24
22  HTAB (40 -  LEN (Z$)) / 2: PRINT Z$;: RETURN
23  HOME :Z$ = "GLASGOW ROYAL INFIRMARY": GOSUB 22: PRINT :
    PRINT :Z$ = "RESPIRATORY INTENSIVE CARE UNIT": GOSUB 22:
    PRINT
24  VTAB 10: HTAB 1: PRINT "1....ROCHE 128 MONITOR"
25  VTAB 13: PRINT "2....ROCHE 2000 SYSTEM"
26  VTAB 16: PRINT "3....SERVO 900B VENTILATOR"
27  VTAB 20: HTAB 1: PRINT "TYPE THE REQUIRED NUMBER FOR THE
    MONITOR"
28  PRINT : PRINT "CONNECTED TO THE PATIENT ->";: GET A$: IF
    A$ = "1" THEN N$ = "ROCHE128": GOTO 39
29  IF A$ = "2" THEN N$ = "R2000": GOTO 39
30  IF A$ = "3" THEN N$ = "ROCHESERVO": GOTO 39
35  GOTO 27
39  PRINT A$
40  PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$: PRINT N$
41  PRINT D$;"RUN ROCHESPACE"
```

```
    REM *************************************************
    REM * ROCHESPACE - MODIFICATION OF COMPUTECT UTILITY *
    REM *************************************************
0   REM SPACE (C) SGL 22/08/79
10  D$ = CHR$ (13) + CHR$ (4)
100 SP = PEEK (106) * 256 + PEEK (105) - 128
110 CALL SP
120 X = PEEK (513)
130 IF X = 255 THEN 200
140  X = X * 256 + PEEK (512)
145 HOME : VTAB 10: PRINT "SPACE FREE ON DISC = ";
150 PRINT X;" SECTORS ("; INT (X / 4.96);"%)";
155 FOR I = 1 TO 1000: NEXT
160 IF X > 28 THEN 196
170 HOME : VTAB 10: PRINT CHR$ (7); CHR$ (7);"DISC IS FULL":
    PRINT : PRINT "PUT IN ANOTHER DISC PLEASE"
180 PRINT : PRINT "AND TYPE <RETURN> ";: GET A$: IF A$ < >
    CHR$ (13) THEN 170
185 PRINT D$
190 PRINT D$;"RUN HELLO"
196 PRINT D$;"RUN ROCHEINSTRUCT"
200 PRINT CHR$ (7);"I/O ERROR "; PEEK (512);" READING VTOC";
```

```
4   REM *****************
5   REM * ROCHEINSTRUCT *
6   REM *****************
10  TEXT
12  HOME :Z$ = "GLASGOW ROYAL INFIRMARY": GOSUB 14:Z$ =
    "RESPIRATORY INTENSIVE CARE UNIT": GOSUB 14
13  Z$ = "ROCHE CARDIOVASCULAR PROGRAMS": GOSUB 14: GOTO 15
14  PRINT : HTAB (40 -  LEN (Z$)) / 2: PRINT Z$: RETURN
15  D$ =  CHR$ (4): IF A$ = "1" THEN 20
17  VTAB 20: HTAB 4: PRINT "DO YOU WISH INSTRUCTIONS? Y/N  ";:
    GET A$: PRINT A$: IF A$ = "N" THEN 20
18  IF A$ = "Y" THEN 40
19  GOTO 17
20  VTAB 19:Z$ = "NAME SELECTION PROGRAM NOW LOADING": GOSUB
    14
30  PRINT D$;"RUN ROCHESETUP"
40  PRINT :X$ = "NEXTPROG": PRINT D$;"OPEN";X$: PRINT
    D$;"READ";X$: INPUT N$: PRINT D$;"CLOSE"
50  IF N$ = "ROCHE128" THEN 100
54  REM ***************************
55  REM * ROCHE 2000 INSTRUCTIONS *
56  REM ***************************
60  HOME : PRINT "   THIS PROGRAM COLLECTS DATA ON-LINE   FROM
    A ROCHE 2000 MONITOR AND DISPLAYS  THE RESULTS AS A TREND
    GRAPH"
65  PRINT : PRINT "   IF A PRINTER IS AVAILABLE, THE DATA
    FROM THE MONITOR TOGETHER WITH DERIVED
70  PRINT "RESULTS ARE PRINTED OUT AS A PERMANENT  RECORD."
75  PRINT : PRINT : PRINT "   THE PROGRAM IS DRIVEN BY
    SELECTING   ONE FROM SEVERAL CHOICES DISPLAYED ON   THE
     SCREEN.
80  FOR I = 1 TO 3000: NEXT
85  VTAB 20: PRINT "WHEN READY PRESS ANY KEY ";: GET A$
90  HOME : PRINT "THE ADC CARD SHOULD BE IN SLOT 4 AND THE":
    PRINT "CLOCK CARD SHOULD BE IN SLOT 3."
92  PRINT : PRINT : PRINT "THESE CARDS WILL NORMALLY BE ":
    PRINT : PRINT "INSERTED BEFORE YOU USE THE SYSTEM.
94  FOR I = 1 TO 3000: NEXT
```

```
95  VTAB 20: PRINT "WHEN READY PRESS ANY KEY ";: GET A$
96   GOTO 300
97   REM *************************
98   REM * ROCHE 128 INSTRUCTIONS *
99   REM *************************
100  HOME : PRINT "   THIS PROGRAM COLLECTS DATA ON-LINE
     FROM A ROCHE 128 MONITOR AND DISPLAYS   THE RESULTS AS A
     TREND GRAPH.": PRINT : PRINT : PRINT "   IF A PRINTER IS
     AVAILABLE,
     THE DATA  FROM THE MONITOR TOGETHER WITH DERIVED"
102  PRINT "RESULTS ARE PRINTED OUT AS A PERMANENT  RECORD."
110  PRINT : PRINT : PRINT "   THE PROGRAM IS DRIVEN BY
     SELECTING   ONE FROM SEVERAL CHOICES DISPLAYED ON   THE
     SCREEN.
115  FOR I = 1 TO 3000: NEXT
116  VTAB 20: PRINT "WHEN READY PRESS ANY KEY ";: GET A$
117  HOME
170  PRINT : PRINT "    YOU WILL ALSO BE ASKED TO ENTER
     DATAINTO THE COMPUTER.  THE CHARACTERS ARE  TYPED ONE
     AFTER THE OTHER.  IF A WRONG  CHARACTER IS TYPED, IT CAN BE
     DELETED BYTYPING THE 'BACKSPACE' OR LEFT ARROW KEY"
180  VTAB 10: PRINT "    ONCE THE CORRECT CHARACTER HAS BEEN
     TYPED, YOU MUST ENTER THIS DATA INTO THECOMPUTER BY
     PRESSING THE <RETURN> KEY."
190  VTAB 22: PRINT "WHEN READY PRESS ANY KEY ";: GET A$
200  HOME : PRINT "   THE ROCHE MUST BE CONNECTED TO THE
     CARD IN SLOT 5 AND THE CLOCK CARD MUST  BE IN SLOT 3."
220  PRINT : PRINT : PRINT "   THE PRINTER IS CONNECTED TO
     SLOT 1.
222  PRINT
225  PRINT "(ALL THESE CONNECTIONS WILL NORMALLY BE MADE
     BEFORE YOU USE THE SYSTEM)
230  PRINT : PRINT : PRINT "   CHANNEL ONE ON THE ROCHE IS
     USUALLY  CONNECTED TO THE SYSTOLIC PRESSURE AND  CHANNEL
     TWO TO THE CVP OR PAWP.
232  PRINT : PRINT : PRINT "   THE LEAD FROM THE CORE
     TEMPERATURE   SENSOR IS PLUGGED INTO TEMP 1 AND THE
     PERIPHERAL PROBE INTO TEMP 2.
```

```
240  VTAB 24: PRINT "WHEN READY TYPE A KEY ";: GET A$
300  HOME : FOR I = 1 TO 3: PRINT "CHR$(7)": NEXT
310  VTAB 5:Z$ = "REMEMBER": FLASH : GOSUB 14: NORMAL : VTAB
     10: PRINT "    YOU CANNOT UNWITTINGLY DESTROY DATA  OR
     DAMAGE THE SYSTEM."
320  VTAB 13: PRINT "    IF YOU ARE IN DIFFICULTY AT ANY
     POINTPRESS THE <RETURN> KEY AND YOU WILL BE  RETURNED TO
     THE SELECTION MENU."
330  VTAB 20: HTAB 1: PRINT "WHEN READY TYPE 1 ";: GET A$: IF
     A$ <  > "1" THEN 330
340  HOME : VTAB 5: PRINT "    THESE ARE THE ONLY OPERATIONS
     WHICH  YOU WILL HAVE TO PERFORM TO USE THIS    PROGRAM.
350  PRINT : PRINT
360  PRINT : PRINT "1. SELECTION OF A NUMBER OR A LETTER
     FROM A MENU DISPLAYED ON THE SCREEN.
370  PRINT
380  PRINT : PRINT "2. ENTRY OF DATA INTO THE COMPUTER. THIS
     REQUIRES THE <RETURN> KEY TO BE          PRESSED ONCE THE
     CORRECT NUMBERS HAVE   BEEN TYPED ON TO THE SCREEN."
390  VTAB 22: HTAB 1: PRINT "TO LOAD THE FILE PROGRAM TYPE '1'
      ";: GET A$: PRINT A$: IF A$ <  > "1" THEN 390
400  GOTO 12
```

```
2   REM **************
3   REM * ROCHESETUP *
4   REM **************
5   SLOTCLOCK = 3
10  D$ =  CHR$ (4)
12  PRINT D$;"IN#";SL: PRINT D$;"PR#";SL: INPUT T$: PRINT
    D$;"IN#0": PRINT D$;"PR#0":REM * GET DATE/TIME *
13  HOME : PRINT "  DATE","TIME": VTAB 4
14  PRINT  MID$ (T$,4,2);"/"; MID$ (T$,1,2);"/";"82", MID$
    (T$,7,2);"."; MID$ (T$,10,2)
15  VTAB 10: PRINT "SET TIME ON THE COMPUTER CLOCK  Y/N ";:
    GET A$: PRINT A$: IF A$ = "N" THEN 50
16  IF A$ <  > "Y" THEN 15
18  GOSUB 20: GOTO 40
19  REM **************************************************
20  REM * WRITE ROCHESETUP INTO 2ND POSITION OF NEXTPROG *
20  REM **************************************************
22  X$ = "NEXTPROG"
25  PRINT D$;"OPEN";X$: PRINT D$;"POSITION";X$;",R";1
30  PRINT D$;"WRITE";X$: PRINT "ROCHESETUP": PRINT D$;"CLOSE"
35  RETURN
40  PRINT D$;"RUN SET TIME"
45  HTAB (40 -  LEN (Z$)) / 2: PRINT Z$;: RETURN
49  REM ******************************
50  REM * OBTAIN CURRENT PATIENT NAME *
60  REM ******************************
144  ONERR  GOTO 220
150  PRINT D$;"NOMONI,C,O"
160  PRINT D$;"OPEN NAMEFILE"
170  PRINT D$;"READ NAMEFILE"
180  INPUT NA$: INPUT NO$: INPUT BW
190  PRINT D$;"CLOSENAMEFILE"
200  HOME : VTAB 10: PRINT "FILE FOR ";NA$;" Y/N ";: GET A$:
     IF A$ = "Y" THEN  POKE 216,0: GOTO 280
210  IF A$ <  > "N" THEN 200
220  POKE 216,0
230  GOSUB 310
240  PRINT D$;"OPEN NAMEFILE"
```

```
250  PRINT D$;"WRITENAMEFILE"
260  PRINT NA$: PRINT NO$: PRINT BW
270  PRINT D$;"CLOSENAMEFILE"
274  REM ******************************
275  REM * ALLOW CALIBRATION OF SERVO *
276  REM ******************************
280  HOME : VTAB 5: PRINT "DO YOU WISH TO CALIBRATE THE
     SEIMENS": PRINT "SERVO VENTILATOR? Y/N ";: GET A$: PRINT
     A$: IF A$ = "N" THEN 295
285  IF A$ <  > "Y" THEN 280
290  PRINT : GOSUB 20: PRINT D$;"RUN SERVOCALIBRATE"
295  ONERR  GOTO 309
296  REM ************************
297  REM * GET MAIN CVS PROGRAM *
298  REM ************************
300  X$ = "NEXTPROG": PRINT D$;"OPEN";X$: PRINT D$;"READ";X$:
     INPUT N$: PRINT D$;"CLOSE"
303  HOME : VTAB 10:Z$ = "PLEASE WAIT": GOSUB 45: PRINT :
     PRINT :Z$ = "THE NEXT PROGRAM IS LOADING": GOSUB 45:
     PRINT
305  IF N$ = "R2000" THEN  PRINT D$;"RUN R2000CALIBR"
307  PRINT D$;"RUN ROCHE128"
309  POKE 216,0: PRINT : PRINT D$;"RUN HELLO"
310  REM **********************
311  REM * ADDNAME SUBROUTINE *
312  REM **********************
315  GOTO 380
320  REM * ADD LEADING ZEROS *
330  IF  LEN (TT$) < TT THEN TT$ = "0" + TT$: GOTO 330
340  RETURN
350  REM * ADD TRAILING DOTS *
360  IF  LEN (TT$) < TT THEN TT$ = TT$ + ".": GOTO 360
370  RETURN
374  REM **************
375  REM * ENTER NAME *
376  REM **************
380  HOME : VTAB 1: HTAB 1: PRINT "TYPE 'SURNAME' <SPACE>
     'INITIAL'": PRINT : PRINT "THEN TYPE <RETURN>   ": VTAB
```

```
    5: INPUT NA$
390 IF  LEN (NA$) > 20 THEN 440
400 FOR I = 1 TO 20:A$ =  MID$ (NA$,I,1): IF A$ < "0" OR A$ >
    "9" THEN  NEXT : GOTO 420
410 GOTO 440
420 FOR I = 3 TO 20: IF  MID$ (NA$,I,1) = " " THEN 450
430 NEXT
440 FOR I = 1 TO  LEN (NA$): PRINT " ";: PRINT : PRINT "ONLY
    20 CHARS INCLUDING 1 SPACE ALLOWED": FOR I = 1 TO 2000:
    NEXT : GOTO 380
450 TT$ = NA$:TT = 20: GOSUB 350:NA$ = TT$
460 VTAB 10: HTAB 33: FOR I = 1 TO  LEN (NO$) + 1: PRINT "
    ";: NEXT : PRINT
464 REM *********************
465 REM * ENTER HOSPITAL NO *
466 REM *********************
470 VTAB 10: PRINT "TYPE HOSPITAL NO. THEN <RETURN> ";: INPUT
    NO$
480 FOR I = 1 TO  LEN (NO$): IF  MID$ (NO$,I,1) < "0" OR
    MID$ (NO$,I,1) > "9" THEN NO$ = "11111111"
490 NEXT
500 IF  LEN (NO$) > 6 OR  LEN (NO$) < 3 THEN  VTAB 10: HTAB
    32: FOR I = 1 TO 1 +  LEN (NO$): PRINT " ";: NEXT : PRINT
    : VTAB 10: PRINT "MIN OF 3 & MAX OF 6 DIGITS ALLOWED": FOR
    I = 1 TO 3000: NEXT : GOTO 460
510 TT$ = NO$:TT = 6: GOSUB 320:NO$ = TT$
520 VTAB 15: HTAB 33: FOR I = 1 TO  LEN (BW$) + 2: PRINT "
    ";: NEXT : PRINT
524 REM *********************
525 REM * ENTER BODY WEIGHT *
526 REM *********************
530 VTAB 15: HTAB 1: PRINT "TYPE BODY WEIGHT (KG) THEN
    <RETURN> ";: INPUT BW$
540 FOR I = 1 TO  LEN (BW$): IF  MID$ (BW$,I,1) < "0" OR
    MID$ (BW$,I,1) > "9" THEN BW$ = "111111"
550 NEXT
560 IF  LEN (BW$) > 3 OR  LEN (BW$) < 2 THEN  VTAB 15: HTAB
    33: FOR I = 1 TO 1 +  LEN (NO$): PRINT " ";: NEXT : PRINT
```

```
    : VTAB 15: PRINT "MIN OF 2 & MAX OF 3 DIGITS ALLOWED": FOR
    I = 1 TO 2000: NEXT : GOTO 520
565 IF  VAL (BW$) > 200 THEN 520
570 TT$ = BW$:TT = 3: GOSUB 320:BW$ = TT$
580 N$ = "": FOR I = 1 TO 20: IF  MID$ (NA$,I,1) < > "."
    THEN N$ = N$ +  MID$ (NA$,I,1): NEXT
590 NA$ = N$:BW =  VAL (BW$): RETURN
```

```
1    REM *******************************************************
2    REM * SET TIME - MODIFICATION OF MOUNTAIN HARDWARE ROUTINE *
3    REM *******************************************************
20   D$ =  CHR$ (4)
21   PRINT D$;"NOMON I,O ,C"
30   CALL  - 936
40   VTAB 10
50   PRINT "MOUNTAIN HARDWARE'S APPLE CLOCK"
60   VTAB 13
70   PRINT "DISPLAY OR SET THE TIME PROGRAM"
75   PRINT : PRINT
80   GOSUB 10000:SLOT = C: REM  FIND SLOT
81   IF C =  - 1 THEN  END
90   PRINT "DO YOU WANT TO SET THE TIME? Y/N ";: GET I$
95   IF I$ = "N" THEN  CALL  - 936: GOTO 2030
100  REM
110  REM  **** POKE IN THE ADVANCE ROUTINE AT LOCATION $1000
120  REM
130  FOR I = 1 TO 68
140  READ J
150  POKE 767 + I,J
160  NEXT I
300  PRINT
301  PRINT "GIVE THE CURRENT TIME PLUS 30 SECONDS"
302  PRINT
310  INPUT "INPUT THE MONTH (1-12) ";MTH
320  INPUT "INPUT THE DAY (1-31) ";D
330  INPUT "INPUT THE HOUR (0-23) ";H
340  INPUT "INPUT THE MINUTE (0-59) ";M
350  INPUT "INPUT THE SECONDS (0-59) ";S
360  PRINT
365  PRINT "HIT RETURN WHEN YOU HAVE SET THE LEAP"
370  PRINT "SWITCH CORRECTLY, AND ARE SWITCHED FOR"
380  INPUT "WRITING TO THE CLOCK ";I$
390  PRINT : PRINT
500  REM
501  REM **** CHECK LEAP YEAR SWITCH
502  REM
```

```
505  REM   IF L=1 THEN IT'S A LEAP YEAR
510  L =  PEEK (49280 + 16 * SLOT)
511  L =  INT (L / 64)
512  IF L > 1 THEN L = L - 2
600  REM
601  REM **** FIND DAYS TO DATE -- DTD --
602  REM
605  DTD = 0
610  FOR I = 1 TO MTH
620  READ J
630  DTD = DTD + J
640  NEXT I
650  DTD = DTD + D - 1
660  IF L = 1 AND MTH > 2 THEN DTD = DTD + 1
700  REM
701  REM **** CALCULATE SECONDS TO DATE --STD --
702  REM
710  STD = DTD * 86400 + H * 3600 + M * 60 + S
800  REM
801  REM **** PREPARE SECONDS FOR CLOCK
802  REM
810  TEMP = 896: REM  RAM STORAGE AREA
820  S0 =  INT (STD / 2 ^ 20)
825  POKE TEMP,S0
830  STD = STD - S0 * 2 ^ 20
840  S1 =  INT (STD / 2 ^ 12)
850  POKE TEMP + 1,S1
860  STD = STD - S1 * 2 ^ 12
870  S2 =  INT (STD / 2 ^ 4)
880  POKE TEMP + 2,S2
890  STD = STD - S2 * 2 ^ 4
900  POKE TEMP + 3,STD * 16
910  REM
911  REM **** ALSO SAVE N2 AND N7
912  REM
920  POKE TEMP + 4,SLOT * 16 + 2
930  POKE TEMP + 5,SLOT * 16 + 7
1000  REM
```

```
1001  REM **** STOP CLOCK AND CALL ADVANCE ROUTINE
1002  REM
1005  SR = 49280 + SLOT * 16 + 5
1006  SP = 49280 + SLOT * 16 + 6
1010  I =  PEEK (SP)
1020  CALL 768: REM  CALL THE MACHINE LANG ADVANCE ROUTINE
2000  INPUT "HIT RETURN AT EXACT TIME ";I$
2010  I =  PEEK (SR): REM   START CLOCK
2020  CALL  - 936
2030  PRINT D$
2032  REM **********************
2035  REM * OBTAIN NEXT PROGRAM *
2037  REM **********************
2040  X$ = "NEXTPROG": PRINT D$;"OPEN";X$: PRINT D$;"READ";X$:
      INPUT N$: INPUT N$: PRINT D$;"CLOSE"
2050  PRINT D$;"RUN";N$
4999  REM **** ADVANCE SUBROUTINE DATA
5000  REM
5010  DATA 72,8,138,72,152,72,174,132
5020  DATA 3,172,133,3,189,128,192,205
5030  DATA 130,3,208,8,189,129,192,205
5040  DATA 131,3,240,6,185,128,192,76
5050  DATA 12,3,202,202,200,189,128,192
5060  DATA 41,31,205,128,3,208,8,189
5070  DATA 129,192,205,129,3,240,6,185
5080  DATA 128,192,76,37,3,104,168,104
5090  DATA 170,40,104,96
5999  REM **** MONTH DATA
6000  DATA  0,31,28,31,30,31,30,31,31,30,31,30
10000  REM  APPLE CLOCK SLOT FINDER
10001  REM  ------SUBROUTINE-------
10010  REM  MOUNTAIN HARDWARE, INC
10020  REM  USES VARIABLES C1,C2,& C
10030  REM  RETURNS WITH C = THE
10031  REM  SLOT # THE CLOCK IS IN
10040  REM  IF NOT FOUNT C =-1
10050  C =  - 1
10060  FOR C2 = 1 TO 7
```

```
10070  C1 =  PEEK ( - 12289): REM  TURN OFF ALL ROMS
10100  IF  PEEK ( - 16384 + C2 * 256 + 19) = 3 OR  PEEK ( -
       16384 + C2 * 256 + 19) = 177 OR  PEEK ( - 16384 + C2 *
       256 + 19) = 236 OR  PEEK ( - 16384 + C2 * 256 + 19) =
       44 THEN GOTO 10105
10101  NEXT C2
10103  GOTO 10120
10105  IF  PEEK ( - 16384 + C2 * 256 + 21) = 248 OR  PEEK ( -
       16384 + C2 * 256 + 21) = 253 OR  PEEK ( - 16384 + C2 *
       256 + 21) = 7 OR  PEEK ( - 16384 + C2 * 256 + 21) = 104
       THEN  GOTO 10150
10110  NEXT C2
10120  PRINT "APPLE CLOCK NOT FOUND"
10130  C1 =  PEEK ( - 12289): REM  KILL ALL ROMS
10140  RETURN
10150  C = C2
10160  C1 =  PEEK ( - 12289): REM  KILL ALL ROMS
10170  RETURN
```

```
************
* ROCHE128 *
************
```

```
1 LOMEM: 28000: TEXT :DR = 30:D$ = CHR$ (4):PT$ =
  "SYSCHAN1":MENU$ = "Y": DIM SH(37),Z$(270): VTAB 10: ONERR
  GOTO 4: REM * SET FLAGS *
2 X$ = "SHUNTFILE": PRINT D$;"OPEN";X$: PRINT D$;"READ";X$:
  INPUT DR: INPUT PR: INPUT CV$: INPUT FLAG$: INPUT PT$:
  INPUT SERVO$: REM * CHECK IF RETURNING FROM PREVIOUS PROGRAM *
3 PRINT D$;"CLOSE": PRINT D$;"DELETE";X$
4 POKE 216,0: GOSUB 360: GOTO 9
6 FOR I = 1 TO 39: PRINT " ";: NEXT : RETURN
7 HTAB (40 - LEN (Z$)) / 2: PRINT Z$: RETURN
9 PRINT D$;"BLOAD CHARGEN A$6000 L$100"
10 PRINT D$;"BLOAD COLOSSAL.SET A$6900 L$300"
11 DEF FN PLT(X) = VAL ( MID$ (Z$(X),Z0,Z1)): DEF FN
   TIME(X) = VAL ( MID$ (Z$(X),1,6))
12 PRINT D$;"NOMONI,C,O": VTAB 20: GOSUB 6: PRINT
13 REM * OBTAIN CURRENT NAME *
14 PRINT D$;"OPEN NAMEFILE": PRINT D$;"READ NAMEFILE"
18 INPUT NA$: INPUT NO$: PRINT D$;"CLOSE NAMEFILE"
19 VTAB 20: HTAB 1: GOSUB 6
21 VTAB 20: HTAB 1: PRINT "NAME";: HTAB 15: PRINT NA$
22 PRINT D$;"BLOAD LOAD.OBJO"
25 PRINT D$;"BLOAD CARD(A$C500;L$FF)
26 PRINT D$;"PR#5": PRINT : PRINT D$;"PR#0": REM * INITIALISE
   SERIAL CARD TO ROCHE 128 *
27 IF FLAG$ < > "" THEN 30
28 VTAB 20: HTAB 14: PRINT "PRINTER Y/N -->";: GET A$: PRINT
   A$: IF A$ = "N" THEN PR = 0: GOTO 30
29 PR = 1: PRINT D$;"PR#1": PRINT : PRINT " ": POKE -
   12528,7: POKE - 12526,83: POKE - 12525,64: POKE -
   12529,255: POKE - 12524,0: PRINT : PRINT D$;"PR#0"
   REM * INIALISE PRINTER *
30 X$ = NA$ + " TEXTLEN": ONERR GOTO 32
```

```
31  PRINT D$;"OPEN";X$: PRINT D$;"READ"X$: INPUT LT: PRINT
    D$;"CLOSE": GOTO 33: REM * OBTAIN NO OF TEXT FILES *
32  POKE 216,0: PRINT D$;"OPEN";X$: PRINT D$;"WRITE"X$: PRINT
    LT: PRINT D$;"CLOSE"
33  VTAB 20: HTAB 1: GOSUB 6: PRINT :X$ = NA$ + " CVS
    DATALEN": ONERR  GOTO 50
34  PRINT D$;"OPEN";X$: PRINT D$;"READ";X$: INPUT FILE: PRINT
    D$;"CLOSE"
35  X$ = NA$ + " CVS DATA"
37  REM ********************
38  REM * OBTAIN CVS FILES *
39  REM ********************
41  PRINT D$;"OPEN";X$
42  PRINT D$;"READ";X$
46  VTAB 20: HTAB 9: PRINT "OBTAINING FILE "
47  FOR I = 1 TO FILE: VTAB 20: HTAB 30: PRINT I: INPUT Z$(I)
49  NEXT
50  PRINT D$;"CLOSE": POKE 216,0
51  GOSUB 24000:DAY = D:MAY = MT: IF CV$ < > "" THEN 60
52  HOME : VTAB 5: PRINT "IS THE SECOND PRESSURE CHANNEL
    RECORDING": PRINT : HTAB 6: PRINT "PULMONARY ARTERY
    PRESSURE ?"
53  VTAB 10: HTAB 18: PRINT "Y/N  ";
54  GET A$: IF A$ = "Y" THEN CVP$ = "N": GOTO 57
    REM * SET 'CVP' FLAG *
55  IF A$ = "N" THEN CVP$ = "Y": GOTO 57
56  GOTO 54
57  POKE  - 16368,0: HOME : VTAB 5: PRINT "IS THE APPLE
    CONNECTED TO THE SERVO ?": VTAB 10: HTAB 18: PRINT "Y/N
    ";: FOR I = 1 TO 1000: NEXT : GET A$: IF A$ = "Y" THEN SERVO$
    = "Y": GOTO 60
58  IF A$ < > "N" THEN 57: REM * SET 'SERVO' FLAG *
59  REM * DATA STORED FROM LOCATION 896 BY LOAD.OBJ0 *
60  CALL 768:D$ =  CHR$ (4): PRINT D$: IF  PEEK (880) = 1 THEN
    HOME : PRINT  CHR$ (7): FLASH : VTAB 12:Z$ = "? ROCHE
    DISCONNECTED": GOSUB 7: NORMAL : GOTO 60
62 Z$ = ""
65 TP$ =  MID$ (T$,7,2) + "." +  MID$ (T$,10,2):Z$ =  STR$ (
```

```
   INT (S / 60)): FOR II = 1 TO 6: IF  LEN (Z$) < 6 THEN Z$ =
   "0" + Z$: NEXT
68 PFLAG$ = "N"
69 REM * SELECT MEMORY LOCATIONS FOR DATA RETRIEVAL *
70 I = 903: GOSUB 3000:HR$ = R$
72 I(1) = 910:I(2) = 916:I(3) = 913: IF PT$ <  > "SYSCHAN1"
   THEN  FOR I = 1 TO 3:I(I) = I(I) + 10: NEXT
73 PF$ = "Y"
74 I = I(1): GOSUB 3000:P1$ = R$
75 QQ$ =  STR$ ( INT ( VAL (HR$) *  VAL (P1$) / 1E2)):QQ = 3:
   GOSUB 2000:RP$ = QQ$
76 I = I(2): GOSUB 3000:P2$ = R$
78 I = I(3): GOSUB 3000:P3$ = R$
80 I(4) = 920:I(5) = 926:I(6) = 923
82  IF PT$ <  > "SYSCHAN1" THEN  FOR I = 4 TO 6:I(I) = I(I) -
   10: NEXT
84 I = I(4): GOSUB 3000:P4$ = R$
86 I = I(5): GOSUB 3000:P5$ = R$
88 I = I(6): GOSUB 3000:P6$ = R$
89 PF$ = "N"
90 I = 930: GOSUB 3000:T1 =  VAL (R$) / 1E1:I = 933: GOSUB
   3000:T2 =  VAL (R$) / 1E1
92 T1$ =  STR$ (T1): IF  LEN (T1$) < 4 THEN T1$ = T1$ + ".0"
93  IF  LEN (T1$) > 4 THEN T1$ =  LEFT$ (T1$,4)
94 T3 =  INT ((T1 - T2) * 10 + .5) / 1E1:QQ$ =  STR$ (T3):QQ =
   3: GOSUB 2000:T2$ = QQ$
98  IF  VAL (P1$) < 40 OR  VAL (P1$) > 300 THEN 60
100  REM *****************
101  REM * PRINT ROUTINE *
102  REM *****************
103  PRINT D$
104  PRINT D$;"PR#";PR
106  IF FLAG$ = "Y" THEN 126
107  GOSUB 24000: PRINT D$;"PR#";PR: FOR I = 1 TO 4: PRINT :
     NEXT : HTAB 1: PRINT NA$;: HTAB 20: PRINT NO$;: HTAB 31:
     PRINT D;"/";MT;"/";"1982"
108  HTAB 1: FOR I = 1 TO  LEN (NA$): PRINT "=";: NEXT : HTAB
20: FOR I = 1 TO  LEN (NO$): PRINT "=";: NEXT : HTAB 31: PRINT
```

```
   "==========": PRINT : PRINT
110 XX$ = "  PULMONARY DATA"
115  HTAB 14: PRINT "SYSTEMIC DATA";: HTAB 35: PRINT "CARDIAC
     DATA"; SPC( 4);XX$; SPC( 8);"TEMP"
116  PRINT
120  HTAB 2: PRINT "TIME";: HTAB 9: PRINT "HR";" SYS/DIA MEA
     CVP SVR"; SPC( 4);"C.O. SVL RPP"; SPC( 3);"SYS/DIA MEA
     PAWP PVR";
124  PRINT  SPC( 3);"CORE DIF"
125  FLAG$ = "Y"
126  PRINT D$;"PR#";PR
127  REM ***********************************
128  REM * PRODUCE CONCATENATED DATA STRING *
129  REM ***********************************
130  IF  VAL (VP$) =  - 99 THEN VP$ = "---"
135  IF  VAL (PW$) =  - 99 THEN PW$ = "---"
140  IF  VAL (CO$) = 0 THEN 160
145  QQ$ = CO$:QQ = 4: GOSUB 2000:CO$ = QQ$
150  IF A = 184 THEN 20153
155  QQ$ =  STR$ (( VAL (P3$) -  VAL (VP$)) /  VAL (CO$)):QQ =
     4: GOSUB 2000:SR$ = QQ$
158  QQ$ =  STR$ (( VAL (P6$) -  VAL (PW$)) /  VAL (CO$)):QQ =
     4: GOSUB 2000:PR$ = QQ$
160  IF CVP$ = "Y" THEN VP$ = P6$:P4$ = "---":P5$ = P4$:P6$ =
     P5$
165  IF VP$ = "---" THEN VP$ = "-99"
167  IF PW$ = "---" THEN PW$ = "-99"
170  Z$ = Z$ + HR$ + P1$ + P2$ + VP$ + PW$ + P3$ + T1$ + T2$
180  IF  VAL (CO$) = 0 THEN Z$ = Z$ + "EOF": GOTO 210
200  Z$ = Z$ + CO$ + SV$ + SR$ + PR$ + "EOF"
210  IF VP$ = "-99" THEN VP$ = "---"
220  IF PW$ = "-99" THEN PW$ = "---"
270  PRINT TP$;"  ";HR$;" ";P1$;"/";P2$;" ";P3$;" ";VP$;"
     ";SR$;
272  PRINT  SPC( 3);CO$;" ";SV$;" ";RP$; SPC( 3);P4$;"/";P5$;"
     ";P6$;" ";PW$;" ";PR$;
274  PRINT  SPC( 2);T1$;" ";T2$;
280  PRINT D$: PRINT D$;"PR#0"
```

```
285  IF Z$ = "" THEN 893
286  REM *****************************
287  REM * WRITE DATA STRING TO DISC *
288  REM *****************************
290  X$ = NA$ + " CVS DATALEN"
296  FI = FI + 1:Z$(FI) = Z$
298  PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$
299  PRINT FI: PRINT D$;"CLOSE";X$
302  X$ = NA$ + " CVS DATA"
310  PRINT D$;"OPEN";X$
316  PRINT D$;"POSITION";X$;",R";FI - 1
320  PRINT D$;"WRITE";X$
330  PRINT Z$
340  PRINT D$;"CLOSE";X$: IF FI < 150 THEN 345:
     REM * WRITE 50 DATA STRINGS TO SEPARATE FILE *
341  GOSUB 929: REM * TEXT MODE *
342  GOSUB 400: REM * WRITE VARS TO SHUNTFILE *
343  HOME : VTAB 10:Z$ = "WRITING 50 FILES TO DISC": GOSUB 7:
     PRINT D$"RUN FILEWRITE"
344  REM * IF SHUNT DATA READY, SAVE VARIABLES AND RUN
     ROCHESHUNT *
345  IF SH$ = "Y" THEN  GOSUB 400:SH$ = "R"
350  GOSUB 360: GOTO 370
360  PW$ = "-99":VP$ = PW$:SV$ = "---":CO$ = "----":SR$ =
     CO$:PR$ = CO$: RETURN
364  REM *******************************************************
365  REM * IF 'MENU$' SET TO 'N',SWITCH FROM HGR MODE TO TEXT *
366  REM *******************************************************
370  IF ME$ = "N" THEN  HOME : CALL 1002: PRINT D$;"PR#0":
     PRINT D$;"IN#0": TEXT : GOSUB 13000
380  GOTO 890
385  REM ****************************
390  REM * WRITE DATA TO SHUNT FILE *
395  REM ****************************
400  X$ = "SHUNTFILE": PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$
405  PRINT DR: PRINT PR: PRINT CV$: PRINT FLAG$: PRINT PT$:
     PRINT SERVO$
410  PRINT TP$: PRINT HR$: PRINT P1$: PRINT P2$: PRINT P3$:
```

```
    PRINT VP$: PRINT SR$:
420 PRINT CO$: PRINT SV$: PRINT RP$: PRINT P4$: PRINT P5$:
    PRINT P6$: PRINT PW$: PRINT PR$
430 PRINT T1$: PRINT T2$: PRINT Z$
440 PRINT D$;"CLOSE"
450 X$ = "NEXTPROG": PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$
460 PRINT "GSROCHENS"
470 PRINT D$;"CLOSE": RETURN
799 'REM ******************
890 REM * TIMING ROUTINE *
891 REM ******************
891 IF VENT = 0 THEN VENT = S
892 S1 = S: HOME
893 GOSUB 24000:S2 = S:
894 IF S2 - S1 > DR * 60 THEN 60
895 IF SE$ = "Y" AND VENT + 60 * 60 <  = S2 THEN  GOSUB 400:
    PRINT D$;"RUN ROCHE/SERVO"
897 IF MENU$ = "N" THEN  GOSUB 14900
898 IF MENU$ = "Y" THEN  GOSUB 20000
899 REM *************************
900 REM * CHECK DAMPING/WEDGING *
901 REM *************************
902 I(1) = 910:I(2) = 916:I(3) = 913: IF PT$ <  > "SYSCHAN1"
    THEN  FOR I = 1 TO 3:I(I) = I(I) + 10: NEXT
904 CALL 768:D$ =  CHR$ (4): PRINT D$: IF  PEEK (880) = 1
    THEN  HOME : PRINT  CHR$ (7): FLASH : VTAB 12:Z$ = "?
    ROCHE DISCONNECTED": GOSUB 7: NORMAL : GOTO 904
906 PFLAG$ = "Y":I = I(1): GOSUB 3000:P1 =  VAL (R$)
908 I = I(2): GOSUB 3000:P2 =  VAL (R$):I = I(3): GOSUB
    3000:P3 =  VAL (R$)
920 IF P3 = 0 THEN  VTAB 23: GOTO 893
925 IF (P1 - P2) > .2 * P3 THEN 942
928 GOSUB 929: GOTO 930
929 CALL 1002: PRINT D$;"PR#0": PRINT D$;"IN#0": TEXT :MENU$
    = "Y": RETURN
930 FOR I = 1 TO 10: PRINT "": NEXT
932 HOME : FLASH : VTAB 6: HTAB 10: PRINT "SYSTOLIC LINE
    DAMPED": VTAB 16: HTAB 9: PRINT "CHECK FOR BLOOD IN LINE"
```

```
935  VTAB 18: HTAB 5: PRINT "CHECK PRESSURE IN PERFUSION BAG"
940  VTAB 24: HTAB 8: NORMAL : PRINT "TYPE SPACEBAR TO RESET
     ";: GET A$: IF A$ <  > " " THEN 940
942  IF CVP$ = "Y" THEN  VTAB 23: GOTO 893
944  I(4) = 920:I(5) = 926:I(6) = 923: IF PT$ <  > "SYSCHAN1"
     THEN  FOR I = 4 TO 6:I(I) = I(I) - 10: NEXT
950  I = I(4): GOSUB 3000:P4 =  VAL (R$):I = I(5): GOSUB
     3000:P5 =  VAL (R$):I = I(6): GOSUB 3000:P6 =  VAL (R$)
955  IF P3 = 0 THEN  VTAB 23: GOTO 893
960  IF (P4 - P5) > .2 * P6 THEN  VTAB 23: GOTO 893
961  GOSUB 929
962  FOR I = 1 TO 10: PRINT "": NEXT
963  HOME : FLASH : VTAB 6: HTAB 4: PRINT "PULMONARY ARTERIAL
     LINE DAMPED": VTAB 8: HTAB 7: PRINT "CHECK FOR BLOOD IN
     LINE"
964  VTAB 14: HTAB 4: PRINT "******************************"
965  VTAB 15: HTAB 4: PRINT "CHECK FOR WEDGING OF CATHETER"
966  VTAB 16: HTAB 4: PRINT "******************************"
970  VTAB 23: HTAB 5: NORMAL : PRINT "TYPE SPACEBAR TO RESET
     ";: GET A$: IF A$ <  > " " THEN 970
972  FOR VV = 14 TO 16: GOSUB 975: NEXT : VTAB 23: GOTO 893
975  VTAB VV: HTAB 1: FOR II = 1 TO 39: PRINT " ";: NEXT :
     RETURN
2000  REM * STRING LENGTH ADJUST *
2010  IF  LEN (QQ$) < QQ THEN QQ$ = " " + QQ$: GOTO 2010
2020  IF  LEN (QQ$) > QQ THEN QQ$ =  LEFT$ (QQ$,QQ)
2030  RETURN
3000  REM *************************
3002  REM * OBTAIN DATA FROM DISC *
3005  REM *************************
3010  R$ =  CHR$ ( PEEK (I)) +  CHR$ ( PEEK (I + 1)) +  CHR$ (
      PEEK (I + 2)): IF PFLAG$ = "Y" THEN R$ =  STR$ ( INT ((
      VAL (R$) - 128) / 2))
3030  IF R$ = "-64" THEN R$ = "---"
3040  QQ$ = R$:QQ = 3: GOSUB 2000:R$ = QQ$
3050  RETURN
```

```
9997  REM ****************************************
9998  REM * SHUNT AND CARDIAC OUTPUT SUB ROUTINE *
9999  REM ****************************************
10000  HOME
10010  IF SH$ = "R" THEN  HOME : VTAB 20: PRINT "LOAD SHUNT
       PROGRAM Y/N ";: GET A$: IF A$ = "Y" THEN  POP : PRINT
       A$: VTAB 20: HTAB 1: PRINT "   LOADING SHUNT PROGRAM   ":
       PRINT  D$;"RUN ROCHESHUNT"
10015  IF A$ = "" THEN  POP : GOTO 20152
10020  IF SH$ = "R" THEN  POP : GOTO 20152
10380  HOME : VTAB 4: HTAB 1: INPUT "CARDIAC OUTPUT L/MIN =
       ";CO$
10385  IF CO$ = "" THEN  POP : GOTO 20152
10390  IF  VAL (CO$) < 1.5 OR  VAL (CO$) > 20 THEN 10380
10420  RETURN
12985  REM **********************************
12990  REM *          TREND GRAPHICS         *
12992  REM *          --------------         *
12994  REM * SELECT FILE RANGE FOR PLOTTING *
12995  REM **********************************
12996  REM
13000  NI = FI:EX = 0:NH = 0:NM = 0:ND = 0: HOME :F2 = NI:PI =
       1: GOTO 13024

13001  HOME : PRINT "HOW FAR BACK DO YOU WISH TO START PLOT?":
       PRINT : PRINT "TYPE THE ELAPSED TIME": GOSUB 13003:NM =
       Z: GOTO 13012: REM * EXPANDED PLOT *
13003  VTAB 5: PRINT "FOLLOWED BY THE UNITS AND <RETURN>":
       VTAB 8: PRINT "EG  10M=10MINS  5H=5HOURS  2D=2DAYS":
       VTAB 12: HTAB 19: INPUT TI$: IF TI$ = "M" THEN  POP :
       RETURN
13004  HOME : IF  RIGHT$ (TI$,1) = "D" THEN Z =  VAL ( LEFT$
       (TI$, LEN (TI$) - 1)) * 1440: RETURN
13006  IF  RIGHT$ (TI$,1) = "H" THEN Z =  VAL ( LEFT$ (TI$,
       LEN (TI$) - 1)) * 60: RETURN
13008  IF  RIGHT$ (TI$,1) = "M" THEN Z =  VAL ( LEFT$ (TI$,
       LEN (TI$) - 1)): RETURN
13010  POP : GOTO 13001
```

```
13012  HOME : PRINT "TYPE TIME INTERVAL TO BE PLOTTED": GOSUB
       13003:Z9 = Z
13013  REM * START OF PLOT *
13015  START =  FN TIME(FI) - NM
13017  FOR I = 1 TO FILE: IF START >  FN TIME(I) THEN  NEXT
13019  START =  FN TIME(I):PI = I
13021  LAST = ST + Z9: FOR I = FI TO 1 STEP  - 1: IF LAST <
       FN TIME(I) THEN  NEXT
13022  F2 = I
13023  LAST =  FN TIME(F2): GOTO 13025: REM * END OF PLOT *
13024  START =  FN TIME(PI):LAST =  FN TIME(FILE): HOME
13025  NM = LAST - START: REM * NO OF MINS TO BE PLOTTED *
13027  ET = ST: GOSUB 25000: GOSUB 26000:SD = DY:MS = MT:SH =
       H:SM = M
13028  ET = LA: GOSUB 25000: GOSUB 26000:LM$ =  STR$ (MT): IF
       LEN (LM$) < 2 THEN LM$ = "0" + LM$
13058  IF NM < 2 THEN  HOME : PRINT " NOT ENOUGH VALUES TO
       PLOT": FOR II = 0 TO 1000: NEXT : RETURN
13234  IF MINS = 9999 THEN BA = 30 + 212 / EXTRA
13240  REM ****************
13250  REM * HI-RES PLOTS *
13255  REM ****************
13260  GOSUB 13420
13262  GOTO 14700
13280  AX = 0:CX = 0:PP = PI
13281  IF BX = 0 THEN BA = 30: GOTO 13290
13282  BA = 30 + 212 / NM * (BX + CX): REM * BASE ADDRESS *
13290  X1 = BA
13300  Y1 = 190 - Z3 *  FN PLT(PP) + Z2
13305  IF PP = FI OR PP = F2 THEN  RETURN
13307  REM * IF VALUE UNACCEPTABLE ADD TIME TO AX *
13310  IF  FN PLT(PP) < LO THEN PP = PP + 1:CX = CX + ( FN
       TIME(PP) -  FN TIME(PP - 1)): GOTO 13282
13311  REM * PLOT RANGE CHECK *
13312  IF Y1 < 0 THEN Y1 = 0
13314  IF X1 > 276 THEN X1 = 276
13316  IF X1 < 30 THEN X1 = 30
13320  FOR X = PP + 1 TO F2:Y2 = 190 - Z3 *  FN PLT(X) + Z2
```

```
13330  IF  FN TIME(X) <  =  FN TIME(X - 1) THEN X = X + 1:
       GOTO 13410
13349  X2 = X1 + 212 / NM * ( FN TIME(X) -  FN TIME(X - 1)) +
       AX
13350  REM * IF VALUE UNACCEPTABLE ADD TIME TO AX *
13370  IF  FN PLT(X) < LO THEN AX = AX + 212 / NM * ( FN
       TIME(X) -  FN TIME(X - 1)): GOTO 13410
13374  REM * RANGE CHECK *
13375  IF X2 > 276 THEN X2 = 276
13376  IF Y2 < 0 THEN Y2 = 0
13377  IF X2 < 30 THEN X2 = 30
13378  IF Y2 > 189 THEN Y2 = 189
13379  AX = 0
13380  FOR II = 0 TO LP: HPLOT X1 + II,Y1 + II TO X2 + II,Y2 +
       II: NEXT
13390  HPLOT X1 + 1,Y1 + 1 TO X2 + 1,Y2 + 1:REM DOUBLE PLOT
       LINE *
13400  X1 = X2:Y1 = Y2
13410  NEXT X: RETURN
13420  GOTO 13430: REM * AXES PLOT *
13421  REM ************************************
13422  REM * ENSURE INTEGER LABELS FOR X-AXIS *
13423  REM ************************************
13424  FOR I = 2 TO 7: IF N / I -  INT (N / I) < .01 THEN NX =
       2 * I: RETURN
13425  NEXT I:N = N + 1: GOTO 13422
13430  IF NM < 61 THEN NT$ = "MINS":N = NM: GOSUB 13422:BX = N
       - NM:NM = N: GOTO 13505
13450  IF NM < 2160 THEN BX = SM: IF NM + BX < 2160 THEN NT$ =
       "TIME": GOTO 13498: REM * IF > 36HRS, PLOT DATE *
13470  NT$ = "DATE":BX = SH * 60 + SM
13490  ND =  INT ((NM + BX) / 1440) + 1
13492  N = ND: GOSUB 13422:ND = N:NM = ND * 1440
13493  NT$ = "DATE": GOTO 13505
13498  NH = (NM + BX) / 60: IF (NM + BX) / 60 -  INT ((NM +
       BX) / 60) <  > 0 THEN NH =  INT ((NM + BX) / 60) + 1
13500  N = NH: GOSUB 13422:NH = N:NM = NH * 60
```

```
13501  REM **************************
13502  REM * INVOKE HI-RES GRAPHICS *
13503  REM **************************
13505  HGR2 : HCOLOR= 3: POKE 54,0: POKE 55,96
13506  FOR V = 0 TO NX:X = 30 + V * 212 / NX: HPLOT X,174 TO
       X,177: HPLOT X + 1,174 TO X + 1,177: NEXT
13510  FOR V = 0 TO NX STEP 2:X = 30 + V * 212 / NX: HPLOT
       X,174 TO X,180: HPLOT X + 1,174 TO X + 1,180: NEXT
13520  HPLOT 30,10 TO 30,180: HPLOT 31,10 TO 31,180
13521  REM ****************
13522  REM * LABEL X-AXIS *
13523  REM ****************
13525  IF NT$ = "MINS" THEN LX = SM:NZ = NM: FOR V = 0 TO NX
       STEP 2: VTAB 24:X = V * 212 / NX:A$ =  STR$ ( INT (LX +
       V * NZ / NX + .5)):HT =  INT (X / 6.975 + 5.5 -  LEN (A$)
       / 2): GOTO 13542
13530  IF NT$ = "TIME" THEN LX = SH:BX = SM:NZ = NH: GOTO
       13540
13532  LX = SD:BX = SH * 60 + SM:NZ = ND
13540  FOR V = 0 TO NX STEP 2: VTAB 24:X = V * 212 / NX:A$ =
       STR$ ( INT (LX + V * NZ / NX + .5)):HT =  INT (X /
       6.975 + 5.5  -  LEN (A$) / 2)
13542  A =  VAL (A$)
13544  IF NT$ = "DATE" THEN 13547
13545  IF NT$ = "TIME" AND A > 24 THEN A = A - 24: GOTO 13545
13546  IF NT$ = "MINS" AND A > 60 THEN A = A - 60
13547  GOTO 13553
13548  RESTORE : READ J:II = 0:REM * READ DAYS IN MONTHS *
13549  II = II + 1: READ J: IF II < MS THEN 13549
13550  IF A > J THEN A = A - J
13551  IF A > J THEN 13549
13553  HT = HT + 1
13554  HTAB HT: PRINT A: NEXT
13555  VTAB 23: HTAB 37: PRINT NT$;: IF NT$ = "DATE" THEN
       VTAB 24: HTAB 38: PRINT "/";LM$;
13600  RETURN
```

```
14690  REM *****************************
14700  REM * CARDIOVASCULAR TREND PLOT *
14705  REM *****************************
14710  JJ = 160: VTAB 2: PRINT JJ: VTAB 7: PRINT JJ - 40: VTAB
       12: HTAB 2: PRINT JJ - 80: VTAB 17: HTAB 2: PRINT JJ -
       120
14720  HPLOT 30,50 TO 242,50: HPLOT 30,90 TO 242,90
14730  HCOLOR= 5
14735  VTAB 6: HTAB 37: PRINT "BP": HPLOT 252,49 TO 268,49 TO
       268,48 TO 252,48
```

```
14739  REM ***********************************************
14740  REM * Z0 = START POSITION OF VALUE TO BE PLOTTED *
14742  REM * Z1 = NO OF CHARACTERS IN DATA STRING       *
14744  REM * Z2 = OFFSET FOR Y-AXIS                     *
14744  REM * Z3 = SCALINGF FACTOR FOR Y-AXIS            *
14746  REM * LO = LOWEST ACCEPTABLE VALUE FOR PLOTTING  *
14748  REM * LP = THICKNESS OF LINE USED FOR PLOTTING   *
14748  REM ***********************************************
14750  Z0 = 10:Z1 = 3:Z2 =  - 20:Z3 = 1:LO = 40:LP = 1: GOSUB
       13280
14770  Z0 = 13:LO = 20: GOSUB 13280: REM  DIA
14790  REM * HR *
14800  HCOLOR= 6
14805  VTAB 10: HTAB 37: PRINT "HR": FOR II = 1 TO 3: HPLOT
       253,80 + II TO 272,80 + II: NEXT
14810  Z0 = 7:LP = 2: GOSUB 13280
14840  REM * CVP *
14850  HCOLOR= 1
14855  HPLOT 30,160 TO 242,160
14857  VTAB 14: HTAB 37: PRINT "CVP": FOR II = 0 TO 2: HPLOT
       252,116 + II TO 273,116 + II: NEXT
14858  VTAB 19: HTAB 3: PRINT "+5": VTAB 22: HTAB 3: PRINT
       "-5"
14860  Z0 = 16:Z1 = 3:Z2 =  - 30:Z3 = 3:LO =  - 10: GOSUB
13280
14865  HCOLOR= 2: VTAB 18: HTAB 37: PRINT "PAWP": HPLOT
       252,146  TO 278,146 TO 278,147 TO 252,147
14870  Z0 = 19:LP = 1: GOSUB 13280: REM * PAWP *
14890  VTAB 1: HTAB 1::Z$ = "M(ENU  X(PAND  P(RINT": PRINT Z$
14895  VTAB 2: HTAB 32: PRINT  MID$ (T$,7,2);"."; MID$
       (T$,10,2);":"; MID$ (T$,13,2)
14900  FL$ = "Y":A = 0: GOSUB 21001
14902  POKE 54,0: POKE 55,96: CALL 1002:REM *RE-SET DOS
       POINTERS TO PRINT TIME ON GRAPHICS PAGE *
14904  FL$ = "Y"
14905  VTAB 1: HTAB 32: PRINT  MID$ (T$,7,2);"."; MID$
       (T$,10,2);":"; MID$ (T$,13,2)
14920  IF A = 205 THEN ME$ = "Y": GOTO 15030: REM * RETURN TO
```

```
      TEXT MENU *
14990 REM * PRINT TREND GRAPH ON SILENTYPE *
15000 IF A = 208 THEN  PRINT D$;"PR#1": PRINT : PRINT : PRINT
      : PRINT  CHR$ (17): PRINT D$;"PR#0": GOTO 15020
15010 IF A = 216 THEN 15040: REM * EXPAND GRAPH *
15020 A = 0:ME$ = "N": RETURN
15030 CALL 1002: PRINT D$;"PR#0": PRINT D$;"IN#0": TEXT :
      RETURN
15040 CALL 1002: PRINT D$;"PR#0": PRINT D$;"IN#0": TEXT :
      GOTO 13001
19997 REM *********************
19998 REM * DISPLAY TEXT MENU *
19999 REM *********************
20000 VTAB 1: HTAB 1:A = 0
20002 PRINT NA$;" / ";NO$
20005 VTAB 1: HTAB 32: PRINT  MID$ (T$,7,2);"."; MID$
      (T$,10,2);":";; MID$ (T$,13,2)
20007 V = 3
20010 VTAB V + 1: PRINT "1. READ PUL ART WEDGE PRES.      "
20020 VTAB V + 3: PRINT "2. ALTER PRESSURE CHANNELS        "
20030 VTAB V + 5: HTAB 1: PRINT "3. TYPE TEXT LABEL          "
20035 VTAB V + 7: HTAB 1: PRINT "4. ENTER CARDIAC OUTPUT"
20040 VTAB V + 9: IF SH$ = "R" THEN  PRINT "5. ENTER SHUNT
      DATA": GOTO 20050
20045 PRINT "5. COLLECT CARDIAC AND SHUNT DATA"
20050 VTAB V + 11: PRINT "6. TREND DISPLAY"
20060 VTAB V + 13: PRINT "7. ALTER DATA RETRIEVAL TIME
      (=";DR;" MIN)"
20070 VTAB V + 15: PRINT "8. TAKE NEW READING / PRINT 24HR
      REPORT
20080 VTAB V + 17: HTAB 1: PRINT "9. MEASURE VENTILATION /
      QUIT"
20085 VTAB 23: HTAB 21: PRINT "* CH1=";: IF PT$ = "SYSCHAN1"
      THEN  PRINT "SYS": GOTO 20090
20086 IF CV$ = "Y" THEN  PRINT "CVP": GOTO 20088
20087 PRINT "PAP"
20088 VTAB 23: HTAB 31: PRINT "CH2=SYS *":: GOTO 20100
20090 VTAB 23: HTAB 31: PRINT "CH2=";: IF CV$ = "Y" THEN
```

```
       PRINT "CVP *": GOTO 20100
20091  PRINT "PAP *"
20100  VTAB V + 20: HTAB 1: PRINT "TYPE NO. REQUIRED  ": GOSUB
       21100
20105  IF A = 177 THEN  GOSUB 20158: GOTO 20153: REM * READ
       PAWP AND CVP *
20110  IF A = 178 THEN 22000: REM * ALTER PRESSURE CHANNELS *
20112  IF A <  > 179 THEN 20128
20114  HOME :Z$ = "TYPE TEXT LABEL": GOSUB 7
20115  VTAB 5: PRINT "TYPE IN THE LABEL YOU WISH TO PRINT OUT
       FOLLOWED BY THE <RETURN> KEY.
20116  PRINT : PRINT : PRINT "USE THE LEFT ARROW KEY '<-' TO
       CORRECT  ANY ERRORS.
20117  PRINT : PRINT :Z$ = "TO RETURN TO MENU, TYPE <RETURN>
       KEY": GOSUB 7
20118  INPUT Z$: IF Z$ = "" THEN A = 0: HOME : RETURN
20119  GOSUB 24000:TP$ =  MID$ (T$,7,2) + "." +  MID$
       (T$,10,2)
20120  REM * PRINT AND STORE TEXT STRING *
20122  POKE 216,0: PRINT D$;"PR#";PR: PRINT : PRINT TP$; SPC(
       3);Z$:A = 0: PRINT D$;"PR#"0: HOME
20123  ZZ$ =  STR$ ( INT (S / 60)): FOR II = 1 TO 6: IF  LEN
       (ZZ$) < 6 THEN ZZ$ = "0" + ZZ$: NEXT
20124  Z$ = ZZ$ + Z$
20125  X$ = NA$ + " TEXTLEN":LT = LT + 1: PRINT D$;"OPEN";X$:
       PRINT D$;"WRITE";X$: PRINT LT: PRINT D$;"CLOSE"
20126  X$ = NA$ + " TEXT": PRINT D$;"OPEN";X$: PRINT
       D$;"POSITION";X$;",R";LT - 1
20127  PRINT D$;"WRITE";X$: PRINT Z$: PRINT D$;"CLOSE":Z$ =
       "": RETURN
20128  IF A = 180 THEN  GOSUB 10380: GOSUB 20158: GOTO 20153:
       REM * OBTAIN CO/PAWP/CVP *
20130  IF A = 181 AND SH$ = "R" THEN : GOSUB 10000: GOSUB
       20158: GOTO 20153: REM * OBTAIN CO/PAWP/CVP AND SET
       SHUNT$ TO 'Y' *
20135  IF A = 181 THEN SH$ = "Y": GOSUB 10380: GOSUB 20158:
       GOTO 20153
20136  IF A <  > 182 THEN 20142
```

```
20137  HOME :Z$ = "TREND PLOT": GOSUB 7: VTAB 8: PRINT "1.
       COMPLETE PLOT": VTAB 12: PRINT "2. EXPANDED PLOT": VTAB
       16: PRINT "TYPE <RETURN> KEY IF NEITHER"
20138  VTAB 20: GET A$: IF A$ =  CHR$ (13) THEN  RETURN
20139  IF A$ = "1" THEN  GOSUB 13000: RETURN
20140  IF A$ = "2" THEN  GOSUB 13001: RETURN
20141  GOTO 20138
20142  IF A = 183 THEN  HOME : VTAB 10: PRINT "TYPE TIME
       REQUIRED BETWEEN MEASUREMENTS": PRINT : PRINT "IN
       MINUTES. THEN TYPE <RETURN> KEY": VTAB 20: HTAB 19: INPUT
       DR$: IF  VAL (DR$) < 1 OR  VAL (DR$) > 120 THEN 20142
20143  IF A = 183 THEN DR =  VAL (DR$): GOTO 20153
20145  IF A = 184 THEN 22200: REM * PRINT 24HR REPORT /
       COLLECT NEW DATA *
20146  IF A <  > 185 THEN 20152
20147  HOME : VTAB 4: PRINT "1. MEASURE VENTILATION": VTAB 8:
       PRINT "2. QUIT PROGRAM": VTAB 12: PRINT "   TYPE
       <RETURN> IF NEITHER
20148  VTAB 20: GET A$: IF A$ =  CHR$ (13) THEN  RETURN
20149  IF A$ = "1" THEN SE$ = "Y": PRINT A$: GOSUB 400: PRINT
       D$;"RUN ROCHESERVO
20150  IF A$ = "2" THEN  HOME :Z$ = "OK FINISHED": VTAB 5:
       GOSUB 7:Z$ = "TO RESTART TYPE THE WORD 'RUN'": VTAB 8:
       GOSUB 7:Z$ = "THEN PRESS THE <RETURN> KEY": VTAB 11:
       GOSUB 7:END
20151  GOTO 20148
20152  A = 0: RETURN
20153  POP : GOTO 60
20155  REM ********
20156  REM * PAWP *
20157  REM ********
20158  HOME : VTAB 4: IF CVP$ = "Y" THEN Z$ = "RECORDING CVP
       AT PRESENT": GOSUB 7: FOR I = 1 TO 2000: POP : GOTO
20152
20159  HTAB 1: PRINT "INFLATE PULMONARY ARTERY BALLOON"
20160  CALL 768:PFLAG$ = "Y": IF PT$ = "SYSCHAN1" THEN I =
       926: GOTO 20164
20162  I = 916
```

```
20164  GOSUB 3000
20165  VTAB 16: HTAB 1: PRINT "PRESSURE OBTAINED = ";: HTAB
       24: PRINT " MM HG"
20170  VTAB 16: HTAB 21: PRINT R$;
20180  VTAB 23: HTAB 1: PRINT "TO RECORD VALUE TYPE
       <SPACEBAR>"
20190  A =  PEEK ( - 16384): POKE 16368,0: IF A = 141 THEN
       POP: GOTO 20152
20192  IF A <  > 160 THEN 20160
20195  QQ$ = R$:QQ = 3: GOSUB 2000:PW$ = QQ$
20199  D$ =  CHR$ (4): PRINT D$
20200  REM ***********
20210  REM * GET CVP *
20220  REM ***********
20230  HOME : VTAB 5:Z$ = "TYPE IN CVP IN CM H20 AT
       MID-AXILLARY": GOSUB 7: PRINT :Z$ = "LINE THEN TYPE
       <RETURN> KEY.": GOSUB 7
20232  VTAB 12:Z$ = "IF CVP UNAVAILABLE, PRESS <RETURN>":
       GOSUB 7: PRINT : PRINT : INPUT VP$
20235  IF VP$ = "" THEN QQ$ =  STR$ ( - 99): GOTO 20270
20237  CV =  VAL (VP$)
20240  IF CV <  - 10 OR CV > 30 THEN 20230
20257  CV = CV / 1.36
20260  QQ$ =  STR$ (CV)
20270  QQ = 3: GOSUB 2000:VP$ = QQ$
20300  RETURN
21001  REM ******************
21005  REM * TEST KEY PRESS *
21007  REM ******************
21100  A =  PEEK ( - 16384): POKE  - 16368,0: RETURN
22000  HOME : VTAB 1:Z$ = "ALTER PRESSURE CHANNELS": GOSUB 7
22010  VTAB 8: PRINT "1. EXCHANGE PUL ART AND CVP CHANNELS
22020  VTAB 12: PRINT "2. MOVE SYSTOLIC PRES TO OTHER CHANNEL"
22025  VTAB 16: PRINT "TYPE <RETURN> KEY IF NEITHER"
22030  VTAB 20: HTAB 1: GET A$: IF A$ = "1" THEN  HOME : GOTO
       22110
22040  IF A$ = "2" THEN  HOME : GOTO 22115
22045  IF A$ =  CHR$ (13) THEN  HOME : GOTO 20152
```

```
22050  GOTO 22030
22110  IF CVP$ = "N" THEN CVP$ = "Y": GOTO 20153
22112  CVP$ = "N": GOTO 20153
22115  IF PT$ = "SYSCHAN1" THEN PT$ = "PULCHAN1": GOTO 20153
22116  PT$ = "SYSCHAN            1": GOTO 20153
22200  HOME : VTAB 8: PRINT "1. TAKE NEW READING": VTAB 10:
       PRINT "2. PRINT 24HR CVS/RESP REPORT": VTAB 12: PRINT "3.
       PRINT 24HR TEXT REPORT": VTAB 18: PRINT "TYPE <RETURN>
       IF NONE OF THESE"
22210  VTAB 22: GET A$: PRINT A$: IF A$ =  CHR$ (13) THEN
       RETURN
22220  IF A$ = "1" THEN 20153
22230  IF A$ = "2" THEN  GOSUB 400: PRINT D$;"RUN ROCHEPRINT"
22240  IF A$ = "3" THEN  GOSUB 400: PRINT D$;"RUN
       ROCHEPRINTEXT"
24000  REM ************
24005  REM * GET TIME *
24007  REM ************
24010  D$ =  CHR$ (4): PRINT D$:SLOT = 3
24030  PRINT D$;"IN#";SL: PRINT D$;"PR#";SL: INPUT T$: PRINT
       D$;"IN#0": PRINT D$;"PR#0"
24040  MT =  VAL ( MID$ (T$,1,2))
24050  D =  VAL ( MID$ (T$,4,2))
24060  H =  VAL ( MID$ (T$,7,2))
24070  M =  VAL ( MID$ (T$,10,2))
24080  S =  VAL ( MID$ (T$,13,6))
24090  RESTORE
24100  DTD = 0
24110  FOR I = 1 TO MT: READ J:DTD = DTD + J: NEXT
24120  DATA  0,31,28,31,30,31,30,31,31,30,31,30,31
24130  DTD = DTD + D
24140  IF MT > 2 AND L = 1 THEN DTD = DTD + 1
24150  STD = DTD * 86400 + H * 3600 + M * 60 + S
24160  S = STD
24170  RETURN
```

```
25000 REM ******************
25020 REM * MINS->DAYS/HRS *
25030 REM ******************
25040 DY =  INT (ET / 1440)
25050 ET = ET - DY * 1440
25060 H =  INT (ET / 60)
25070 M = ET - H * 60
25080 RETURN
26000 REM **************
26005 REM * DAYS->DATE *
26007 REM **************
26010 IF L = 1 AND DY = 60 THEN MT = 2:DY = 29: GOTO 26070
26015 IF L = 1 AND DY > 60 THEN DY = DY - 1
26017 RESTORE
26018 READ J
26020 FOR II = 1 TO 12: READ J:MT = MT + 1
26030 IF DY - J < 1 THEN ND = DY: GOTO 26070
26035 DY = DY - J
26040 NEXT
26070 RETURN
```

```
4   REM *************
5   REM * FILEWRITE *
6   REM *************
10  Z$ = "FILEPROG":D$ =  CHR$ (4)
14  REM ******************************
15  REM * CHECK DISC SPACE AVAILABLE *
16  REM ******************************
20  ONERR  GOTO 40
30  PRINT D$;"OPEN";Z$: PRINT D$;"READ";Z$: INPUT N$: PRINT
    D$;"CLOSE": GOTO 60: REM * N$='OK' IF SPACE AVAILABLE *
40  POKE 216,0: PRINT D$;"OPEN";Z$: PRINT D$;"WRITE";Z$: PRINT
    "FILEWRITE": PRINT D$;"CLOSE"
50  PRINT D$;"RUN FILESPACE":REM * RUN DISC SPACE CHECK *
60  POKE 216,0: PRINT D$;"DELETE";Z$
70  GOTO 100
80  HTAB (40 -  LEN (Z$)) / 2: PRINT Z$
90  RETURN
100  D$ =  CHR$ (13) +  CHR$ (4)
110  PRINT D$;"  MONI,C,O"
120  Y = 50: REM * NO OF FILES WRITTEN *
130  DIM Z$(270)
140  HOME :Z$ = "WRITING " +  STR$ (Y) + " FILES TO DISC":
     GOSUB 80
144  REM ********************
145  REM * OBTAIN DATA FILES *
146  REM ********************
150  PRINT D$;"OPEN NAMEFILE": PRINT D$;"READ NAMEFILE": INPUT
     NA$: PRINT D$;"CLOSE"
160  Z$ = NA$ + " CVS DATALEN"
170  PRINT D$;"OPEN";Z$: PRINT D$;"READ";Z$: INPUT FI: PRINT
     D$;"CLOSE"
180  Z$ = NA$ + " CVS DATA"
190  PRINT D$;"OPEN";Z$: PRINT D$;"READ";Z$: FOR I = 1 TO FI:
     INPUT Z$(I): NEXT : PRINT D$;"CLOSE"
195  IF N$ = "OK" THEN 200
196  HOME : FLASH : PRINT  CHR$ (7): VTAB 10:Z$ = "DISC IS
     FULL": GOSUB 80: PRINT :Z$ = "PLEASE PUT IN A NEW DISC":
     GOSUB 80: PRINT: Z$ = "AND TYPE <RETURN>": GOSUB80
```

```
197  FOR I = 1 TO 3000: NEXT : IF  PEEK ( - 16384) = 13 + 128
     THEN 200
198  POKE  - 16368,0: GOTO 196
200  NO = Y: NORMAL
210  Z$ = NA$ + " CVS " +  STR$ (NO)
220  ONERR  GOTO 260
224  REM *********************************
225  REM * FIND LAST NO OF FILES WRITTEN *
226  REM *********************************
230  PRINT D$;"OPEN";Z$
240  PRINT D$;"READ";Z$: INPUT Z$(1): PRINT D$;"CLOSE";Z$
250  POKE 216,0:NO = NO + Y: GOTO 210
260  POKE 216,0
264  REM ********************************
265  REM * LAST NO OF FILES WRITTEN = Y *
266  REM ********************************
270  PRINT D$;"OPEN";Z$: PRINT D$;"WRITE";Z$
280  FOR I = 1 TO Y: PRINT Z$(I): NEXT
290  PRINT D$;"CLOSE"
300  Z$ = NA$ + " CVS DATALEN"
310  PRINT D$;"OPEN";Z$: PRINT D$;"WRITE";Z$: PRINT FI - Y:
     PRINT D$;"CLOSE"
320  Z$ = NA$ + " CVS DATA"
330  PRINT D$;"DELETE";Z$
340  PRINT D$;"OPEN";Z$: PRINT D$;"WRITE";Z$
350  FOR I = Y + 1 TO FI: PRINT Z$(I): NEXT : PRINT D$;"CLOSE"
360  X$ = "NEXTPROG"
370  PRINT D$;"OPEN";X$: PRINT D$;"READ";X$
380  INPUT N$: PRINT D$;"CLOSE"
390  PRINT D$;"RUN";N$
```

```
    REM *********************************************
    REM * FILESPACE - MODIFIED FROM COMPUTEC UTILITY *
    REM *********************************************
0   REM  SPACE (C) SGL 22/08/79
10  D$ =  CHR$ (13) +  CHR$ (4)
100  SP =  PEEK (106) * 256 +  PEEK (105) - 128
110  CALL SP
120  X =  PEEK (513)
130  IF X = 255 THEN 200
140  X = X * 256 +  PEEK (512)
145  HOME : VTAB 10: PRINT "SPACE FREE ON DISC = ";
150  PRINT X;" SECTORS ("; INT (X / 4.96);"%)";
155  FOR I = 1 TO 1000: NEXT
160  IF X > 28 THEN N$ = "OK": GOTO 196
170  N$ = "NO"
196  X$ = "FILEPROG"
197  PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$: PRINT N$: PRINT
     D$;"CLOSE"
198  PRINT D$;"RUN FILEWRITE"
200  PRINT  CHR$ (7);"I/O ERROR "; PEEK (512);" READING VTOC";
```

```
2    REM **************
3    REM * ROCHESHUNT *
4    REM **************
5    TEXT :D$ =  CHR$ (4): PRINT D$;"NOMONI,C,O"
6    PRINT D$;"NOMONI,C,O": GOSUB 10: GOTO 35
10   HOME :X$ = "GLASGOW ROYAL INFIRMARY": GOSUB 30
20   X$ = "RESPIRATORY INTENSIVE CARE UNIT": GOSUB 30
25   X$ = "ROCHE SHUNT PROGRAM": GOSUB 30: RETURN
30   PRINT : HTAB (40 -  LEN (X$)) / 2: PRINT X$: RETURN
35   GOSUB 600
36   REM ******************************
37   REM * OBTAIN DATA FROM SHUNTFILE *
38   REM ******************************
40   X$ = "SHUNTFILE"
45   ONERR  GOTO 500
50   PRINT D$;"OPEN";X$: PRINT D$;"READ";X$
52   INPUT DR: INPUT PR: INPUT CV$: INPUT FLAG$: INPUT PT$:
     INPUT SERVO$
53   INPUT TP$:QQ = 3
54   GOSUB 900:HR$ = QQ$: GOSUB 900:P1$ = QQ$: GOSUB 900:P2$ =
     QQ$: GOSUB 900:P3$ = QQ$: GOSUB 900:VP$ = QQ$:QQ = 4:
     GOSUB 900:SR$ = QQ$
56   GOSUB 900:CO$ = QQ$:QQ = 3: GOSUB 900:SV$ = QQ$: GOSUB
     900:RP$ = QQ$: GOSUB 900:P4$ = QQ$: GOSUB 900:P5$ = QQ$:
     GOSUB 900:P6$ = QQ$
57   GOSUB 900:PW$ = QQ$:QQ = 4: GOSUB 900:PR$ = QQ$:
66   PRINT D$;"CLOSE"
68   POKE 216,0:Z$ = ""
70   Z$ = TP$ + HR$ + P1$ + P2$ + VP$ + PW$ + P3$ + T1$ + T2$ +
     CO$ + SV$ + SR$ + PR$: REM * CONCATENATE DATA *
80   GOSUB 10000
100  REM *****************
101  REM * PRINT ROUTINE *
102  REM *****************
103  PRINT D$
104  PRINT D$;"PR#";PR
116  PRINT
120  HTAB 2: PRINT "TIME";: HTAB 9: PRINT "QS/QT A-V O2CONS
```

```
    02DEL  PEEP"
220 QQ$ = Q$:QQ = 2: GOSUB 1000:Q$ = QQ$
222 QQ$ = AV$:QQ = 3: GOSUB 1000:AV$ = QQ$
224 QQ$ = OC$:QQ = 3: GOSUB 1000:OC$ = QQ$
226 QQ$ = OA$:QQ = 4: GOSUB 1000:OA$ = QQ$
228 QQ$ = PE$:QQ = 2: GOSUB 1000:PE$ = QQ$
230 Z$ = Z$ + Q$ + AV$ + OC$ + OA$ + PE$
240 IF CX$ = "N" THEN Z$ = Z$ + "EOF": GOTO 270
250 Z$ = Z$ + PA$ + AS$ + PV$ + VS$
270 PRINT TP$;"    ";Q$;"   ";AV$;"  ";OC$;"   ";OA$;"
    ";PE$
276 PRINT
280 PRINT D$: PRINT D$;"PR#0"
282 REM ************************************
283 REM * SAVE DATA STRING TO 'SHUNT DATA' *
284 REM ************************************
285 X$ = NA$ + " SHUNT DATALEN"
287 ONERR  GOTO 294
290 PRINT D$;"OPEN";X$: PRINT D$;"READ";X$: INPUT FL: PRINT
    D$;"CLOSE"
294 POKE 216,0
295 PRINT D$;"OPEN";X$:FL = FL + 1: PRINT D$;"WRITE";X$:
    PRINT FL: PRINT D$;"CLOSE"
300 X$ = NA$ + " SHUNT DATA"
305 PRINT D$;"OPEN";X$
310 PRINT D$;"POSITION";X$;",R";FL - 1
315 PRINT D$;"WRITE";X$
320 PRINT Z$
325 PRINT D$;"CLOSE"
350 GOSUB 10
370 VTAB 10: PRINT "LOADING MAIN PROGRAM"
380 X$ = "NEXTPROG"
390 PRINT D$;"OPEN";X$: PRINT D$;"READ";X$
400 INPUT N$: PRINT D$;"CLOSE": PRINT D$;"RUN";N$
499 REM *****************
500 REM * NO SHUNT FILE *
501 REM *****************
510 HOME : VTAB 5: PRINT "NO FILE AVAILABLE "
```

```
512  STOP
597  REM ************
598  REM * GET NAME *
599  REM ************
600  PRINT D$;"OPEN NAMEFILE": PRINT D$;"READ NAMEFILE"
610  INPUT NA$: INPUT NO$: PRINT D$;"CLOSE NAMEFILE": RETURN
900  INPUT QQ$
1000  REM * STR LEN ADJUST *
1010  IF  LEN (QQ$) < QQ THEN QQ$ = " " + QQ$: GOTO 1010
1020  IF  LEN (QQ$) > QQ THEN QQ$ =  LEFT$ (QQ$,QQ)
1030  RETURN
9997  REM ********************************
9998  REM * SHUNT AND CARDIAC OUTPUT SUB *
9999  REM ********************************
10000  HOME : VTAB 2
10010  GOTO 10140
10100  FOR I = 1 TO  LEN (X$):XX$ =  MID$ (X$,I,1): IF XX$ <
       "0" OR XX$ > "9" THEN  IF XX$ <  > "." THEN XX$ = "N":
       RETURN
10110  NEXT :XX$ = "Y": RETURN
10140  INPUT "BAR PRES MMHG=";X$: GOSUB 10100
10145  IF XX$ = "N" THEN 10000
10150  IF X$ < "730" OR X$ > "790" THEN 10000
10155  P =  VAL (X$): REM * BAR PRES *
10160  VTAB 4: PRINT "ARTERIAL"
10170  VTAB 4: HTAB 10: INPUT "PH  = ";X$: GOSUB 10100
10172  IF XX$ = "N" THEN 10170
10175  AH =  VAL (X$): REM * ART PH *
10180  IF AH < 6.7 OR AH > 7.9 THEN 10170
10190  VTAB 5: HTAB 10: INPUT "PCO2= ";X$: GOSUB 10100
10192  IF XX$ = "N" THEN 10190
10195  AC =  VAL (X$): REM * ART CO2 *
10200  IF AC < 10 OR AC > 140 THEN 10190
10210  VTAB 6: HTAB 10: INPUT "PO2 = ";X$: GOSUB 10100:AO =
       VAL (X$): REM * ART PO2 *
10212  IF XX$ = "N" THEN 10210
10220  IF AO < 25 OR AO > 700 THEN 10210
10230  VTAB 4: HTAB 23: PRINT "VENOUS"
```

```
10240  VTAB 4: HTAB 30: INPUT "PH  = ";X$: GOSUB 10100:VH =
       VAL (X$)
10242  IF XX$ = "N" THEN 10240
10250  IF VH > = AH THEN 10240: REM * VEN PH MUST BE < ART PH *
10260  IF VH < 6.7 OR VH > 7.9 THEN 10240
10270  VTAB 5: HTAB 30: INPUT "PCO2= ";X$: GOSUB 10100:VC =
       VAL (X$)
10272  IF XX$ = "N" THEN 10270
10280  IF VC <  = AC THEN 10270
10290  IF VC < 10 OR VC > 140 THEN 10270: REM * VEN PCO2 MUST
       BE > ART PO2 *
10300  VTAB 6: HTAB 30: INPUT "PO2 = ";X$: GOSUB 10100:VO =
       VAL (X$)
10302  IF XX$ = "N" THEN 10300
10310  IF VO >  = AO THEN 10300: REM * VEN PO2 MUST BE < ART
       PO2 *
10320  IF VO < 20 OR VO > 700 THEN 10300
10330  VTAB 8: INPUT "HB G%= ";X$: GOSUB 10100:B =  VAL (X$)
10332  IF XX$ = "N" THEN 10330
10340  IF B < 4 OR B > 22 THEN 10330: REM * HG MUST BE >4 AND
       <22 *
10350  VTAB 10: INPUT "%FIO2= ";X$: GOSUB 10100:FI =  VAL (X$)
10352  IF XX$ = "N" THEN 10350
10355  IF FI < 20 OR FI > 100 THEN 10350: REM * FIO MUST BE
       >20 AND <100
10357  VTAB 14: HTAB 10: INPUT "PEEP = ";X$: GOSUB 10100:PE =
       VAL (X$):PE$ = X$
10360  IF XX$ = "N" THEN 10357
10361  IF PE < 0 OR PE > 30 THEN 10357
10362  VTAB 20: HTAB 10: PRINT "DATA OK ?  Y/N ";
10364  GET A$: IF A$ = "Y" THEN 10440
10366  IF A$ = "N" THEN 10130
10368  GOTO 10362
10440  O = AO:H = AH:C = AC
10450  GOSUB 10800
10452  GOSUB 10455:PA$ = QQ$: GOTO 10460
10455  QQ$ =  STR$ (S):QQ = 3: GOSUB 1000: RETURN
10460  L = (S + .0031 * AO): REM * ART O2 CONT
```

```
10470  O = VO:H = VH:C = VC
10480  GOSUB 10800
10482  GOSUB 10455:PV$ = QQ$
10490  M = (S + .0031 * VO): REM * VEN O2 CONT
10491  HOME : VTAB 10: PRINT "CO-OXIMETER RESULTS AVAILABLE ?
       Y/N ";
10492  GET CX$: IF CX$ = "Y" THEN 10498
10493  IF CX$ < > "N" THEN 10493
10494  HOME : GOTO 10508
10495  REM **************************
10496  REM * ENTER OXIMETER RESULTS *
10497  REM **************************
10498  HOME : VTAB 5: HTAB 4: INPUT "ARTERIAL OXYGEN
       SATURATION = ";SA
10500  IF SA < 60 OR SA > 100 THEN 10498
10502  VTAB 8: HTAB 4: INPUT "VENOUS OXYGEN SATURATION = ";SV
10503  S = SA
10504  IF SV < 20 OR SV > S THEN 10502
10506  S = SA: GOSUB 10455:AS$ = QQ$:S = SV: GOSUB 10455:VS$ =
       QQ$
10507  L =  VAL (AS$) * B * 1.34 + AO * .0031:M =  VAL (VS$) *
       B * 1.34 + VO * .0031: REM * ART & VEN O2 CONTENTS *
10508  E = FI / 100:F = (P - 47) * E - C * (E + (1 - E) / .8)
10510  O = F:H = AH:C = AC
10520  GOSUB 10800
10530  N = (S + .0031 * F): REM * CAP O2 CONTENT *
10540  D =  VAL (CO$)
10560  Q$ =  STR$ ( INT (((N - L) * 100 / (N - M)) + .5)): REM
       * QS/QT *
10590  AV$ =  STR$ ( INT (((L - M)) * 1E1 + .5) / 1E1): REM *
       D(A-V)O2 *
10640  OA$ =  STR$ ( INT ((L * D * 10) + .5)): REM * O2
       AVAILABILITY *
10660  OC$ =  STR$ ( INT ((D * (L - M) * 10) + .5)):REM * O2
       CONSUMPTION *
10680  CAL$ =  STR$ ( INT ((D * (L - M) * 60.9552) + .5))
10700  IF SA$ = "" THEN SA$ = "---"
10702  IF SV$ = "" THEN SV$ = SA$
```

```
10750  RETURN : REM * TO 60 *
10770  REM ***********************************************
10780  REM * CALCULATE SATURATION FROM PARTIAL PRESSURE *
10790  REM ***********************************************
10800  O = O * 10 ^ ((.48 * (H - 7.4)) + .006 * ( LOG (40) -
       LOG (C)))
10810  S = 99.95 - 100 / (1 + ((O + 7) / 33.7) ^ 3.3)
10820  S = S - .5 / (1 + ((O - 130) / 35) ^ 2)
10830  S = S + .45 / (1 + ((O - 68) / 12) ^ 6)
10840  S = S - .5 / (1 + ((O - 35) / 3) ^ 4)
10850  S = S - .5 / (1 + ((O - 15) / 4) ^ 4)
10860  S = S + .35 / (1 + ((O - 26) / 3) ^ 6)
10870  S = S + .2 / (1 + ((O - 53) / 8) ^ 4)
10880  S = S - .4 / (1 + ((O - 40) / .9) ^ 4)
10890  S = S - .2 / (1 + ((O - 200) / 65) ^ 8)
10900  S = S + .4 / (1 + ((O - 9) / 3) ^ 2)
10910  S = S * B * 1.34 / 100
10920  RETURN
```

```
1    REM **************
2    REM * ROCHEPRINT *
3    REM **************
4    LOMEM: 27800: DIM Z$(270): TEXT : GOSUB 60
5    D$ =  CHR$ (4): GOTO 9
6    HTAB1: CALL -868: RETURN
7    HTAB (40 -  LEN (Z$)) / 2: PRINT Z$: RETURN
9    PRINT D$;"BLOAD CHARGEN A$6000 L$100"
10   PRINT D$;"BLOAD  CHARTAB A$6800 L$400"
11   DEF  FN PLT(X) =  VAL ( MID$ (Z$(X),Z0,Z1)): DEF  FN
     TIME(X) =  VAL ( MID$ (Z$(X),1,6))
12   PRINT D$;"NOMONI,C,O": VTAB 20: GOSUB 6: PRINT
13   REM **********************
14   REM * OBTAIN CURRENT NAME *
15   REM **********************
16   PRINT D$;"OPEN NAMEFILE": PRINT D$;"READ NAMEFILE"
18   INPUT NA$: INPUT NO$: PRINT D$;"CLOSE NAMEFILE"
19   VTAB 20: HTAB 1: GOSUB 6
21   VTAB 20: HTAB 1: PRINT "NAME";: HTAB 15: PRINT NA$
22   REM ********************************************
23   REM * INITIALISE PRINTER FOR BI-DIRECTIONAL MODE *
24   REM ********************************************
25   PR = 1: PRINT D$;"PR#1": PRINT : PRINT " ": POKE  -
     12528,7: POKE  - 12526,83: POKE  - 12525,64: POKE  -
     12529,0: POKE  - 12524,0: PRINT : PRINT D$;"PR#0"
26   REM **********************************
27   REM * SELECT CVS THEN RS FOR PRINTOUT *
28   REM **********************************
29   ZZ$ = " CVS DATA":X$ = " CVS ": GOSUB 30: PRINT : PRINT
     :NI = 0:FI = 0
30   ZZ$ = " RS DATA":X$ = " RS ": GOSUB 30: GOTO 300
34   REM **********************
35   REM * OBTAIN DATA STRINGS *
36   REM **********************
37   VTAB 20: HTAB 1: GOSUB 6: PRINT :XD$ = NA$ + ZZ$ + "LEN":
     ONERR  GOTO 300
38   PRINT D$;"OPEN";XD$: PRINT D$;"READ";XD$: INPUT FI: PRINT
     D$;"CLOSE"
```

```
39  NI = FI
40  XD$ = NA$ + ZZ$
41  PRINT D$;"OPEN";XD$
42  PRINT D$;"READ";XD$
46  VTAB 20: HTAB 9: PRINT "OBTAINING FILE "
47  FOR I = 1 TO FILE: VTAB 20: HTAB 30: PRINT I: INPUT Z$(I)
48  NEXT
49  PRINT D$;"CLOSE": POKE 216,0
50  GOTO 100
60  POKE 216,0: HOME :Z$ = "GLASGOW ROYAL INFIRMARY": GOSUB 7:
    PRINT :Z$ = "RESPIRATORY INTENSIVE CARE UNIT": GOSUB 7
62  Z$ = "REPORT PRINT PROGRAM ": PRINT : GOSUB 7: RETURN
100  GOSUB 13001: IF NM < 2 THEN 300: REM * SELECT FILES FOR
     PRINTING *
110  GOSUB 24000: REM * OBTAIN DATE/TIME *
120  PRINT D$;"PR#";PR: PRINT NA$,NO$,D;"/";MT;"/1981": PRINT
130  GOSUB 5000: PRINT D$;"PR#0": REM * PRINT DATA *
134  REM **********************************************
135  REM * INITIALISE PRINTER FOR UNI-DIRECTIONAL MODE *
136  REM **********************************************
140  PR = 1: PRINT D$;"PR#1": PRINT : PRINT " ": POKE  -
     12528,7: POKE  - 12526,83: POKE  - 12525,64: POKE  -
     12529,255: POKE  - 12524,0: PRINT : PRINT D$;"PR#0"
200  GOSUB 13260: REM * PRINT TREND GRAPH *
210  RETURN
230  REM *****************************************
230  REM * RETURN TO MAIN CARDIOVASCULAR PROGRAM *
240  REM *****************************************
300  X$ = "SHUNTFILE": PRINT D$;"OPEN";X$: PRINT D$;"READ";X$:
     INPUT DR: INPUT PR: INPUT CV$: INPUT FLAG$: INPUT PT$:
     PRINT D$;"CLOSE";X$
310  PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$: PRINT DR: PRINT
     PR: PRINT CV$: PRINT N: PRINT PT$: PRINT D$;"CLOSE";X$
4000  X$ = "NEXTPROG"
4010  PRINT D$;"OPEN";X$: PRINT D$;"READ";X$
4020  INPUT N$
4030  PRINT D$;"CLOSE": PRINT D$;"RUN";N$
```

```
4985  REM ***********************
4990  REM * SET PRINT FUNCTIONS *
4995  REM ***********************
5000  GOSUB 5010: GOTO 5110
5010  N2 = 0:PN = 12:S(1) = 1:L(1) = 6:S(2) = 7:L(2) = 3:S(3)
      = 10:L(3) = 3:S(4) = 13:L(4) = 3:S(5) = 16:L(5) = 3:S(6) =
      19:L(6) = 3:S(7) = 22:L(7) = 3:NV = 7:N1 = 6
5020  IF X$ = " CVS " THEN  PRINT " DATE    TIME    HR   SYS
      DIA   CVP   PAWP  RPP   CORE  DIFF"
5025  IF X$ = " CVS " THEN S(7) = 22:L(7) = 3:S(8) = 25:L(8) =
      4:S(9) = 29:L(9) = 3:NV = 9:N1 = 9: RETURN
5047  IF X$ = " RS " THEN  PRINT " DATE    TIME  RATE  IN VT
      EX VT  V MIN  PEAK PEEP COMP   WORK   RESIST": RETURN
5100  RETURN
5110  FOR X = F2 TO PP STEP  - 1
5130  FOR AB = 1 TO NVINSTR
5140  SM$(AB) =  MID$ (Z$(X),S(AB),L(AB)): IF SM$(AB) = "-99"
      THEN SM$(AB) = "---": REM * SELECT INDIVIDUAL VALUES
      FROM DATA STRINGS *
5150  NEXT AB
5160  ET =  VAL (SM$(1)): GOSUB 5290
5170  HTAB 1: IF DT$ <  > SM$(0) THEN  PRINT SM$(0);:DT$ =
      SM$(0): GOTO 5200: REM * PRINT DATE (SM$(0) IF DIFFERENT
      FROM PREVIOUS DATE *
5180  PRINT "     ";
5200  FOR J = 0 TO N1 - 1: PRINT "   ";: PRINT SM$(J + 1);:
      NEXT J: PRINT: REM * PRINT INDIVIDUAL VALUES *
5270  NEXT X
5280  PRINT : RETURN
5284  REM * ***********************
5285  REM * CALCULATE DATE & TIME *
5286  REM * ***********************
5290  GOSUB 25000: GOSUB 26000: REM * CALCULATE DATE & TIME *
5300  ND$ =  STR$ (ND): IF  LEN (ND$) < 2 THEN ND$ = " " + ND$
5310  MT$ =  STR$ (MT): IF  LEN (MT$) < 2 THEN MT$ = " " + MT$
5320  SM$(0) = ND$ + "/" + MT$: REM * DATE *
5330  H$ =  STR$ (H): IF  LEN (H$) < 2 THEN H$ = " " + H$
5340  M$ =  STR$ (M): IF  LEN (M$) < 2 THEN M$ = "0" + M$
```

```
5350  SM$(1) = H$ + "." + M$: REM * TIME *
5350  ND$ = "": MT$ = "": H$ = "": M$ = "": RETURN
12985  REM **************************
12990  REM * GRAPHICS TREND DISPLAY *
12995  REM **************************
13001  TI$ = "24H": GOSUB 13004:NM = Z: GOTO 13012: REM *
       OBTAIN DATA STRINGS FOR PAST 24HRS *
13004  IF  RIGHT$ (TI$,1) = "D" THEN Z =  VAL ( LEFT$ (TI$,
       LEN (TI$) - 1)) * 1440: RETURN
13006  IF  RIGHT$ (TI$,1) = "H" THEN Z =  VAL ( LEFT$ (TI$,
       LEN (TI$) - 1)) * 60: RETURN
13008  IF  RIGHT$ (TI$,1) = "M" THEN Z =  VAL ( LEFT$ (TI$,
       LEN (TI$) - 1)): RETURN
13010  POP : GOTO 13001: REM * INCORRECT ENTRY *
13012  TI$ = "24H": GOSUB 13004:Z9 = Z
13015  START =  FN TIME(FI) - NM
13017  FOR I = 1 TO FILE: IF START >  FN TIME(I) THEN  NEXT
13019  START =  FN TIME(I):PI = I
13021  LAST = ST + Z9: FOR I = FI TO 1 STEP  - 1: IF LAST <
       FN TIME(I) THEN  NEXT
13022  F2 = I
13023  LAST =  FN TIME(F2): GOTO 13025
13024  START =  FN TIME(PI):LAST =  FN TIME(FILE): HOME
13025  NM = LAST - START
13027  ET = ST: GOSUB 25000: GOSUB 26000:SD = DY:MS = MT:SH =
       H:SM = M
13028  ET = LA: GOSUB 25000: GOSUB 26000:LM$ =  STR$ (MT): IF
       LEN (LM$) < 2 THEN LM$ = "0" + LM$
13058  IF NM < 2 THEN  HOME : PRINT " NOT ENOUGH VALUES TO
       PLOT": FOR II = 0 TO 1000: NEXT : RETURN
13080  PP = PI: RETURN
13245  REM ****************
13250  REM * HI-RES PLOTS *
13255  REM ****************
13260  GOSUB 13420
13262  IF X$ = " CVS " THEN 14700
13264  IF X$ = " RS " THEN 16000
13280  AX = 0:CX = 0:PP = PI
```

```
13281  IF BX = 0 THEN BA = 30: GOTO 13290
13282  BA = 30 + 212 / NM * (BX + CX): REM  * BASE ADDRESS *
13290  X1 = BA
13300  Y1 = 190 - Z3 *  FN PLT(PP) + Z2
13305  IF PP = FI OR PP = F2 THEN  RETURN
13310  IF  FN PLT(PP) < LO THEN PP = PP + 1:CX = CX + ( FN
       TIME(PP) -  FN TIME(PP - 1)): GOTO 13282
13312  IF Y1 < 0 THEN Y1 = 0
13314  IF X1 > 278 THEN X1 = 278
13316  IF X1 < 30 THEN X1 = 30
13320  FOR X = PP + 1 TO F2:Y2 = 190 - Z3 *  FN PLT(X) + Z2
13330  IF  FN TIME(X) <  =  FN TIME(X - 1) THEN X = X + 1:
       GOTO 13410
13349  X2 = X1 + 212 / NM * ( FN TIME(X) -  FN TIME(X - 1)) +
       AX
13374  IF  FN PLT(X) < LO THEN AX = AX + 212 / NM * ( FN
       TIME(X) -  FN TIME(X - 1)): GOTO 13410
13375  IF X2 > 278 THEN X2 = 278
13376  IF Y2 < 0 THEN Y2 = 0
13377  IF X2 < 30 THEN X2 = 30
13378  AX = 0
13379  IF Y2 > 189 THEN Y2 = 189
13380  FOR II = 0 TO LP: HPLOT X1 + II,Y1 + II TO X2 + II,Y2 +
       II: NEXT
13390  HPLOT X1 + 1,Y1 + 1 TO X2 + 1,Y2 + 1
13400  X1 = X2:Y1 = Y2
13410  NEXT X: RETURN
13420  GOTO 13430: REM  * AXES PLOT *
13422  FOR I = 3 TO 7: IF N / I -  INT (N / I) < .01 THEN NX =
       2 * I: RETURN
13424  NEXT I:N = N + 1: GOTO 13422
13430  IF NM < 61 THEN NT$ = "MINS":N = NM: GOSUB 13422:BX = N
       - NM:NM = N: GOTO 13505
13450  IF NM < 2160 THEN BX = SM: IF NM + BX < 2160 THEN NT$ =
       "TIME": GOTO 13498
13470  NT$ = "DATE":BX = SH * 60 + SM
13490  ND =  INT ((NM + BX) / 1440) + 1
13492  N = ND: GOSUB 13422:ND = N:NM = ND * 1440
```

```
13493  NT$ = "DATE": GOTO 13505
13498  NH = (NM + BX) / 60: IF (NM + BX) / 60 -  INT ((NM +
       BX) / 60) <  > 0 THEN NH =  INT ((NM + BX) / 60) + 1
13500  N = NH: GOSUB 13422:NH = N:NM = NH * 60
13505  HGR2 : HCOLOR= 3: POKE 54,0: POKE 55,96: CALL 1002
13506  FOR V = 0 TO NX:X = 30 + V * 212 / NX: HPLOT X,174 TO
       X,177: HPLOT X + 1,174 TO X + 1,177: NEXT
13510  FOR V = 0 TO NX STEP 2:X = 30 + V * 212 / NX: HPLOT
       X,174 TO X,180: HPLOT X + 1,174 TO X + 1,180: NEXT
13520  HPLOT 30,10 TO 30,180: HPLOT 31,10 TO 31,180
13525  IF NT$ = "MINS" THEN LX = SM:NZ = NM: FOR V = 0 TO NX
       STEP 2: VTAB 24:X = V * 212 / NX:A$ =  STR$ ( INT (LX +
       V * NZ / NX + .5)):HT =  INT (X / 6.975 + 5.5 -  LEN (A$)
       / 2): GOTO 13542
13530  IF NT$ = "TIME" THEN LX = SH:BX = SM:NZ = NH: GOTO
       13540
13532  LX = SD:BX = SH * 60 + SM:NZ = ND
13540  FOR V = 0 TO NX STEP 2: VTAB 24:X = V * 212 / NX:A$ =
       STR$ ( INT (LX + V * NZ / NX + .5)):HT =  INT (X /
       6.975 + 5.5  -  LEN (A$) / 2)
13542  A =  VAL (A$)
13544  IF NT$ = "DATE" THEN 13547
13545  IF NT$ = "TIME" AND A > 24 THEN A = A - 24: GOTO 13545
13546  IF NT$ = "MINS" AND A > 60 THEN A = A - 60
13547  GOTO 13553
13548  RESTORE : READ J:II = 0
13549  II = II + 1: READ J: IF II < MS THEN 13549
13550  IF A > J THEN A = A - J
13551  IF A > J THEN 13549
13553  HT = HT + 1
13554  HTAB HT: PRINT A: NEXT
13555  VTAB 23: HTAB 37: PRINT NT$;: IF NT$ = "DATE" THEN
       VTAB 24: HTAB 38: PRINT "/";LM$;
13600  RETURN
14700  REM  * BP *
14710  JJ = 160: VTAB 2: PRINT JJ: VTAB 7: PRINT JJ - 40: VTAB
       12: HTAB 2: PRINT JJ - 80: VTAB 17: HTAB 2: PRINT JJ -
       120
```

```
14720  HPLOT 30,50 TO 242,50: HPLOT 30,90 TO 242,90
14730  HCOLOR= 5
14735  VTAB 6: HTAB 37: PRINT "BP": HPLOT 252,49 TO 268,49 TO
       268,48 TO 252,48
14750  Z0 = 10:Z1 = 3:Z2 =  - 20:Z3 = 1:LO = 40:LP = 1: GOSUB
       13280
14770  Z0 = 13:LO = 20: GOSUB 13280: REM * DIA *
14790  REM  * HR *
14800  HCOLOR= 6
14805  VTAB 10: HTAB 37: PRINT "HR": FOR II = 1 TO 3: HPLOT
       253,80 + II TO 272,80 + II: NEXT
14810  Z0 = 7:LP = 1: GOSUB 13280
14840  REM  * CVP *
14850  HCOLOR= 1
14855  HPLOT 30,160 TO 242,160
14857  VTAB 14: HTAB 37: PRINT "CVP": FOR II = 0 TO 2: HPLOT
       252,116 + II TO 273,116 + II: NEXT
14858  VTAB 19: HTAB 3: PRINT "+5": VTAB 22: HTAB 3: PRINT
       "-5"
14860  Z0 = 16:Z1 = 3:Z2 =  - 30:Z3 = 3:LO =  - 10: GOSUB
       13280
14865  HCOLOR= 2: VTAB 18: HTAB 37: PRINT "PAWP": HPLOT
       252,146 TO 278,146 TO 278,147 TO 252,147
14870  Z0 = 19:LP = 1: GOSUB 13280: REM * PAWP *
14895  VTAB 1: PRINT NA$;: HTAB 32: PRINT  MID$ (T$,7,2);".";
       MID$ (T$,10,2);":"; MID$ (T$,13,2)
14900  PRINT D$;"PR#1": PRINT : PRINT : PRINT : PRINT  CHR$
       (17): PRINT D$;"PR#0"
14902  CALL 1002: PRINT D$;"PR#0": PRINT D$;"IN#0": TEXT :
       RETURN
16000  REM * RESP *
16010  HPLOT 30,20 TO 245,20
16020  VTAB 2: HTAB 3: PRINT "10": VTAB 4: HTAB 4: PRINT "0":
       VTAB 3: HTAB 37: PRINT "PEEP"
16030  HPLOT 30,70 TO 245,70: VTAB 9: HTAB 37: PRINT "VT"
16050  HCOLOR= 1:Z0 = 24:Z1 = 2:Z2 =  - 160:Z3 = 2: GOSUB
       13280: REM * PEEP *
16060  VTAB 8: HTAB 2: PRINT "700": VTAB 11: HTAB 2: PRINT
```

```
      "500"
16070 HCOLOR= 5:Z0 = 13:Z1 = 4:Z2 =  - 60:Z3 = .1: GOSUB
      13280: REM VEXP
16080 VTAB 17: HTAB 3: PRINT "30": VTAB 19: HTAB 3: PRINT
      "20"
16090 HCOLOR= 3: HPLOT 30,130 TO 242,130: HPLOT 30,150 TO
      242,150
16100 VTAB 16: HTAB 37: PRINT "PK P": VTAB 20: HTAB 37: PRINT
      "COMP"
16110 HCOLOR= 2:Z0 = 22:Z1 = 2:Z2 = 0:Z3 = 2: GOSUB 13280:
      HPLOT 255,130 TO 271,130 TO 271,131 TO 255,131
16130 HCOLOR= 6:Z0 =26:Z1 = 2:Z3 = 2: GOSUB 13280: HPLOT
      255,163 TO 271,163 TO 271,164 TO 255,164: REM COMP
16150 GOSUB 24000: VTAB 1: HTAB 35: PRINT  MID$ (T$,7,2);".";
      MID$ (T$,10,2)
16160 PRINT D$;"PR#1": PRINT : PRINT : PRINT : PRINT  CHR$
      (17): PRINT D$;"PR#0"
16170 CALL 1002: PRINT D$;"PR#0": PRINT D$;"IN#0": TEXT :
      RETURN
24004 REM ****************
24005 REM *GET DATE/TIME *
24006 REM ****************
24010 D$ =  CHR$ (4): PRINT D$:SLOT = 3
24030 PRINT D$;"IN#";SL: PRINT D$;"PR#"SL: INPUT T$: PRINT
      D$;"IN#0": PRINT D$;"PR#0"
24040 MT =  VAL ( MID$ (T$,1,2))
24050 D =  VAL ( MID$ (T$,4,2))
24060 H =  VAL ( MID$ (T$,7,2))
24070 M =  VAL ( MID$ (T$,10,2))
24080 S =  VAL ( MID$ (T$,13,6))
24090 RESTORE
24100 DTD = 0
24110 FOR I = 1 TO MT: READ J:DTD = DTD + J: NEXT
24120 DATA    0,31,28,31,30,31,30,31,31,30,31,30,31
24130 DTD = DTD + D
24140 IF MT > 2 AND L = 1 THEN DTD = DTD + 1
24150 STD = DTD * 86400 + H * 3600 + M * 60 + S
24160 S = STD
```

```
24170  RETURN
25000  REM *****************
25005  REM * MIN->DAYS/HRS *
25006  REM *****************
25030  MT = 0
25040  DY =  INT (ET / 1440)
25050  ET = ET - DY * 1440
25060  H = INT (ET / 60)
25070  M = ET - H * 60
25080  RETURN
26000  REM ***************
26005  REM * DAYS->DATE *
26006  REM ***************
26010  IF L = 1 AND DY = 60 THEN MT = 2:DY = 29: GOTO 26070
26015  IF L = 1 AND DY > 60 THEN DY = DY - 1
26017  RESTORE
26018  READ J
26020  FOR II = 1 TO 12: READ J:MT = MT + 1
26030  IF DY - J < 1 THEN ND = DY: GOTO 26070
26035  DY = DY - J
26040  NEXT
26070  RETURN
```

```
1  REM *****************
2  REM * ROCHEPRINTEXT *
3  REM *****************
4  DIM Z$(270): TEXT : GOSUB 60
5  D$ =  CHR$ (4): GOTO 10
6  FOR I = 1 TO 39: PRINT " ";: NEXT : RETURN
7  HTAB (40 -  LEN (Z$)) / 2: PRINT Z$: RETURN
10  DEF  FN TIME(X) =  VAL ( LEFT$ (Z$(X),6))
11  ONERR  GOTO 400
12  PRINT D$;"NOMONI,C,O": VTAB 20: GOSUB 6: PRINT
14  REM ***********************
15  REM * OBTAIN CURRENT NAME *
16  REM ***********************
17  PRINT D$;"OPEN NAMEFILE": PRINT D$;"READ NAMEFILE"
18  INPUT NA$: INPUT NO$: PRINT D$;"CLOSE NAMEFILE"
19  VTAB 20: HTAB 1: GOSUB 6
20  VTAB 20: HTAB 1: PRINT "NAME";: HTAB 15: PRINT NA$: PR = 1
24  REM **************************************************
25  REM * INITIALISE SILENTYPE FOR BI-DIRECTIONAL MODE *
26  REM **************************************************
27  PRINT D$;"PR#";PR: PRINT : PRINT " ": POKE  - 12528,7:
    POKE  - 12526,83: POKE  - 12525,64: POKE  - 12529,0: POKE
    - 12524,0: PRINT : PRINT D$;"PR#0"
29  REM ***********************
30  REM * OBTAIN DATA STRINGS *
31  REM ***********************
32  ZZ$ = " TEXT"
33  VTAB 20: HTAB 1: GOSUB 6: PRINT :XD$ = NA$ + ZZ$ + "LEN"
34  PRINT D$;"OPEN";XD$: PRINT D$;"READ";XD$: INPUT FI: PRINT
    D$;"CLOSE"
35  XD$ = NA$ + ZZ$
41  PRINT D$;"OPEN";XD$
42  PRINT D$;"READ";XD$
46  VTAB 20: HTAB 9: PRINT "OBTAINING FILE "
47  FOR I = 1 TO FI: VTAB 20: HTAB 30: PRINT I: INPUT Z$(I)
48  NEXT
49  PRINT D$;"CLOSE": POKE 216,0
50  GOTO 90
```

```
60  POKE 216,0: HOME :Z$ = "GLASGOW ROYAL INFIRMARY": GOSUB 7:
    PRINT :Z$ = "RESPIRATORY INTENSIVE CARE UNIT": GOSUB 7
62  Z$ = "TEXT REPORT": PRINT : GOSUB 7: RETURN
90  PR = 1
110  GOSUB 24000: REM * GET DATE/TIME *
115  GOSUB 13001: REM * SELECT DATA STRINGS FOR REPORT *
120  PRINT D$;"PR#";PR: GOSUB 60: PRINT : PRINT
     NA$,NO$,D;"/";MT;"/1982": PRINT
125  FOR I = PP TO F2
130  ET =  VAL ( LEFT$ (Z$(I),6))
140  GOSUB 25000: GOSUB 26000: REM * CALCULATE DATE & TIME *
150  ND$ =  STR$ (ND): IF  LEN (ND$) < 2 THEN ND$ = " " + ND$
160  MT$ =  STR$ (MT): IF  LEN (MT$) < 2 THEN MT$ = " " + MT$
170  SM$(0) = ND$ + "/" + MT$
180  HTAB 1: IF DT$ <  > SM$(0) THEN  PRINT SM$(0);:DT$ =
     SM$(0): GOTO 192: REM * PRINT DATE IF DIFFERENT FROM
     PREVIOUS DATE *
190  PRINT "     ";
192  H$ =  STR$ (H): IF  LEN (H$) < 2 THEN H$ = " " + H$
194  M$ =  STR$ (M): IF  LEN (M$) < 2 THEN M$ = "0" + M$
196  SM$(1) = H$ + "." + M$
198  PRINT "   ";: PRINT SM$(1);: REM * PRINT TIME *
200  PRINT "   ";: PRINT  MID$ (Z$(I),7,240): REM * PRINT TEXT
     COMMENTS *
210  NEXT
220  PRINT D$;"PR#0"
285  REM *****************************************
280  REM * RETURN TO MAIN CARDIOVASCULAR PROGRAM *
295  REM *****************************************
300  X$ = "SHUNTFILE": PRINT D$;"OPEN";X$: PRINT D$;"READ";X$:
     INPUT DR: INPUT PR: INPUT CV$: INPUT FLAG$: INPUT PT$:
     PRINT D$;"CLOSE";X$
310  PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$: PRINT DR: PRINT
     PR: PRINT CV$: PRINT N: PRINT PT$: PRINT D$;"CLOSE";X$
320  GOTO 410
```

```
380  REM *******************
385  REM * DISC READ ERROR *
390  REM *******************
400  PRINT D$;"PR#0": HOME : VTAB 10:Z$ = "FILE UNAVAILABLE":
     GOSUB 7: PRINT :Z$ = "LOADING ROCHE PROGRAM": GOSUB 7:
     PRINT
410  PRINT D$;"PR#0": HOME : VTAB 10:Z$ = "LOADING ROCHE
     PROGRAM": GOSUB 7: PRINT
4000  X$ = "NEXTPROG"
4010  PRINT D$;"OPEN";X$: PRINT D$;"READ";X$
4020  INPUT N$
4030  PRINT D$;"CLOSE": PRINT D$;"RUN";N$
12885  REM *******************************
12900  REM * SELECT FILES FOR PAST 12HRS *
12905  REM *******************************
13001  TI$ = "12H": GOSUB 13004:NM = Z: GOTO 13012
13004  IF  RIGHT$ (TI$,1) = "D" THEN Z =  VAL ( LEFT$ (TI$,
       LEN (TI$) - 1)) * 1440: RETURN
13006  IF  RIGHT$ (TI$,1) = "H" THEN Z =  VAL ( LEFT$ (TI$,
       LEN (TI$) - 1)) * 60: RETURN
13008  IF  RIGHT$ (TI$,1) = "M" THEN Z =  VAL ( LEFT$ (TI$,
       LEN (TI$) - 1)): RETURN
13010  POP : GOTO 13001
13012  TI$ = "12H": GOSUB 13004:Z9 = Z
13015  START =  FN TIME(FI) - NM
13017  FOR I = 1 TO FILE: IF START >  FN TIME(I) THEN  NEXT
13019  START =  FN TIME(I):PI = I
13021  LAST = ST + Z9: FOR I = FI TO 1 STEP  - 1: IF LAST <
       FN TIME(I) THEN  NEXT
13022  F2 = I
13023  LAST =  FN TIME(F2): GOTO 13025
13024  START =  FN TIME(PI):LAST =  FN TIME(FILE): HOME
13025  NM = LAST - START
13027  ET = ST: GOSUB 25000: GOSUB 26000:SD = DY:MS = MT:SH =
       H:SM = M
13028  ET = LA: GOSUB 25000: GOSUB 26000:LM$ =  STR$ (MT): IF
       LEN (LM$) < 2 THEN LM$ = "0" + LM$
13080  PP = PI: RETURN
```

```
26035  DY = DY - J
26040  NEXT
26070  RETURN
```

```
3   REM ******************
4   REM * R2000CALIBRATE *
5   REM ******************
6   ADC =  - 16384 + (256 * 4):CHAN = ADC + 1: REM * ADC SLOT
    4 *
10  D$ =  CHR$ (4)
20  GOTO 60
45  HTAB (40 -  LEN (Z$)) / 2: PRINT Z$;: RETURN
49  REM **************************************
50  REM * OBTAIN PREVIOUS CALIBRATION VALUES *
51  REM **************************************
60  X$ = "R2000VARIABLES"
65  ONERR  GOTO 90
70  PRINT D$;"OPEN";X$: PRINT D$;"READ";X$
73  FOR I = 1 TO 4: INPUT V$(I): NEXT
75  INPUT CRT
76  INPUT CSYS(1): INPUT CSYS(2): INPUT CSYS(3): INPUT
    CPUL(1): INPUT CPUL(2): INPUT CPUL(3): INPUT CCVP: INPUT
    CO2SA
77  INPUT ZHR: INPUT ZSYS(1): INPUT ZSYS(2): INPUT ZSYS(3):
    INPUT ZPUL(1): INPUT ZPUL(2): INPUT ZPUL(3): INPUT ZCVP:
    INPUT ZCO2SA: PRINT D$;"CLOSE"
80  GOTO 95
84  REM *****************************
85  REM * DEFAULT CHANNEL FUNCTIONS *
86  REM *****************************
90  V$(1) = "HR":V$(2) = "PAP":V$(3) = "CVP":V$(4) = "SYS"
91  ER$ = "Y"
95  PRINT D$;"CLOSE": POKE 216,0
96  REM *****************************
97  REM * DISPLAY CHANNEL FUNCTIONS *
98  REM *****************************
100  HOME :Z$ = "GLASGOW ROYAL INFIRMARY": GOSUB 45: PRINT :
     PRINT :Z$ = "ROCHE 2000 SET UP": GOSUB 45: PRINT
105  VTAB 8: PRINT "CHANNEL";: HTAB 18: PRINT "FUNCTION"
106  VTAB 9: PRINT "=======";: HTAB 18: PRINT "========"
110  VTAB 12: FOR I = 1 TO 4: PRINT "   ";I;".";: HTAB 21:
     PRINT V$(I): PRINT : NEXT
```

```
115  VTAB 22: HTAB 1: PRINT "A(LTER   L(EAVE  VARIABLES ";:
     GET A$: IF A$ = "L" THEN 132
116  IF A$ <  > "A" THEN 115
117  REM ***************************
118  REM * ALTER CHANNEL FUNCTIONS *
119  REM ***************************
120  VTAB 22: HTAB 1: PRINT "ENTER CHANNEL NUMBER TO BE
     ALTERED ";: GET A$: IF A$ < "1" OR A$ > "4" THEN 120
125  VTAB 22: HTAB 1: CALL  - 868: PRINT "ENTER THE NEW
     FUNCTION AND TYPE <RETURN> ";: INPUT V$( VAL (A$))
130  GOTO 100
131  IF ER$ = "Y" THEN  GOSUB 700: REM * NO PREVIOUS
     CALIBRATION *
132  GOSUB 5000: REM * DISPLAY CHANNEL VALUES :
133  X$ = "R2000VARIABLES": VTAB 4: PRINT
134  REM *************************************
135  REM * WRITE CALIBRATION VALUES TO DISC *
136  REM *************************************
137  PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$: FOR I = 1 TO 4:
     PRINT V$(I): NEXT : PRINT CRT
138  PRINT CSYS(1): PRINT CSYS(2): PRINT CSYS(3): PRINT
     CPUL(1): PRINT CPUL(2): PRINT CPUL(3): PRINT CCVP: PRINT
     CO2SA
139  PRINT ZHR: PRINT ZSYS(1): PRINT ZSYS(2): PRINT ZSYS(3):
     PRINT ZPUL(1): PRINT ZPUL(2): PRINT ZPUL(3): PRINT ZCVP:
     PRINT ZCO2SA: PRINT D$;"CLOSE"
140  HOME : VTAB 10:Z$ = "PLEASE WAIT": GOSUB 45: PRINT :
     PRINT :Z$ = "THE NEXT PROGRAM IS NOW LOADING": GOSUB 45:
     PRINT
150  PRINT D$;"RUN R2000"
699  REM *************
700  REM * CALIBRATE *
701  REM *************
704  POKE  - 16368,0
705  A$ = ""
710  HOME : VTAB 5: PRINT "OPEN ALL TRANSDUCERS TO ATMOSPHERE
     AND": PRINT : PRINT "ZERO THE ANALOGUE AND DIGITAL
     DISPLAYS"
```

```
720  VTAB 10: PRINT "TYPE <SPACEBAR> WHEN READY ";: GET A$: IF
     A$ <  > " " THEN 720
724  REM ***********************
725  REM * OBTAIN ZERO OFFSETS *
726  REM ***********************
730  GOSUB 3000
732  ZHR = H1:ZSYS(1) = S1:ZSYS(2) = S2:ZSYS(3) = S3
733  ZPUL(1) = P1:ZPUL(2) = P2:ZPUL(3) = P3:ZCVP = C1
734  REM *********************************
735  REM * CALCULATE CALIBRATION FACTORS *
736  REM *********************************
737  HOME : VTAB 5: PRINT "SET HEART RATE ON CHANNEL 1 TO":
     PRINT : PRINT "80-110 //MIN AND ENTER THIS INTO": PRINT :
     PRINT "THE COMPUTER. THEN TYPE <RETURN> ";
738  INPUT HR: IF HR < 80 OR HR > 110 THEN 735
739  GOSUB 3000:CRT = HR / (H1 - ZHR)
741  REM ********************
742  REM * SYSTEMIC CHANNEL *
743  REM ********************
749  HOME : VTAB 5: PRINT "SET SYSTEMIC PRESSURE ON CHANNEL 4
     TO": PRINT : PRINT "180-210 MMHG AND ENTER THIS INTO ":
     PRINT : PRINT "THE COMPUTER THEN TYPE <RETURN> ";
750  INPUT SYS: IF SYS < 180 OR SYS > 210 THEN 740
760  GOSUB 3000:CSYS(1) = SYS / (S1 - ZSYS(1))
762  CSYS(2) = SYS / (S2 - ZSYS(2))
764  CSYS(3) = SYS / (S3 - ZSYS(3))
766  REM *********************
767  REM * PULMONARY CHANNEL *
768  REM *********************
770  HOME : VTAB 5: PRINT "SET PULMONARY PRESSURE ON CHANNEL 2
     TO": PRINT : PRINT "40-60 MMHG AND ENTER THIS INTO ":
     PRINT : PRINT "THE COMPUTER THEN TYPE <RETURN> ";
780  INPUT PUL: IF PUL < 40 OR PUL > 60 THEN 770
790  GOSUB 3000:CPUL(1) = PUL / (P1 - ZPUL(1))
792  CPUL(2) = PUL / (P2 - ZPUL(2))
794  CPUL(3) = PUL / (P3 - ZPUL(3))
```

```
796  REM ***************
797  REM * CVP CHANNEL *
798  REM ***************
799  IF V$(3) <  > "CVP" THEN 830
800  HOME : VTAB 5: PRINT "SET VENOUS PRESSURE ON CHANNEL 3
     TO": PRINT : PRINT "15-20 MMHG AND ENTER THIS INTO ":
     PRINT : PRINT "THE COMPUTER THEN TYPE <RETURN> ";
810  INPUT CVP: IF CV < 15 OR CV > 20 THEN 800
820  GOSUB 3000:CCVP = CVP / (C1 - ZCVP)
830  HOME : RETURN
2000  REM * STRING LENGTH ADJUST *
2010  FOR I = 1 TO QQ: IF  LEN (QQ$) < QQ THEN QQ$ = " " +
      QQ$: NEXT
2020  IF  LEN (QQ$) > QQ THEN QQ$ =  LEFT$ (QQ$,3)
2030  RETURN
2999  REM ***************************************
3000  REM * SELECT ADC CHANNELS FOR CALIBRATION *
3001  REM ***************************************
3010  Q = 8: GOSUB 4000:H1 = QQ
3020  REM * SYSTEMIC *
3030  Q = 6: GOSUB 4000:S1 = QQ: REM * SYSTOLIC *
3032  Q = 4: GOSUB 4000:S2 = QQ: REM * DIASTOLIC *
3034  Q = 12: GOSUB 4000:S3 = QQ: REM * MEAN *
3060  REM * PAP *
3070  Q = 2: GOSUB 4000:P1 = QQ
3072  Q = 0: GOSUB 4000:P2 = QQ
3074  Q = 1: GOSUB 4000:P3 = QQ: REM * MEAN *
3100  REM * CVP *
3110  Q = 5: GOSUB 4000:C1 = QQ: REM * MEAN *
3140  RETURN
3999  REM ************
4000  REM * READ ADC *
4001  REM ************
4010  POKE CHAN,Q:QQ =  PEEK (ADC): RETURN
```

```
4990 REM *********************
5000 REM * DISPLAY PRESSURES *
5001 REM *********************
5003 HOME
5005 Q = 8:CF = CRT:ZF = ZHR:QQ = 3: GOSUB 5150:HR$ = QQ$
5010 REM * SYS *
5020 Q = 6:CF = CSYS(1):ZF = ZSYS(1): GOSUB 5150:P1$ = QQ$
5030 Q = 4:CF = CSYS(2):ZF = ZSYS(2): GOSUB 5150:P2$ = QQ$
5040 Q = 12:CF = CSYS(3):ZF = ZSYS(3): GOSUB 5150:P3$ = QQ$
5050 REM * PAP *
5060 Q = 2:CF = CPUL(1):ZF = ZPUL(1): GOSUB 5150:P4$ = QQ$
5070 Q = 0:CF = CPUL(2):ZF = ZPUL(2): GOSUB 5150:P5$ = QQ$
5080 Q = 1:CF = CPUL(3):ZF = ZPUL(3): GOSUB 5150:P6$ = QQ$
5090 REM * CVP *
5100 IF V$(3) <  > "CVP" THEN P7$ = "-99": GOTO 5122
5110 Q = 5:CF = CCVP:ZF = ZCVP: GOSUB 5150:P7$ = QQ$
5122 VTAB 1: HTAB 1: PRINT "HR   = ";HR$;"    "
5124 VTAB 3: PRINT "SYS = ";P1$;"/";P2$,"MEAN =";P3$
5126 VTAB 5: PRINT "PUL = ";P4$;"/";P5$,"MEAN =";P6$
5128 VTAB 7: PRINT "CVP = ";P7$
5130 VTAB 12: PRINT "SAT = ";SA$
5132 VTAB 22: PRINT "TYPE <SPACEBAR> TO RECALIBRATE": VTAB
     24: PRINT "TYPE <RETURN> TO CONTINUE ";
5134 IF  PEEK ( - 16384) = 128 + 13 THEN  RETURN
5136 IF  PEEK ( - 16384) = 128 + 32 THEN  GOSUB 700
5140 GOTO 5005
5145 REM ***************************************
5150 REM * READ ADC & RETURN CALIBRATED VALUE *
5155 REM ***************************************
5160 POKE CHAN,Q:QQ$ =  STR$ (( PEEK (ADC) - ZF) * CF): GOSUB
     2000: RETURN
```

```
2    REM **************
3    REM * R2000SETUP *
4    REM **************
5    TEXT : HOME : GOTO 13
7    HTAB (40 -  LEN (Z$)) / 2: PRINT Z$;: RETURN
13   Z$ = "GLASGOW ROYAL INFIRMARY": HOME : VTAB 3: GOSUB 7:
     PRINT :Z$ = "RESPIRATORY INTENSIVE CARE UNIT": GOSUB 7:
     PRINT:Z$ = "ROCHE 2000 SYSTEM PROGRAM": GOSUB 7:IPPV$ = "IPPV"
20   GOSUB 30000: REM * SET FLAGS *
30   DR = 30: REM * DATA RETRIEVAL TIME *
40   FLAG$ = "N"
50   D$ =  CHR$ (4)
60   VTAB 10: HTAB 1: CALL  - 868: VTAB 15:Z$ = "THE NEXT
     PROGRAM IS NOW LOADING": GOSUB 7: VTAB 15: HTAB 1: CALL
     - 868: VTAB 18: CALL  - 868
380   REM ******************************
390   REM * WRITE FLAGS INTO SHUNT FILE *
391   REM ******************************
395   PRINT
400   X$ = "SHUNTFILE": PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$
405   PRINT DR: PRINT PR: PRINT EAR$: PRINT FLAG$: PRINT IPPV$:
      PRINT SERVO$
500   PRINT D$;"RUN R2000"
1999   REM **************
2000   REM * STR ADJUST *
2001   REM **************
2010   FOR I = 1 TO QQ: IF  LEN (QQ$) < QQ THEN QQ$ = " " +
       QQ$: NEXT
2020   IF  LEN (QQ$) > QQ THEN QQ$ =  LEFT$ (QQ$,QQ)
2030   RETURN
29999   REM ********************************
30000   REM * SET UP FLAGS FOR USE IN R2000 *
30001   REM ********************************
30010   POKE  - 16368,0: HOME : VTAB 10: PRINT "IS THE PRINTER
        CONNECTED? Y/N ";: GET A$: PRINT A$: IF A$ = "N" THEN
        PR = 0: GOTO 30030
30020   PR = 1: PRINT D$;"PR#1": PRINT : PRINT " ": POKE  -
        12528,7: POKE  - 12526,83: POKE  - 12525,64: POKE  -
```

```
      12529,255: POKE  - 12524,0: PRINT : PRINT D$;"PR#0"
30030 HOME : VTAB 10: HTAB 1: POKE  - 16368,0: PRINT "IS THE
       PATIENT VENTILATED? Y/N ";: GET A$
30032 IF A$ = "Y" THEN IPPV$ = "IPPV": GOTO 30040
30034 IF A$ = "N" THEN IP$ = "SPONT": GOTO 30070
30036 GOTO 30032
30040 POKE  - 16368,0: VTAB 15: HTAB 1: PRINT "IS THE APPLE
      CONNECTED TO THE SERVO ?": VTAB 18: HTAB 18: PRINT "Y/N
      ";: FOR I = 1 TO 1000: NEXT : GET A$: IF A$ = "Y" THEN
      SERVO$ = "Y": GOTO 30070
30050 IF A$ < > "N" THEN 30040
30060 SE$ = "N"
30070 HOME : VTAB 10: HTAB 1: PRINT "IS THE EAR CO-OXIMETER
      CONNECTED? Y/N ";: GET A$: IF A$ = "Y" THEN EAR$ = "Y":
      GOTO 30100
30080 IF A$ = "N" THEN EAR$ = "N": GOTO 30150
30090 GOTO 30070
30100 HOME : VTAB 10: PRINT "ENTER HB AND TYPE <RETURN> ";:
      INPUT HB: IF HB < 5 OR HG > 20 THEN 30100
30110 VTAB 15: PRINT "ENTER %FIO2 AND TYPE <RETURN> ";: INPUT
      FO2: IF FO2 < 25 OR FO2 > 100 THEN 30110
30120 QQ$ =  STR$ (HB):QQ = 4: GOSUB 2000:HB$ = QQ$
30130 QQ$ =  STR$ (FO2):QQ = 3: GOSUB 2000:FO2$ = QQ$
30140 EAR$ = "Y" + HB$ + FO2$
30150 HOME : RETURN
```

```
*********
* R2000 *
*********
```

```
1 LOMEM: 28000: TEXT :DR = 30:D$ = CHR$ (4)::MENU$ = "Y":
  DIM SH(37),Z$(270): VTAB 10: ONERR GOTO 4
2 X$ = "SHUNTFILE": PRINT D$;"OPEN";X$: PRINT D$;"READ";X$:
  INPUT DR: INPUT PR: INPUT EAR$: INPUT FLAG$: INPUT IPPV$:
  INPUT SERVO$:REM * EAR$ AND IPPV$ DIFFERENT FROM ROCHE128 *
3 PRINT D$;"CLOSE"
4 POKE 216,0: GOSUB 360: VTAB 15: CALL - 868: VTAB 18: CALL
  - 868: GOTO 9
6 HTAB 1: CALL - 868: RETURN
7 HTAB (40 - LEN (Z$)) / 2: PRINT Z$: RETURN
9 PRINT D$;"BLOAD CHARGEN A$6000 L$100"
10 PRINT D$;"BLOAD COLOSSAL.SET A$6900 L$300"
11 DEF FN PLT(X) = VAL ( MID$ (Z$(X),Z0,Z1)): DEF FN
   TIME(X) = VAL ( MID$ (Z$(X),1,6))
13 PRINT D$;"OPEN NAMEFILE": PRINT D$;"READ NAMEFILE"
14 INPUT NA$: INPUT NO$: PRINT D$;"CLOSE NAMEFILE"
15 VTAB 20: HTAB 1: GOSUB 6
16 VTAB 20: HTAB 1: PRINT "NAME";: HTAB 15: PRINT NA$
18 ADC = - 16384 + (256) * 4:CHAN = ADC + 1
19 REM *******************************************
20 REM * OBTAIN CALIBRATION FACTORS FOR CHANNELS *
21 REM *******************************************
24 X$ = "R2000VARIABLES": PRINT D$;"OPEN";X$: PRINT
   D$;"READ";X$: FOR I = 1 TO 4: INPUT V$(I): NEXT
25 INPUT CRT: INPUT CSYS(1): INPUT CSYS(2): INPUT CSYS(3):
   INPUT CPUL(1): INPUT CPUL(2): INPUT CPUL(3): INPUT CCVP:
   INPUT CO2SA
26 INPUT ZHR: INPUT ZSYS(1): INPUT ZSYS(2): INPUT ZSYS(3):
   INPUT ZPUL(1): INPUT ZPUL(2): INPUT ZPUL(3): INPUT ZCVP:
   INPUT ZO2SA: PRINT D$;"CLOSE"
30 X$ = NA$ + " TEXTLEN": ONERR GOTO 32
31 PRINT D$;"OPEN";X$: PRINT D$;"READ"X$: INPUT LT: PRINT
```

```
    D$;"CLOSE": GOTO 33
32  POKE 216,0: PRINT D$;"OPEN";X$: PRINT D$;"WRITE"X$: PRINT
    LT: PRINT D$;"CLOSE"
33  VTAB 20: HTAB 1: GOSUB 6: PRINT :X$ = NA$ + " CVS
    DATALEN": ONERR  GOTO 50
34  PRINT D$;"OPEN";X$: PRINT D$;"READ";X$: INPUT FILE: PRINT
    D$;"CLOSE"
35  X$ = NA$ + " CVS DATA"
41  PRINT D$;"OPEN";X$
42  PRINT D$;"READ";X$
46  VTAB 20: HTAB 9: PRINT "OBTAINING FILE "
47  FOR I = 1 TO FILE: VTAB 20: HTAB 30: PRINT I: INPUT Z$(I)
49  NEXT
50  PRINT D$;"CLOSE": POKE 216,0
52  GOSUB 24000
55  IF FLAG$ <  > "" THEN 60
60  GOSUB 3000
62  Z$ = ""
70  TP$ =  MID$ (T$,7,2) + "." +  MID$ (T$,10,2):Z$ =  STR$ (
    INT (S / 60)): FOR II = 1 TO 6: IF  LEN (Z$) < 6 THEN Z$ =
    "0" + Z$: NEXT
71  QQ$ =  STR$ ( INT ( VAL (HR$) *  VAL (P1$) / 1E2)):QQ = 3:
    GOSUB 2000:RP$ = QQ$
72  P1 =  VAL (P1$):P2 =  VAL (P2$):P3 =  VAL (P3$)
73  P4 =  VAL (P4$):P5 =  VAL (P5$):P6 =  VAL (P6$)
74  REM ****************************************
75  REM * OBTAIN VENOUS & CAPILLARY SATURATION *
76  REM ****************************************
80  GOSUB 95: GOTO 98
95  GOSUB 96:QQ$ =  STR$ (SAO2):QQ = 4: GOSUB 2000:SA$ =
    QQ$:QQ$ =  STR$ (SVO2):QQ = 4: GOSUB 2000:SV$ = QQ$:
    RETURN
96  POKE CHAN,13:SA =  PEEK (ADC) / 2.36734: REM * EAR CO-OX *
97  POKE CHAN,9:SV =  PEEK (ADC) / 2.67: RETURN
98  IF P1 < 40 OR P1 > 250 THEN 60
99  IF EAR$ = "Y" THEN  GOSUB 31000
103  PRINT D$
104  PRINT D$;"PR#";PR
```

```
106  IF FLAG$ = "Y" THEN 126
107  GOSUB 24000: PRINT D$;"PR#";PR: FOR I = 1 TO 4: PRINT :
     NEXT : HTAB 1: PRINT NA$;: HTAB 20: PRINT NO$;: HTAB 31:
     PRINT D;"/";MT;"/";"1982"
108  HTAB 1: FOR I = 1 TO  LEN (NA$): PRINT "=";: NEXT : HTAB
     20: FOR I = 1 TO  LEN (NO$): PRINT "=";: NEXT : HTAB 31:
     PRINT "==========": PRINT : PRINT
110  XX$ = "  PULMONARY DATA"
115  HTAB 14: PRINT "SYSTEMIC DATA";: HTAB 35: PRINT "CARDIAC
     DATA"; SPC( 27): IF  LEFT$ (EAR$,1) = "Y" THEN  PRINT
     "SHUNT": GOTO 119
116  PRINT "SATN": REM * IF NO EAR OXIMETER *
119  PRINT
120  HTAB 2: PRINT "TIME";: HTAB 9: PRINT "HR";" SYS/DIA MEA
     CVP SVR"; SPC( 4);"C.O. SVL RPP"; SPC( 3);"SYS/DIA MEA
     PAWP  PVR"
125  FLAG$ = "Y"
126  PRINT D$;"PR#";PR
130  IF  VAL (VP$) =  - 99 THEN VP$ = "---"
140  IF  VAL (CO$) = 0 THEN 160
145  QQ$ = CO$:QQ = 4: GOSUB 2000:CO$ = QQ$
150  SV$ =  STR$ ( INT ( VAL (CO$) /  VAL (HR$) * 1E3))
155  QQ$ =  STR$ (( VAL (P3$) -  VAL (VP$)) /  VAL (CO$)):QQ =
     4: GOSUB 2000:SR$ = QQ$
158  QQ$ =  STR$ (( VAL (P6$) -  VAL (PW$)) /  VAL (CO$)):QQ =
     4: GOSUB 2000:PR$ = QQ$
160  IF PW$ = "-99" THEN VP$ = P7$
170  Z$ = Z$ + HR$ + P1$ + P2$ + VP$ + PW$ + P3$: IF  LEFT$
     (EAR$,1) = "Y" THEN Z$ = Z$ + Q$ + " SHUNT": GOTO 180
175  Z$ = Z$ + SV$ + "SATN": REM * IF NO EAR OXIMETER *
180  IF  VAL (CO$) = 0 THEN Z$ = Z$ + "EOF": GOTO 210
200  Z$ = Z$ + CO$ + SV$ + SR$ + PR$ + "EOF"
210  IF VP$ = "---" THEN VP$ = "-99"
220  IF PW$ = "-99" THEN PW$ = "---"
270  PRINT TP$;"  ";HR$;" ";P1$;"/";P2$;" ";P3$;" ";VP$;"
     ";SR$;
272  PRINT  SPC( 3);CO$;" ";SV$;" ";RP$; SPC( 3);P4$;"/";P5$;"
     ";P6$;" ";PW$;" ";PR$;
```

```
273  IF  LEFT$ (EAR$,1) = "Y" THEN  PRINT  SPC( 2);"QS=";Q$;:
     GOTO 276
274  PRINT  SPC( 4);SV$;
276  PW$ = "-99"
280  PRINT D$: PRINT D$;"PR#0"
285  IF Z$ = "" THEN 893
290  X$ = NA$ + " CVS DATALEN"
296  FI = FI + 1:Z$(FI) = Z$
298  PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$
299  PRINT FI: PRINT D$;"CLOSE";X$
302  X$ = NA$ + " CVS DATA"
310  PRINT D$;"OPEN";X$
316  PRINT D$;"POSITION";X$;",R";FI - 1
320  PRINT D$;"WRITE";X$
330  PRINT Z$
340  PRINT D$;"CLOSE";X$: IF FI < 150 THEN 345
341  GOSUB 929: REM * TEXT MODE
342  GOSUB 400: REM * WRITE VARS *
343  HOME : VTAB 10:Z$ = "WRITING 50 FILES TO DISC": GOSUB 7:
     PRINT D$"RUN FILEWRITE"
345  IF SH$ = "Y" THEN  GOSUB 400:SH$ = "R"
350  GOSUB 360: GOTO 370
360  PW$ = "-99":VP$ = PW$:SV$ = "---":CO$ = "----":SR$ =
     CO$:PR$ = CO$: RETURN
370  IF ME$ = "N" THEN  HOME : CALL 1002: PRINT D$;"PR#0":
     PRINT D$;"IN#0": TEXT : GOSUB 13000
380  GOTO 890
390  REM * WRITE SHUNT FILE *
400  X$ = "SHUNTFILE": PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$
405  PRINT DR: PRINT PR: PRINT EAR$: PRINT FLAG$: PRINT IPPV$:
     PRINT SERVO$
410  PRINT TP$: PRINT HR$: PRINT P1$: PRINT P2$: PRINT P3$:
     PRINT VP$: PRINT SR$:
420  PRINT CO$: PRINT SV$: PRINT RP$: PRINT P4$: PRINT P5$:
     PRINT P6$: PRINT PW$: PRINT PR$
430  PRINT Z$
440  PRINT D$;"CLOSE"
450  X$ = "NEXTPROG": PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$:
```

```
    PRINT "R2000": PRINT D$;"CLOSE"
460 RETURN
890 REM * TIMING ROUTINE *
891 IF VENT = 0 THEN VENT = S
892 S1 = S: HOME
893 GOSUB 24000:S2 = S:
894 IF S2 - S1 > DR * 60 THEN 60
895 IF SE$ = "Y" AND VENT + 60 * 60 <  = S2 THEN  GOSUB 400:
    PRINT D$;"RUN ROCHE/SERVO"
897 IF MENU$ = "N" THEN  GOSUB 14900
898 IF MENU$ = "Y" THEN  GOSUB 20000
900 REM * CHECK DAMPING/WEDGING *
910 GOSUB 3000
920 IF P3 = 0 THEN  VTAB 23: GOTO 893
925 IF (P1 - P2) > .2 * P3 THEN 942
928 GOSUB 929: GOTO 930
929 CALL 1002: PRINT D$;"PR#0": PRINT D$;"IN#0": TEXT :MENU$
    = "Y": RETURN
930 FOR I = 1 TO 10: PRINT "": NEXT
932 HOME : FLASH : VTAB 6: HTAB 10: PRINT "SYSTOLIC LINE
    DAMPED": VTAB 16: HTAB 9: PRINT "CHECK FOR BLOOD IN LINE"
935 VTAB 18: HTAB 5: PRINT "CHECK PRESSURE IN PERFUSION BAG"
940 VTAB 24: HTAB 8: NORMAL : PRINT "TYPE SPACEBAR TO RESET
    ";: GET A$: IF A$ <  > " " THEN 940
942 IF V$(2) <  > "PAP" THEN 972
955 IF P3 = 0 THEN  VTAB 23: GOTO 893
960 IF (P4 - P5) > .2 * P6 THEN  VTAB 23: GOTO 893
961 GOSUB 929
962 FOR I = 1 TO 10: PRINT "": NEXT
963 HOME : FLASH : VTAB 6: HTAB 4: PRINT "PULMONARY ARTERIAL
    LINE DAMPED": VTAB 8: HTAB 7: PRINT "CHECK FOR BLOOD IN
    LINE"
964 VTAB 14: HTAB 4: PRINT "*****************************"
965 VTAB 15: HTAB 4: PRINT "CHECK FOR WEDGING OF CATHETER"
966 VTAB 16: HTAB 4: PRINT "*****************************"
970 VTAB 23: HTAB 5: NORMAL : PRINT "TYPE SPACEBAR TO RESET
    ";: GET A$: IF A$ <  > " " THEN 970
972 FOR VV = 14 TO 16: GOSUB 975: NEXT : VTAB 23: GOTO 893
```

```
975  VTAB VV: HTAB 1: CALL  - 868: RETURN
2000  REM * STRING LENGTH ADJUST *
2010  IF  LEN (QQ$) < QQ THEN QQ$ = " " + QQ$: GOTO 2010
2020  IF  LEN (QQ$) > QQ THEN QQ$ =  LEFT$ (QQ$,QQ)
2030  RETURN
2995  REM **************************
3000  REM * OBTAIN VALUES FROM ADC *
3005  REM **************************
3010  Q = 8:CF = CRT:ZF = ZHR:QQ = 3: GOSUB 4000:HR$ = QQ$
3015  REM **********************
3020  REM * SYSTEMIC PRESSURES *
3025  REM **********************
3030  Q = 6:CF = CSYS(1):ZF = ZSYS(1): GOSUB 4000:P1$ = QQ$:
      REM * SYSTOLIC *
3040  Q = 4:CF = CSYS(2):ZF = ZSYS(2): GOSUB 4000:P2$ = QQ$:
      REM * DIASTOLIC *
3050  Q = 12:CF = CSYS(3):ZF = ZSYS(3): GOSUB 4000:P3$ = QQ$:
      REM * MEAN *
3055  REM ***********************
3056  REM * PULMONARY PRESSURES *
3057  REM ***********************
3070  Q = 2:CF = CPUL(1):ZF = ZPUL(1): GOSUB 4000:P4$ = QQ$:
      REM * SYSTOLIC *
3080  Q = 0:CF = CPUL(2):ZF = ZPUL(2): GOSUB 4000:P5$ = QQ$:
      REM * MEAN *
3090  IF IP$ = "IPPV" THEN P6$ = "999":P7$ = P6$: GOTO 3110
3095  P6$ = "-99":P7$ = P6$: REM * P6$ = MEAN PAP: P7$ = MEAN
      CVP*
3097  REM ********************************
3098  REM *         CVP/PA MEANS         *
3100  REM *------------------------------*
3101  REM * IF IPPV, TAKE SMALLEST VALUE *
3102  REM * IF SPONT, TAKE HIGHEST VALUE *
3105  REM ********************************
3110  FOR I = 1 TO 20:Q = 1:CF = CPUL(3):ZF = ZPUL(3): GOSUB
      4000: IF IP$ = "IPPV" AND QQ$ < P6$ THEN P6$ = QQ$
3111  IF IP$ = "SPONT" AND QQ$ > P6$ THEN P6$ = QQ$
3112  Q = 5:CF = CCVP:ZF = ZCVP: GOSUB 4000: IF IP$ = "IPPV"
```

```
     AND QQ$ < P7$ THEN P7$ = QQ$
3113  IF IP$ = "SPONT" AND QQ$ > P7$ THEN P7$ = QQ$
3114  NEXT
3115  IF V$(3) < > "CVP" THEN VP$ = "-99"
3117  IF V$(2) < > "PAP" THEN P4$ = "-99":P5$ = P4$:P6$ = P4$
3140  P1 = VAL (P1$):P2 = VAL (P2$):P3 = VAL (P3$)
3150  P4 = VAL (P4$):P5 = VAL (P5$):P6 = VAL (P6$)
3160  RETURN
3995  REM ************************************
4000  REM * READ ADC AND RETURN ACTUAL VALUE *
4005  REM ************************************
4010  POKE CHAN,Q:QQ$ = STR$ (( PEEK (ADC) - ZF) * CF): GOSUB
      2000: RETURN
9997  REM ***************************************
9998  REM * SHUNT AND CARDIAC OUTPUT SUBROUTINE *
9999  REM ***************************************
10000  HOME
10010  IF SH$ = "R" THEN HOME : VTAB 20: PRINT "LOAD SHUNT
       PROGRAM Y/N ";: GET A$: IF A$ = "Y" THEN POP : PRINT
       A$: VTAB 20: HTAB 1: PRINT "   LOADING SHUNT PROGRAM    ":
       PRINT D$;"RUN ROCHESHUNT"
10015  IF A$ = "" THEN POP : GOTO 20152
10020  IF SH$ = "R" THEN POP : GOTO 20152
10380  HOME : VTAB 4: HTAB 1: INPUT "CARDIAC OUTPUT L/MIN =
       ";CO$
10385  IF CO$ = "" THEN POP : GOTO 20152
10390  IF VAL (CO$) < 1.5 OR VAL (CO$) > 20 THEN 10380
10420  RETURN
12985  REM **************************
12990  REM * GRAPHICS TREND DISPLAY *
12995  REM **************************
13000  NI = FI:EX = 0:NH = 0:NM = 0:ND = 0: HOME :F2 = NI:PI =
       1: GOTO 13024
13001  HOME : PRINT "HOW FAR BACK DO YOU WISH TO START PLOT?":
       PRINT : PRINT "TYPE THE ELAPSED TIME": GOSUB 13003:NM =
       Z: GOTO 13012
13003  VTAB 5: PRINT "FOLLOWED BY THE UNITS AND <RETURN>":
       VTAB 8: PRINT "EG  10M=10MINS  5H=5HOURS  2D=2DAYS":
```

```
      VTAB 12: HTAB 19: INPUT TI$: IF  LEN (TI$) < 2 THEN
      PRINT : GOTO 13003
13004 HOME : IF  RIGHT$ (TI$,1) = "D" THEN Z =  VAL ( LEFT$
      (TI$, LEN (TI$) - 1)) * 1440: RETURN
13006 IF  RIGHT$ (TI$,1) = "H" THEN Z =  VAL ( LEFT$ (TI$,
      LEN (TI$) - 1)) * 60: RETURN
13008 IF  RIGHT$ (TI$,1) = "M" THEN Z =  VAL ( LEFT$ (TI$,
      LEN (TI$) - 1)): RETURN
13010 POP : GOTO 13001
13012 HOME : PRINT "TYPE TIME INTERVAL TO BE PLOTTED": GOSUB
      13003:Z9 = Z
13015 START =  FN TIME(FI) - NM
13017 FOR I = 1 TO FILE: IF START >  FN TIME(I) THEN  NEXT
13019 START =  FN TIME(I):PI = I
13021 LAST = ST + Z9: FOR I = FI TO 1 STEP  - 1: IF LAST <
      FN TIME(I) THEN  NEXT
13022 F2 = I
13023 LAST =  FN TIME(F2): GOTO 13025
13024 START =  FN TIME(PI):LAST =  FN TIME(FILE): HOME
13025 NM = LAST - START
13027 ET = ST: GOSUB 25000: GOSUB 26000:SD = DY:MS = MT:SH =
      H:SM = M
13028 ET = LA: GOSUB 25000: GOSUB 26000:LM$ =  STR$ (MT): IF
      LEN (LM$) < 2 THEN LM$ = "0" + LM$
13058 IF NM < 2 THEN  HOME : PRINT " NOT ENOUGH VALUES TO
      PLOT": FOR II = 0 TO 1000: NEXT : RETURN
13234 IF MINS = 9999 THEN BA = 30 + 212 / EXTRA
13250 REM    ** HI-RES PLOTS **
13260 GOSUB 13420
13262 GOTO 14700
13280 AX = 0:CX = 0:PP = PI
13281 IF BX = 0 THEN BA = 30: GOTO 13290
13282 BA = 30 + 212 / NM * (BX + CX): REM   * BASE ADDRESS *
13290 X1 = BA
13300 Y1 = 190 - Z3 *  FN PLT(PP) + Z2
13305 IF PP = FI OR PP = F2 THEN  RETURN
13310 IF  FN PLT(PP) < LO THEN PP = PP + 1:CX = CX + ( FN
      TIME(PP) -  FN TIME(PP - 1)): GOTO 13282
```

```
13312  IF Y1 < 0 THEN Y1 = 0
13314  IF X1 > 276 THEN X1 = 276
13316  IF X1 < 30 THEN X1 = 30
13320  FOR X = PP + 1 TO F2:Y2 = 190 - Z3 *  FN PLT(X) + Z2
13330  IF  FN TIME(X) <  =  FN TIME(X - 1) THEN X = X + 1:
       GOTO 13410
13349  X2 = X1 + 212 / NM * ( FN TIME(X) -  FN TIME(X - 1)) +
       AX
13374  IF  FN PLT(X) < LO THEN AX = AX + 212 / NM * ( FN
       TIME(X) -  FN TIME(X - 1)): GOTO 13410
13375  IF X2 > 276 THEN X2 = 276
13376  IF Y2 < 0 THEN Y2 = 0
13377  IF X2 < 30 THEN X2 = 30
13378  AX = 0
13379  IF Y2 > 189 THEN Y2 = 189
13380  FOR II = 0 TO LP: HPLOT X1 + II,Y1 + II TO X2 + II,Y2 +
       II: NEXT
13390  HPLOT X1 + 1,Y1 + 1 TO X2 + 1,Y2 + 1
13400  X1 = X2:Y1 = Y2
13410  NEXT X: RETURN
13420  GOTO 13430: REM * AXES PLOT *
13422  FOR I = 2 TO 7: IF N / I -  INT (N / I) < .01 THEN NX =
       2 * I: RETURN
13424  NEXT I:N = N + 1: GOTO 13422
13430  IF NM < 61 THEN NT$ = "MINS":N = NM: GOSUB 13422:BX = N
       - NM:NM = N: GOTO 13505
13450  IF NM < 2160 THEN BX = SM: IF NM + BX < 2160 THEN NT$ =
       "TIME": GOTO 13498
13470  NT$ = "DATE":BX = SH * 60 + SM
13490  ND =  INT ((NM + BX) / 1440) + 1
13492  N = ND: GOSUB 13422:ND = N:NM = ND * 1440
13493  NT$ = "DATE": GOTO 13505
13498  NH = (NM + BX) / 60: IF (NM + BX) / 60 -  INT ((NM +
       BX) / 60) <  > 0 THEN NH =  INT ((NM + BX) / 60) + 1
13500  N = NH: GOSUB 13422:NH = N:NM = NH * 60
13505  HGR2 : HCOLOR= 3: POKE 54,0: POKE 55,96
13506  FOR V = 0 TO NX:X = 30 + V * 212 / NX: HPLOT X,174 TO
       X,177: HPLOT X + 1,174 TO X + 1,177: NEXT
```

```
13510  FOR V = 0 TO NX STEP 2:X = 30 + V * 212 / NX: HPLOT
        X,174 TO X,180: HPLOT X + 1,174 TO X + 1,180: NEXT
13520  HPLOT 30,10 TO 30,180: HPLOT 31,10 TO 31,180
13525  IF NT$ = "MINS" THEN LX = SM:NZ = NM: FOR V = 0 TO NX
       STEP 2: VTAB 24:X = V * 212 / NX:A$ =  STR$ ( INT (LX +
       V * NZ / NX + .5)):HT =  INT (X / 6.975 + 5.5 -  LEN (A$)
       / 2): GOTO 13542
13530  IF NT$ = "TIME" THEN LX = SH:BX = SM:NZ = NH: GOTO
       13540
13532  LX = SD:BX = SH * 60 + SM:NZ = ND
13540  FOR V = 0 TO NX STEP 2: VTAB 24:X = V * 212 / NX:A$ =
       STR$ ( INT (LX + V * NZ / NX + .5)):HT =  INT (X /
       6.975 + 5.5  -  LEN (A$) / 2)
13542  A =  VAL (A$)
13544  IF NT$ = "DATE" THEN 13547
13545  IF NT$ = "TIME" AND A > 24 THEN A = A - 24: GOTO 13545
13546  IF NT$ = "MINS" AND A > 60 THEN A = A - 60
13547  GOTO 13553
13548  RESTORE : READ J:II = 0
13549  II = II + 1: READ J: IF II < MS THEN 13549
13550  IF A > J THEN A = A - J
13551  IF A > J THEN 13549
13553  HT = HT + 1
13554  HTAB HT: PRINT A: NEXT
13555  VTAB 23: HTAB 37: PRINT NT$;: IF NT$ = "DATE" THEN
       VTAB 24: HTAB 38: PRINT "/";LM$;
13600  RETURN
14700  REM * BP *
14710  JJ = 160: VTAB 2: PRINT JJ: VTAB 7: PRINT JJ - 40: VTAB
       12: HTAB 2: PRINT JJ - 80: VTAB 17: HTAB 2: PRINT JJ -
       120
14720  HPLOT 30,50 TO 242,50: HPLOT 30,90 TO 242,90
14730  HCOLOR= 5
14735  VTAB 6: HTAB 37: PRINT "BP": HPLOT 252,49 TO 268,49 TO
       268,48 TO 252,48
14750  Z0 = 10:Z1 = 3:Z2 =  - 20:Z3 = 1:L0 = 40:LP = 1: GOSUB
       13280
14770  Z0 = 13:L0 = 20: GOSUB 13280: REM  DIA
```

```
14790 REM * HR *
14800 HCOLOR= 6
14805 VTAB 10: HTAB 37: PRINT "HR": FOR II = 1 TO 3: HPLOT
      253,80 + II TO 272,80 + II: NEXT
14810 Z0 = 7:LP = 2: GOSUB 13280
14840 REM * CVP *
14850 HCOLOR= 1
14855 HPLOT 30,160 TO 242,160
14857 VTAB 14: HTAB 37: PRINT "CVP": FOR II = 0 TO 2: HPLOT
      252,116 + II TO 273,116 + II: NEXT
14858 VTAB 19: HTAB 3: PRINT "+5": VTAB 22: HTAB 3: PRINT
      "-5"
14860 Z0 = 16:Z1 = 3:Z2 =  - 30:Z3 = 3:L0 =  - 10: GOSUB
13280
14865 HCOLOR= 2: VTAB 18: HTAB 37: PRINT "PAWP": HPLOT
      252,146 TO 278,146 TO 278,147 TO 252,147
14870 Z0 = 19:LP = 1: GOSUB 13280: REM * PAWP *
14890 VTAB 1: HTAB 1::Z$ = "M(ENU  X(PAND  P(RINT": PRINT Z$
14895 VTAB 2: HTAB 32: PRINT  MID$ (T$,7,2);"."; MID$
      (T$,10,2);":"; MID$ (T$,13,2)
14900 FL$ = "Y":A = 0: GOSUB 21001
14902 POKE 54,0: POKE 55,96: CALL 1002
14904 FL$ = "Y"
14905 VTAB 1: HTAB 32: PRINT  MID$ (T$,7,2);"."; MID$
      (T$,10,2);":"; MID$ (T$,13,2)
14920 IF A = 205 THEN ME$ = "Y": GOTO 15030
15000 IF A = 208 THEN  PRINT D$;"PR#1": PRINT : PRINT : PRINT
      : PRINT  CHR$ (17): PRINT D$;"PR#0": GOTO 15020
15010 IF A = 216 THEN 15040
15020 A = 0:ME$ = "N": RETURN
15030 CALL 1002: PRINT D$;"PR#0": PRINT D$;"IN#0": TEXT :
      RETURN
15040 CALL 1002: PRINT D$;"PR#0": PRINT D$;"IN#0": TEXT :
      GOTO 13001
19997 REM ****************
19998 REM * DISPLAY MENU *
19999 REM ****************
20000 VTAB 1: HTAB 1:A = 0
```

```
20002  PRINT NA$;" / ";NO$
20005  VTAB 1: HTAB 32: PRINT  MID$ (T$,7,2);"."; MID$
       (T$,10,2);":";; MID$ (T$,13,2)
20007  V = 3
20010  VTAB V + 1: PRINT "1. ENTER PUL ART WEDGE PRESSURE  "
20020  VTAB V + 3: PRINT "2. ALTER CHANNELS / VENT MODE /
       HB:FIO2
20030  VTAB V + 5: HTAB 1: CALL  - 868: PRINT "3. ENTER TEXT
       COMMENT"
20035  VTAB V + 7: HTAB 1: PRINT "4. ENTER CARDIAC OUTPUT"
20040  VTAB V + 9: IF SH$ = "R" THEN  PRINT "5. ENTER SHUNT
       DATA": GOTO 20050
20045  PRINT "5. COLLECT CARDIAC AND SHUNT DATA"
20050  VTAB V + 11: PRINT "6. TREND DISPLAY"
20060  VTAB V + 13: PRINT "7. ALTER DATA RETRIEVAL TIME
       (=";DR;" MIN)"
20070  VTAB V + 15: PRINT "8. TAKE NEW READING / PRINT 24HR
       REPORT
20080  VTAB V + 17: HTAB 1: PRINT "9. MEASURE VENTILATION /
       QUIT"
20085  GOSUB 95: VTAB 23: HTAB 1: CALL  - 868: PRINT "TYPE NO.
       REQUIRED";
20086  REM **********************************************
20087  REM * PRINT VENOUS SATURATION IF NO EAR OXIMETER *
20088  REM **********************************************
20090  HTAB 21: PRINT "*";IP$;"*";: IF  LEFT$ (EAR$,1) <  >
       "Y" THEN : HTAB 30: PRINT "V SAT=";SV$;: GOTO 20100
20095  GOSUB 31000: VTAB 21: HTAB 33: PRINT "HB=";HB;"  ";:
       VTAB 22: HTAB 31: PRINT "FIO2=";FO2;" ";: VTAB 23: HTAB
       30: PRINT "QS/QT=";Q$;"%";: REM * PRINT QS/QT IF EAR$ =
       'Y' *
20100  GOSUB 21100
20105  IF A = 177 THEN  GOSUB 20158: GOTO 20153
20110  IF A = 178 THEN 22000
20112  IF A <  > 179 THEN 20128
20114  HOME :Z$ = "TYPE TEXT LABEL": GOSUB 7
20115  VTAB 5: PRINT "TYPE IN THE LABEL YOU WISH TO PRINT OUT
       FOLLOWED BY THE <RETURN> KEY.
```

```
20116  PRINT : PRINT : PRINT "USE THE LEFT ARROW KEY '<-' TO
       CORRECT  ANY ERRORS.
20117  PRINT : PRINT :Z$ = "TO RETURN TO MENU, TYPE <RETURN>
       KEY": GOSUB 7
20118  INPUT Z$: IF Z$ = "" THEN A = 0: HOME : RETURN
20119  GOSUB 24000:TP$ =  MID$ (T$,7,2) + "." +  MID$
       (T$,10,2)
20122  POKE 216,0: PRINT D$;"PR#";PR: PRINT : PRINT TP$; SPC(
       3);Z$:A = 0: PRINT D$;"PR#"0: HOME
20123  ZZ$ =  STR$ ( INT (S / 60)): FOR II = 1 TO 6: IF  LEN
       (ZZ$) < 6 THEN ZZ$ = "0" + ZZ$: NEXT
20124  Z$ = ZZ$ + Z$
20125  X$ = NA$ + " TEXTLEN":LT = LT + 1: PRINT D$;"OPEN";X$:
       PRINT D$;"WRITE";X$: PRINT LT: PRINT D$;"CLOSE"
20126  X$ = NA$ + " TEXT": PRINT D$;"OPEN";X$: PRINT
       D$;"POSITION";X$;",R";LT - 1
20127  PRINT D$;"WRITE";X$: PRINT Z$: PRINT D$;"CLOSE":Z$ =
       "":A = 0: HOME : RETURN
20128  IF A = 180 THEN  GOSUB 10380: GOSUB 20158: GOTO 20153
20130  IF A = 181 AND SH$ = "R" THEN : GOSUB 10000: GOSUB
       20158: GOTO 20153
20135  IF A = 181 THEN SH$ = "Y": GOSUB 10380: GOSUB 20158:
       GOTO 20153
20136  IF A <  > 182 THEN 20142
20137  HOME :Z$ = "TREND PLOT": GOSUB 7: VTAB 8: PRINT "1.
       COMPLETE PLOT": VTAB 12: PRINT "2. EXPANDED PLOT": VTAB
       16: PRINT "TYPE <RETURN> KEY IF NEITHER"
20138  VTAB 20: GET A$: IF A$ =  CHR$ (13) THEN  RETURN
20139  IF A$ = "1" THEN  GOSUB 13000: RETURN
20140  IF A$ = "2" THEN  GOSUB 13001: RETURN
20141  GOTO 20138
20142  IF A = 183 THEN  HOME : VTAB 10: PRINT "TYPE TIME
       REQUIRED BETWEEN MEASUREMENTS": PRINT : PRINT "IN
       MINUTES. THEN TYPE <RETURN> KEY": VTAB 20: HTAB 19: INPUT
       DR$: IF  VAL (DR$) < 1 OR  VAL (DR$) > 120 THEN 20142
20143  IF A = 183 THEN DR =  VAL (DR$): GOTO 20153
20145  IF A = 184 THEN 22200
20146  IF A <  > 185 THEN 20152
```

```
20147 HOME : VTAB 4: PRINT "1. MEASURE VENTILATION": VTAB 8:
      PRINT "2. QUIT PROGRAM": VTAB 12: PRINT "   TYPE
      <RETURN> IF NEITHER
20148 VTAB 20: GET A$: IF A$ =  CHR$ (13) THEN  RETURN
20149 IF A$ = "1" THEN SE$ = "Y": PRINT A$: GOSUB 400: PRINT
      D$;"RUN ROCHESERVO
20150 IF A$ = "2" THEN  HOME :Z$ = "OK FINISHED": VTAB 5:
      GOSUB 7:Z$ = "TO RESTART TYPE THE WORD 'RUN'": VTAB 8:
      GOSUB 7:Z$ = "THEN PRESS THE <RETURN> KEY": VTAB 11:
      GOSUB 7: END
20151 GOTO 20148
20152 A = 0: RETURN
20153 POP : GOTO 60
20157 REM * PAWP *
20158 HOME : HTAB 1: PRINT "INFLATE PULMONARY ARTERY BALLOON"
20160 VTAB 6: INPUT "ENTER PAWP AND TYPE <RETURN> ";QQ$
20170 IF  VAL (QQ$) <  - 10 OR  VAL (QQ$) > 35 THEN 20160
20180 QQ = 3: GOSUB 2000:PW$ = QQ$
20200 REM * GET CVP *
20230 HOME : VTAB 5:Z$ = "TYPE IN CVP IN MM HG AT
      MID-AXILLARY": GOSUB 7: PRINT :Z$ = "LINE THEN TYPE
      <RETURN>  KEY.": GOSUB 7
20232 VTAB 12:Z$ = "IF CVP UNAVAILABLE, PRESS <RETURN>":
      GOSUB 7: PRINT : PRINT : INPUT VP$
20235 IF VP$ = "" THEN QQ$ =  STR$ ( - 99): GOTO 20270
20237 CV =  VAL (VP$)
20240 IF CV <  - 10 OR CV > 30 THEN 20230
20260 QQ$ =  STR$ (CV)
20270 QQ = 3: GOSUB 2000:VP$ = QQ$
20300 RETURN
21001 REM * TEST KEY PRESS *
21100 A =  PEEK ( - 16384): POKE  - 16368,0: RETURN
22000 HOME : VTAB 1:Z$ = "RE-CALIBRATE": GOSUB 7
22004 REM *******************************************
22005 REM * ALTER CHANNELS/VENTLATION MODE/FIO2/HB *
22006 REM *******************************************
22010 VTAB 3: PRINT "1. RE-CALIBRATE PRESSURE TRANSDUCERS
22015 VTAB 5: PRINT "2. RE-CALIBRATE SERVO VENTILATOR
```

```
22017  VTAB 7: PRINT "3. CHANGE VENTILATION MODE
22020  VTAB 9: PRINT "4. ENTER CALIBRATION FOR OXIMETRIX
22022  VTAB 11: PRINT "5. ENTER NEW HB OR FIO2
22025  VTAB 22:Z$ = "TYPE <RETURN> KEY IF NONE": GOSUB 7
22027  VTAB 24: HTAB 1: GET A$: IF A$ = "1" THEN  VTAB 3:
       PRINT : GOSUB 400: PRINT D$;"RUN R2000CALIBR"
22029  IF A$ = "2" THEN  VTAB 3: PRINT : GOSUB 400: PRINT
       D$;"RUN SERVOCALIBRATE"
22030  IF A$ = "3" THEN  IF IPPV$ = "IPPV" THEN IP$ = "SPONT":
       HOME : GOTO 20152
22035  IF A$ = "3" THEN IP$ = "IPPV": HOME : GOTO 20152
22036  REM *******************************
22037  REM * ALTER OXIMETRIX CALIBRATION *
22038  REM *******************************
22039  IF A$ = "4" THEN  HOME : VTAB 5: INPUT "ENTER
       SATURATION ";TSAN: GOSUB 96:CSVO2 = TS / SVO2: HOME :
       GOTO 20152
22040  IF A$ = "5" THEN  HOME : VTAB 5: INPUT "ENTER HB ";HB:
       IF HB < 5 OR HB > 20 THEN 22040
22042  IF A$ = "5" THEN : VTAB 10: INPUT "ENTER %FIO2 ";FO2:
       IF FO2 < 20 OR FO2 > 100 THEN 22042
22043  REM ******************************
22044  REM * HB AND FIO2 STORED IN EAR$ *
22044  REM ******************************
22047  QQ$ =  STR$ (HB):QQ = 4: GOSUB 2000:HB$ = QQ$:QQ$ =
       STR$ (FO2):QQ = 3: GOSUB 2000:FO2$ = QQ$:EAR$ = "Y" +
       HB$ + FO2$
22048  IF A$ = "5" THEN 20152
22049  IF A$ =  CHR$ (13) THEN  HOME : GOTO 20152
22050  GOTO 22030
22110  IF CVP$ = "N" THEN CVP$ = "Y": GOTO 20153
22112  CVP$ = "N": GOTO 20153
22115  IF PT$ = "SYSCHAN1" THEN PT$ = "PULCHAN1": GOTO 20153
22116  PT$ = "SYSCHAN            1": GOTO 20153
22200  HOME : VTAB 8: PRINT "1. TAKE NEW READING": VTAB 10:
       PRINT "2. PRINT 24HR CVS/RESP REPORT": VTAB 12: PRINT
       "3. PRINT 24HR TEXT REPORT": VTAB 18: PRINT "TYPE <RETURN>
       IF NONE OF THESE"
```

```
22210  VTAB 22: GET A$: PRINT A$: IF A$ =  CHR$ (13) THEN
       RETURN
22220  IF A$ = "1" THEN 20153
22230  IF A$ = "2" THEN  GOSUB 400: PRINT D$;"RUN ROCHEPRINT"
22240  IF A$ = "3" THEN  GOSUB 400: PRINT D$;"RUN
       ROCHEPRINTEXT"
24000  REM *****************
24005  REM * GET DATE/TIME *
24006  REM *****************
24010  D$ =  CHR$ (4): PRINT D$:SLOT = 3
24030  PRINT D$;"IN#";SL: PRINT D$;"PR#";SL: INPUT T$: PRINT
       D$;"IN#0": PRINT D$;"PR#0"
24040  MT =  VAL ( MID$ (T$,1,2))
24050  D =  VAL ( MID$ (T$,4,2))
24060  H =  VAL ( MID$ (T$,7,2))
24070  M =  VAL ( MID$ (T$,10,2))
24080  S =  VAL ( MID$ (T$,13,6))
24090  RESTORE
24100  DTD = 0
24110  FOR I = 1 TO MT: READ J:DTD = DTD + J: NEXT
24120  DATA  0,31,28,31,30,31,30,31,31,30,31,30,31
24130  DTD = DTD + D
24140  IF MT > 2 AND L = 1 THEN DTD = DTD + 1
24150  STD = DTD * 86400 + H * 3600 + M * 60 + S
24160  S = STD
24170  RETURN
25000  REM ******************
25010  REM * MINS->DAYS/HRS *
25020  REM ******************
25040  DY =  INT (ET / 1440)
25050  ET = ET - DY * 1440
25060  H =  INT (ET / 60)
25070  M = ET - H * 60
25080  RETURN
26000  REM **************
26005  REM * DAYS->DATE *
26006  REM **************
26010  IF L = 1 AND DY = 60 THEN MT = 2:DY = 29: GOTO 26070
```

```
26015  IF L = 1 AND DY > 60 THEN DY = DY - 1
26017  RESTORE
26018  READ J
26020  FOR II = 1 TO 12: READ J:MT = MT + 1
26030  IF DY - J < 1 THEN ND = DY: GOTO 26070
26035  DY = DY - J
26040  NEXT
26070  RETURN
30999  REM *****************************
31000  REM * ON-LINE QS/QT CALCULATION *
31001  REM *****************************
31004  HB$ =  MID$ (EAR$,2,4):HB =  VAL (HB$)
31005  FO2$ =  MID$ (EAR$,6,3):FO2 =  VAL (FO2$)
31006  IF SA$ = "" OR SV$ = "" THEN  RETURN
31010  PB = 760:F = FO2 / 100:F = (PB - 47) * F - 40 * (F + (1
       - F) / .8): REM * PO2 END CAPO2 *
31015  CPO2 = HB * 100 * 1.34 / 100 + .0031 * F:
       REM * CAPILLARY O2 CONTENT *
31020  CAO2 = HB *  VAL (SA$) * 1.34 / 100 + .0031 * 100:
       REM * ARTERIAL O2 CONTENT *
31030  CVO2 = HB *  VAL (SV$) * 1.34 / 100 + .0031 * 40:
       REM * VENOUS O2 CONTENT *
31040  Q =  INT ((CPO2 - CAO2) * 100 / (CPO2 - CVO2) + .5)
31050  QQ$ =  STR$ (Q):QQ = 2: GOSUB 2000:Q$ = QQ$: REM *
       QS/QT *
31060  RETURN
```

```
             *************
             * LOAD.OBJ0 *
             *************

        ORG $300
START   LDA $C0D0       INITIALISE ROCHE 128 SERIAL INTERFACE
        AND #$02        CARD
        BEQ START
        LDA #$05        LOAD 'A' REGISTER WITH TRANSMIT
        STA $C0D1       CHARACTER 05 AND SEND TO SERIAL CARD
        LDY #$00
        LDX #$00
        NOP
        NOP
REPEAT  INX             BETWEEN RECEPTION OF EACH CHARACTER
        CPX #$FA        INCREMENT 'X' AND COMPARE WITH $FA
        BCS TIMEOUT     IF GREATER THEN GO TO 'TIMEOUT'
        NOP
        LDA #$0A        DELAY LOOP
        JSR DELAY
        NOP
        LDA $C0D0       TEST FOR ARRIVAL OF NEW CHARACTER
        LSR A           AND BRANCH TO 'REPEAT' IF NONE
        BCC REPEAT
        LDA $C0D1       GET NEW CHARACTER AND CHECK FOR END OF
        CMP #$04        TRANSMISSION CHARACTER 04
        BNE STORE
        STA $380,Y
        LDA #$00        ZERO STORED IN $370 TO INDICATE DATA
        STA $370        SUCCESSFULLY TRANSFERED
        RTS             RETURN TO BASIC
        NOP
        NOP
STORE   STA $380,Y
        LDX #$00        ZERO 'X' TO RE-COMMENCE COUNT FOR
        INY             TIMEOUT
```

```
        CLC
        BCC REPEAT
        NOP
        NOP
TIMEOUT LDA #01         TIMEOUT ERROR: 01 LOADED INTO
        STA $370        $370 TO INDICATE ERROR
        RTS             RETURN TO BASIC
        NOP
        NOP
DELAY   SEC
SECOND  PHA
FIRST   SBC #$01
        BNE FIRST
        PLA
        SBC #$01
        BNE SECOND
        RTS
```

```
1  REM **************
2  REM * ROCHESERVO *
3  REM **************
4  LOMEM: 32100:D$ =  CHR$ (4):DR = 30: ONERR  GOTO 20
5  ADC =  - 16384 + (256 * 4):CHAN = ADC + 1: REM * ADC SLOT 4
   *
6  GOTO 15
7  HTAB (40 -  LEN (Z$)) / 2: PRINT Z$;: RETURN
15  X$ = "SHUNTFILE": PRINT D$;"OPEN";X$: PRINT D$;"READ";X$:
    INPUT DR: INPUT PR: INPUT CV$: INPUT FLAG$: INPUT PT$:
    INPUT SERVO$: PRINT D$;"CLOSE": GOTO30
20  ROCHE$ = "N": ONERR GOTO 30: REM * NOT CALLED FROM ROCHE *
25  X$ = "RESPFILE": PRINT D$;"OPEN";X$: PRINT D$;"READ";X$:
    INPUT DR: INPUT PR: INPUT FLAG$
30  POKE 216,0: PRINT D$;"CLOSE": TEXT : GOSUB 40: GOTO 45
40  HOME :Z$ = "GLASGOW ROYAL INFIRMARY": GOSUB 7: PRINT :
    PRINT :Z$ = "RESPIRATORY INTENSIVE CARE UNIT": GOSUB 7:
    PRINT: PRINT : RETURN
45  VTAB 20:Z$ = "ASSEMBLER DATA COLLECTION": GOSUB 7: PRINT
46  REM ********************************************
47  REM * CHECK FOR CONNECTION TO SERVO VENTILATOR *
48  REM ********************************************
50  I = I + 1: POKE CHAN,7: IF  PEEK (ADC) < 25 AND I < 400
    THEN 50: REM INSP
55   J = J + 1: POKE CHAN,7: IF  PEEK (ADC) > 200 AND J < 400
    THEN 55: REM EXP
60  I = I + 1: POKE CHAN,7: IF  PEEK (ADC) < 25 AND I < 400
    THEN 60
70  J = J + 1: POKE CHAN,7: IF  PEEK (ADC) > 200 AND J < 400
    THEN 70
80  IF I > 30 AND I < 400 AND J > 30 AND J < 400 THEN 110: REM
    * SERVO CONNECTED *
85  IF RO$ <  > "N" THEN 97:REM * RETURN TO CVS PROG *
90  VTAB 12: Z$ = "SERVO NOT CONNECTED": FLASH : GOSUB 35:
    PRINT  CHR$ (7); CHR$ (7): VTAB 15:Z$ = "CONNECT SERVO AND
    TYPE <SPACEBAR>: GET A$: IF A$ <  > " " THEN 90
92  PRINT : NORMAL
95  RUN
```

```
96  REM ******************************************
97  REM * RETURN TO MAIN CARDIOVASCULAR PROGRAM *
98  REM ******************************************
99  X$ = "NEXTPROG": PRINT D$;"OPEN";X$: PRINT D$;"READ";X$:
    INPUT N$: PRINT D$;"CLOSE"
100  X$ = "SHUNTFILE": PRINT D$;"OPEN";X$: PRINT
     D$;"WRITE";X$: PRINT DR: PRINT PR: PRINT CV$: PRINT FLAG$:
     PRINT PT$: PRINT "N": PRINT D$;"CLOSE"
105  PRINT D$;"RUN";N$
110  PRINT D$;"NOMONI,C,O": DIM Z$(200)
115  NP = 2: REM  DIV FAC
120  PRINT D$;"BLOADCHARGEN A$6000 L$100,D1"
125  PRINT D$;"BLOAD COLOSSAL.SET A$6900 L$300"
130  PRINT D$;"BLOAD RESP.OBJ0"
135  DEF  FN PLT(X) =  VAL ( MID$ (Z$(X),Z0,Z1)): DEF  FN
     TIME(X) =  VAL ( MID$ (Z$(X),1,6))
140  GOTO 3000: REM * OBTAIN PREVIOUS DATA STRINGS *
200  POKE 27798,0: CALL 31000
210  IF  PEEK (27798) <  > 34 THEN 225
212  GOSUB 2500
215  HOME : VTAB 10: FLASH :Z$ = "ARRAY EXCEEDED": GOSUB 7:
     GOSUB 220: GOTO 200
220  FOR I = 1 TO 3: PRINT "": NEXT
223  VTAB 15:Z$ = "?  DISCONNECTED": GOSUB 3020: NORMAL : VTAB
     24: HTAB 8: PRINT "TYPE SPACEBAR TO RESET ";: GOSUB
     21100: IF AA = 160 THEN  HOME : RETURN
224  REM ********************************
225  REM * START AND END OF INSP ARRAY *
226  REM ********************************
230  L1 = 27800
240  L2 =  PEEK (27776) * 256 +  PEEK (27777) + 9 * 16 + 8
242  IC = L2 - L1
250  REM ******************************
252  REM * START AND END OF EXP ARRAY *
255  REM ******************************
260  L3 = L2 + 1
262  L4 =  PEEK (27778) * 256 +  PEEK (27779) + 9 * 16 + 8
264  EC = L4 - L3
```

```
275  T1$ = "":T2$ = "":T3$ = ""
276  REM **************************
277  REM * GET INSP AND EXP TIMES *
278  REM **************************
280  FOR I = 0 TO 5:T1$ = T1$ +  CHR$ ( PEEK (27780 + I) -
     128): NEXT
290  FOR I = 0 TO 5:T2$ = T2$ +  CHR$ ( PEEK (27786 + I) -
     128): NEXT
295  FOR I = 0 TO 5:T3$ = T3$ +  CHR$ ( PEEK (27792 + I) -
     128): NEXT
305  T1 =  VAL (T1$):T2 =  VAL (T2$):T3 =  VAL (T3$)
307  IF T2 < T1 THEN T1 = T1 - 60
308  IF T3 < T2 THEN T3 = T3 + 60
310  TI = T2 - T1:TE = T3 - T2
315  RTE =  INT (600 / (TI + TE) + .5) / 10: GOTO 318
316  FOR I = 1 TO IC:FI(I) = FI(I) * (PI(I) + 559) * SP / 559
318  CI = TI / IC * NP:CE = TE / EC * NP
320  GOSUB 2473: REM * KEYTEST *
340  IF TR$ = "Y" AND RPLT = 0 THEN  GOSUB 13000
350  IF XP$ = "Y" AND RPLT = 0 THEN  GOSUB 13001
360  IF MENU$ = "Y" THEN  GOSUB 20000
370  AA = 0:X = 0
440  IF PLTFLAG$ = "Y" AND RPLT = 1 THEN  GOSUB 2400
450  IF PLTFLAG$ = "Y" AND RPLT = 0 THEN  GOSUB 2040
800  REM ****************
801  REM * CALCULATIONS *
802  REM ****************
803  IW = 0:EW = 0:PP = 0:IV = 0:VI = 0:EV = 0:CO = 0:IR =
     0:IP = 0:EM = 0:ZZ = 0:YY = 0
810  IF PL$ < > "Y" AND TR$ < > "Y" AND XP$ < > "Y" THEN
     VTAB 24:Z$ = "   CALCULATING RESULTS   ": GOSUB 7
814  REM ********************
815  REM * INSP CALCULATIONS *
816  REM ********************
820  F = 27799:P = F + 1
830  FOR I = 1 TO IC STEP 2 * NP
835  VI = VI +  PEEK (F + I):IW = IW +  PEEK (F + I) *  PEEK
     (P + I)
```

```
837  IF  PEEK (P + I) > PP THEN PP =  PEEK (P + I)
840  NEXT
845 VI =  INT (VI * SI * CI + .5)
850 IW = IW * SP * SI * CI
852 IW =  INT (.6 * IW / (TI + TE) + .5) / 1E3
855 PP =  INT (PP * SP + .5)
856 PAUSEP =  INT ( PEEK (P + I - 2) * SP + .5)
857  REM *******************
858  REM * EXP CALCULATIONS *
859  REM *******************
860 F = 27799 + IC:P = F + 1
870  FOR E = 1 TO EC STEP 2 * NP
880 EV = EV +  PEEK (F + E):EW = EW +  PEEK (F + E) *  PEEK (P
    + E): NEXT
890 EV =  INT (EV * SE * CE + .5):EW = EW * SP * SE * .00001 *
    CE
892 EM =  INT (EV * 60 / (TI + TE) + .5) / 1E3
900 PEEP =  INT ( PEEK (P + EC - 2) * SP + .5)
910  IF PA - PE = 0 THEN 930
920 COMP =  INT (10 * VI / (PAUSEP - PEEP) + .5) / 10
930 YY =  INT ((PE * VI + .5 * VI * (PP - PE)) * 60 / 1E5 /
    (TI + TE) * 1E3 + .5) / 1E3
940 IRES =  INT ((IW - YY) * 1E3 + .5) / 1E3:IPERCENT =  INT
    ((IW - YY) / IW * 1000) / 10
989  IF TR$ = "Y" OR PL$ = "Y" OR XP$ = "Y" THEN 2600
990  IF ME$ = "Y" THEN 1180
995  TEXT
997 REM ****************
998 REM * DISPLAY TABLE *
999 REM ****************
1000  IF BR = 0 THEN  HOME
1002  VTAB 1: HTAB 1: PRINT NA$;" / ";NO$;"    ";BR + 1
1003  GOSUB 24000: VTAB 1: HTAB 32: PRINT  MID$ (T$,7,2);".";
      MID$ (T$,10,2);":"; MID$ (T$,13,2)
1010  VTAB 4: HTAB 1: PRINT "INSP VOL=";VI;" ML  ": PRINT
1018  PRINT "EXP. VOL=";EV;" ML  ": PRINT
1020  PRINT "MIN. VOL=";EM;" L    "
1036  VTAB 12: HTAB 1: PRINT "E/I RATIO="; INT (100 * TE / TI
```

```
     + .5) / 100;"    ";
1040 VTAB 16: HTAB 1: PRINT "COMPLIANCE   = ";COMP" ML/CM H2O   "
1120 VTAB 4: HTAB 21: PRINT "PAUSE P=";PA;" CM H2O  "
1130 VTAB 6: HTAB 21: PRINT "PEAK P =";PP;" CM H2O  "
1140 VTAB 8: HTAB 21: PRINT "PEEP   =";PE;" CM H2O  "
1150 VTAB 12: HTAB 21: PRINT "RATE=";RTE;" /MIN    "
1158 IF IW = 0 THEN 200
1160 VTAB 18: PRINT "INSPIR WORK= ";IW;" KG.M/MIN "
1170 VTAB 20: PRINT "INSP RES WK= ";IR;" KG.M/MIN  ";" =
     ";IP;"%";"   ";
1180 VTAB 24:Z$ = "V(OL/PRES  G(RAPH  M(ENU": GOSUB 7
1190 GOTO 2600
2000 REM * ADJ LEN *
2010 FOR I = 1 TO QQ: IF  LEN (QQ$) < QQ THEN QQ$ = " " +
     QQ$: NEXT
2020 IF  LEN (QQ$) > QQ THEN QQ$ =  LEFT$ (QQ$,QQ)
2025 PRINT QQ$;"   ";
2030 Z$ = Z$ + QQ$
2035 RETURN
2040 REM *************
2041 REM * AXES PLOT *
2042 REM *************
2070 GOSUB 2135: GOTO 2190
2090 FOR J = 1 TO  LEN (A$):L = 32:L$ =  MID$ (A$,J,1): IF L$
     = "@" OR L$ = "&" THEN J = J + 1:L = 0 - 64 * (L$ = "&")
2100 I =  ASC ( MID$ (A$,J)): IF I <  ASC ("A") OR I >  ASC
     ("Z") THEN L = 0
2110 PRINT  CHR$ (I + L);: NEXT : PRINT : RETURN
2134 REM **************
2135 REM * SCALE AXES *
2136 REM **************
2140 IF VI < 500 THEN HY = 500
2141 IF VI > 500 THEN HY = 1000
2142 IF VI > 1000 THEN HY = 1500
2145 LY = 0:LX = 0:HX = 40:NX = 4
2150 IF PP < 22 THEN HX = 20
2151 IF PP < 45 THEN HX = 40
2152 IF PP > 45 THEN HX = 60
```

```
2168  SS = 160 / HY: RETURN
2190  X$ = "@PRES"
2210  Y$ = "@VOL"
2230  HOME : HGR2 : HCOLOR= 3: POKE 54,0: POKE 55,96
2250  HPLOT 37,0 TO 37,175 TO 250,175
2252  HPLOT 38,0 TO 38,175 TO 250,175
2260  FOR V = 0 TO NY:Y = 175 - 175 * V / NY: HPLOT 33,Y TO
      37,Y: NEXT
2270  FOR V = 0 TO NX:X = 37 + V * 200 / NX: HPLOT X,175 TO
      X,179: HPLOT X + 1,175 TO X + 1,179: NEXT
2280  FOR V = 0 TO NY STEP 2:Y = 175 - 175 * V / NY:A$ =  STR$
      (LY + V * (HY - LY) / NY): VTAB (Y / 8 + 1): HTAB (5 -
      LEN (A$)): GOSUB 2090: HPLOT 30,Y TO 37,Y: NEXT
2283  LX = 0
2290  FOR V = 0 TO NX STEP 2: VTAB 24:X = 37 + V * 200 / NX:A$
      =  STR$ (LX + V * (HX - LX) / NX):HT = 13 + (X - 80) *
      28 /  200 -  LEN (A$) / 2: IF HT +  LEN (A$) > 40 THEN HT
      = 41 - LEN (A$)
2300  HTAB HT: GOSUB 2090: HPLOT X,175 TO X,182: HPLOT X +
      1,175 TO X + 1,182: NEXT
2310  VTAB 1: HTAB 7:A$ = ""
2320  A$ = Y$: GOSUB 2090
2340  VTAB 21: HTAB 41 -  LEN (X$):A$ = X$: GOSUB 2090
2400  REM ************
2401  REM * V/P PLOT *
2402  REM ************
2403  A$ = "@M(ENU @T(ABLE @G(RAPH @V(OL/@P(RES": VTAB 1: HTAB
      11: GOSUB 2090
2404  IV = 0:VI = 0:PP = 0: VTAB 18: HTAB 30: PRINT
      "RATE=";RTE;"   "
2406  SV(1) = .9643 * 1000 / HY:SV(2) = SV(1) * SE / SI:PS =
      .2558 * 4.9 * 40 / HX
2407  X1 = 37:Y1 = 175
2408  SV = CI * SV(1):F = 27799:P = F + 1
2410  FOR I = 1 TO L2 - L1 STEP 2 * NP
2415  IV = IV +  PEEK (F + I) * SV
2416  IF  PEEK (P + I) > PP THEN PP =  PEEK (P + I)
2417  X2 = 37 +  PEEK (P + I) * PS:Y2 = 175 - IV
```

```
2422  GOSUB 2466
2424  HPLOT X1,Y1 TO X2,Y2
2425  X1 = X2:Y1 = Y2
2430  NEXT I
2432  VI = IV / SV(1) * SI:PP = PP * SP
2437  ZZ = HY:XX = HX: GOSUB 2135: IF ZZ <  > HY OR XX <  > HX
      THEN 2040
2439  F = L2 + 1:P = F + 1:SV = SV(2) * CE
2440  FOR E = 1 TO L4 - L3 STEP 2 * NP
2441  IV = IV -  PEEK (F + E) * SV
2444  X2 = 37 +  PEEK (P + E) * PS:Y2 = 175 - IV
2448  GOSUB 2466
2452  HPLOT X1,Y1 TO X2,Y2
2455  X1 = X2:Y1 = Y2
2457  NEXT
2464  RP = 1: RETURN
2465  REM * RANGE CHECK *
2466  IF X2 < 0 THEN X2 = 0
2467  IF X2 > 276 THEN X2 = 276
2468  IF Y2 < 0 THEN Y2 = 0
2469  IF Y2 > 185 THEN Y2 = 185
2470  RETURN
2470  REM *********************
2471  REM * TEST FOR KEYPRESS *
2472  REM *********************
2474  GOSUB 21100: IF X = 212 OR X = 177 THEN PLT$ = "N":TR$ =
      "N":XP$ = "N":ME$ = "N":RP = 0: GOSUB 2500: REM * TABLE *
2475  IF X = 214 OR X = 178 THEN XP$ = "N":PL$ = "Y":TR$ =
      "N":ME$ = "N":RP = 0: REM * NEW V/P PLOT *
2476  IF X = 216 THEN XP$ = "Y":RPLT = 0:TR$ = "N":PL$ =
      "N":ME$ = "N": GOSUB 2500: REM * EXPAND GRAPH *
2477  IF X = 199 OR X = 182 THEN TR$ = "Y":RPLT = 0:XP$ =
      "N":ME$ = "N":PL$ = "N": REM * TREND GRAPH *
2480  IF X = 205 THEN ME$ = "Y":XP$ = "N":PLT$ = "N":TR$ =
      "N": GOSUB 2500: REM  * DISPLAY MENU *
2485  X = 0: RETURN
2500  CALL 1002: PRINT D$;"PR#0": PRINT D$;"IN#0": TEXT : HOME
      : RETURN: REM * RE-SET DOS POINTERS FOR TEXT MODE *
```

```
2579 REM *******************************
2580 REM * TIMING & AVERAGE SUBROUTINE *
2581 REM *******************************
2590 GOSUB 24000:S1 = S
2600 VTAB 2: GOSUB 24000:S2 = S: IF S2 - S1 < DR * 60 THEN
     200
2602 IF RTE < 4 OR RTE > 35 THEN 200
2605 BRETHCOUNT = BR + 1
2610 SM(1) = SM(1) + RTE
2612 SM(2) = SM(2) + VI
2615 SM(3) = SM(3) + EV
2617 SM(4) = SM(4) + EM
2620 SM(5) = SM(5) + PP
2625 SM(6) = SM(6) + PEEP
2630 SM(7) = SM(7) + COMP
2632 SM(8) = SM(8) + IW
2633 SM(9) = SM(9) + IRES
2635 IF BR = 10 THEN  VTAB 20: GOSUB 2500: GOTO 2638
2637 GOTO 200
2638 GOSUB 24000:Z$ =  STR$ ( INT (S / 60))
2640 IF  LEN (Z$) < 6 THEN Z$ = "0" + Z$: GOTO 2640
2641 FOR I = 1 TO 7:X$(I) =  STR$ ( INT (SM(I) / BR + .5)):
     NEXT
2642 X$(4) =  STR$ ( INT (SM(4) / BR * 1E2 + .5) / 1E2)
2643 X$(8) =  STR$ ( INT (SM(8) / BR * 1E3 + .5) / 1E3)
2644 X$(9) =  STR$ ( INT (SM(9) / BR * 1E3 + .5) / 1E3)
2646 REM **************************
2647 REM * PRINT AVERAGED RESULTS *
2648 REM **************************
2649 PRINT D$;"PR#";PR: IF FLAG$ = "Y" AND RO$ = "N" THEN
     2649
2650 IF FLAG$ = "Y" THEN 2648
2651 PRINT : PRINT : PRINT NA$;: HTAB 20: PRINT NO$;: HTAB
     31: PRINT D;"/";MT;"/1982":FL$ = "Y"
2652 PRINT : PRINT " TIME  RATE  IN VT  EX VT  V MIN  PEAK
     PEEP  COMP  WORK  RESIST"
2653 PRINT  MID$ (T$,7,2) + "." + MID$ (T$,10,2);"   ";
2655 QQ = 2:QQ$ = X$(1): GOSUB 2000
```

```
2656  QQ = 4:QQ$ = X$(2): GOSUB 2000
2657  QQ$ = X$(3): GOSUB 2000
2668  QQ = 5:QQ$ = X$(4): GOSUB 2000
2669  QQ = 2:QQ$ = X$(5): GOSUB 2000
2670  QQ$ = X$(6): GOSUB 2000: PRINT " ";
2677  QQ$ = X$(7): GOSUB 2000
2679  QQ = 5:QQ$ = X$(8): GOSUB 2000
2680  QQ = 5:QQ$ = X$(9): GOSUB 2000: PRINT
2685  PRINT D$;"PR#0"
2695  BR = 0: FOR I = 1 TO 10:SM(I) = 0: NEXT
2700  REM ********************************
2702  REM * WRITE AVERAGED DATA TO DISC *
2704  REM ********************************
2710  X$ = NA$ + " RS DATALEN"
2720  FI = FI + 1:Z$(FI) = Z$
2740  PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$: PRINT FI: PRINT
      D$;"CLOSE"
2760  X$ = NA$ + " RS DATA"
2770  PRINT D$;"OPEN";X$
2780  PRINT D$;"POSITION";X$;",R";FI - 1
2790  PRINT D$;"WRITE";X$
2800  PRINT Z$(FI)
2810  PRINT D$;"CLOSE"
2812  IF FI > 150 THEN  GOSUB 2816: PRINT D$;"RUN FILEWRITE"
2814  IF ROCHE$ = "N" THEN 2890
2815  GOTO 2820
2816  X$ = "NEXTPROG": PRINT D$;"OPEN";X$: PRINT
      D$;"WRITE";X$: PRINT "ROCHESERVO": PRINT D$;"CLOSE"
2817  X$ = "RESPFILE": PRINT D$;"OPEN";X$: PRINT
      D$;"WRITE";X$: PRINT DR: PRINT PR: PRINT FLAG$: PRINT D$;
      "CLOSE": RETURN
2820  X$ = "NEXTPROG": PRINT
2830  PRINT D$;"OPEN";X$: PRINT D$;"READ";X$
2840  INPUT N$
2845  VTAB 10:Z$ = "LOADING ROCHE PROGRAM": GOSUB 7: PRINT
2850  PRINT D$;"CLOSE": PRINT D$;"RUN";N$
2890  TE = 0:RPLT = 0
2896  HOME : VTAB 24:Z$ = "V(OL/PRES  G(RAPH  M(ENU": GOSUB 7:
```

```
     GOTO 2590
3000 REM ***************************
3002 REM * GET PREV DATA FROM DISC *
3003 REM ***************************
3004 IF PR = 1 THEN 3050
3005 GOTO 3030
3010 HTAB 1: CALL  - 868: RETURN
3030 VTAB 20: GOSUB 3010:Z$ = "PRINTER? Y/N  ": GOSUB 7:: GET
     A$: PRINT A$: IF A$ = "N" THEN PR = 0: GOTO 3040
3031 REM **********************
3032 REM * INITIALISE PRINTER *
3033 REM **********************
3035 PR = 1: PRINT D$;"PR#1": PRINT : PRINT " ": POKE  -
     12528,7: POKE  - 12526,83: POKE  - 12525,64: POKE  -
     12529,255: POKE  - 12524,0: PRINT : PRINT D$;"PR#0"
3050 PRINT D$;"OPEN NAMEFILE": PRINT D$;"READ NAMEFILE"
3060 INPUT NA$: INPUT NO$: PRINT D$;"CLOSE NAMEFILE"
3110 X$ = NA$ + " RS DATALEN": ONERR  GOTO 3220
3120 PRINT D$;"OPEN";X$
3130 PRINT D$;"READ";X$: INPUT FILE
3140 PRINT D$;"CLOSE"
3150 X$ = NA$ + " RS DATA"
3170 PRINT D$;"OPEN";X$
3180 PRINT D$;"READ";X$
3190 VTAB 20: GOSUB 3010:Z$ = "OBTAINING FILE   ": GOSUB 7
3200 FOR I = 1 TO FI: VTAB 20: HTAB 28: PRINT I: INPUT Z$(I)
3210 NEXT
3220 PRINT D$;"CLOSE": POKE 216,0
3230 REM ****************************************
3235 REM * OBTAIN CALIBRATION FACTORS FOR SERVO *
3240 REM ****************************************
3250 X$ = "SERVOVARIABLES": ONERR  GOTO 3280
3260 PRINT D$;"OPEN";X$: PRINT D$;"READ";X$: INPUT SI: INPUT
     SE: INPUT SP
3270 GOTO 3300
3280 SI = 11.228:SE = 20.766:SP = .262
3300 PRINT D$;"CLOSE": POKE 216,0
3310 HY = 1000:NY = 10:HX = 40:NX = 4:RPLT = 0
```

```
3320  POKE  - 16368,0:S1 =  - 99999: GOTO 200
12989  REM ***************************
12990  REM * GRAPHICS TREND DISPLAY *
12991  REM ***************************
13000  NI = FI:EX = 0:NH = 0:NM = 0:ND = 0: HOME :F2 = NI:PI =
       1: GOTO 13024
13001  HOME : PRINT "HOW FAR BACK DO YOU WISH TO START PLOT?":
       PRINT : PRINT "TYPE THE ELAPSED TIME": GOSUB 13003:NM =
       Z: GOTO 13012
13003  VTAB 5: PRINT "FOLLOWED BY THE UNITS AND <RETURN>":
       VTAB 8: PRINT "EG  10M=10MINS  5H=5HOURS  2D=2DAYS":
       VTAB 12: HTAB 19: INPUT TI$
13004  HOME : IF  RIGHT$ (TI$,1) = "D" THEN Z =  VAL ( LEFT$
       (TI$, LEN (TI$) - 1)) * 1440: RETURN
13006  IF  RIGHT$ (TI$,1) = "H" THEN Z =  VAL ( LEFT$ (TI$
       LEN (TI$) - 1)) * 60: RETURN
13008  IF  RIGHT$ (TI$,1) = "M" THEN Z =  VAL ( LEFT$ (TI$,
       LEN (TI$) - 1)): RETURN
13010  POP : GOTO 13001
13012  HOME : PRINT "TYPE TIME INTERVAL TO BE PLOTTED": GOSUB
       13003:Z9 = Z
13015 START =  FN TIME(FI) - NM
13017  FOR I = 1 TO FILE: IF START >  FN TIME(I) THEN  NEXT
13019 START =  FN TIME(I):PI = I
13021 LAST = ST + Z9: FOR I = FI TO 1 STEP  - 1: IF LAST <  FN
      TIME(I) THEN  NEXT
13022 F2 = I
13023 LAST =  FN TIME(F2): GOTO 13025
13024 START =  FN TIME(PI):LAST =  FN TIME(FILE): HOME
13025 NM = LAST - START
13027 ET = ST: GOSUB 25000: GOSUB 26000:SD = DY:MS = MT:SH =
      H:SM = M
13028 ET = LA: GOSUB 25000: GOSUB 26000:LM$ =  STR$ (MT): IF
      LEN (LM$) < 2 THEN LM$ = "0" + LM$
13058  IF NM < 2 THEN  HOME : VTAB 10: PRINT " NOT ENOUGH
       VALUES TO PLOT":TR$ = "N":XP$ = "N": FOR II = 0 TO
       1000: NEXT : VTAB 10: CALL  - 868: RETURN
13234  IF MINS = 9999 THEN BA = 30 + 212 / EXTRA
```

```
13250  REM   ** HI-RES PLOTS **
13260  GOSUB 13420
13262  GOTO 16000
13280  AX = 0:CX = 0:PP = PI
13281  IF BX = 0 THEN BA = 30: GOTO 13290
13282 BA = 30 + 212 / NM * (BX + CX): REM  * BASE ADDRESS *
13290 X1 = BA
13300 Y1 = 190 - Z3 *  FN PLT(PP) + Z2
13305  IF PP = FI OR PP = F2 THEN  RETURN
13310  IF  FN PLT(PP) < LO THEN PP = PP + 1:CX = CX + ( FN
       TIME(PP) -  FN TIME(PP - 1)): GOTO 13282
13312  IF Y1 < 0 THEN Y1 = 0
13314  IF X1 > 278 THEN X1 = 278
13316  IF X1 < 30 THEN X1 = 30
13320  FOR X = PP + 1 TO F2:Y2 = 190 - Z3 *  FN PLT(X) + Z2
13330  IF  FN TIME(X) <  =  FN TIME(X - 1) THEN X = X + 1:
       GOTO 13410
13349  X2 = X1 + 212 / NM * ( FN TIME(X) -  FN TIME(X - 1)) +
       AX
13374  IF  FN PLT(X) < LO THEN AX = AX + 212 / NM * ( FN
       TIME(X) -  FN TIME(X - 1)): GOTO 13410
13375  IF X2 > 278 THEN X2 = 278
13376  IF Y2 < 0 THEN Y2 = 0
13377  IF X2 < 30 THEN X2 = 30
13378  AX = 0
13379  IF Y2 > 189 THEN Y2 = 189
13380  FOR II = 0 TO LP: HPLOT X1 + II,Y1 + II TO X2 + II,Y2 +
       II: NEXT
13390  HPLOT X1 + 1,Y1 + 1 TO X2 + 1,Y2 + 1
13400  X1 = X2:Y1 = Y2
13410  NEXT X: RETURN
13420  GOTO 13430: REM * AXES *
13422  FOR I = 3 TO 7: IF N / I -  INT (N / I) < .01 THEN NX =
       2 * I: RETURN
13424  NEXT I:N = N + 1: GOTO 13422
13430  IF NM < 61 THEN NT$ = "MINS":N = NM: GOSUB 13422:BX = N
       - NM:NM = N: GOTO 13505
13450  IF NM < 2160 THEN BX = SM: IF NM + BX < 2160 THEN NT$ =
```

```
      "TIME": GOTO 13498
13470 NT$ = "DATE":BX = SH * 60 + SM
13490 ND =  INT ((NM + BX) / 1440) + 1
13492 N = ND: GOSUB 13422:ND = N:NM = ND * 1440
13493 NT$ = "DATE": GOTO 13505
13498 NH = (NM + BX) / 60: IF (NM + BX) / 60 -  INT ((NM + BX)
      / 60) < > 0 THEN NH =  INT ((NM + BX) / 60) + 1
13500 N = NH: GOSUB 13422:NH = N:NM = NH * 60
13505  HOME : HGR2 : HCOLOR= 3: POKE 54,0: POKE 55,96: CALL
       1002
13506  FOR V = 0 TO NX:X = 30 + V * 212 / NX: HPLOT X,174 TO
       X,177: HPLOT X + 1,174 TO X + 1,177: NEXT
13510  FOR V = 0 TO NX STEP 2:X = 30 + V * 212 / NX: HPLOT
       X,174 TO X,180: HPLOT X + 1,174 TO X + 1,180: NEXT
13520  HPLOT 30,10 TO 30,180: HPLOT 31,10 TO 31,180
13525  IF NT$ = "MINS" THEN LX = SM:NZ = NM: FOR V = 0 TO NX
       STEP 2: VTAB 24:X = V * 212 / NX:A$ =  STR$ ( INT (LX +
       V * NZ / NX + .5)):HT =  INT (X / 6.975 + 5.5 -  LEN (A$)
       / 2): GOTO 13542
13530  IF NT$ = "TIME" THEN LX = SH:BX = SM:NZ = NH: GOTO
       13540
13532  LX = SD:BX = SH * 60 + SM:NZ = ND
13540  FOR V = 0 TO NX STEP 2: VTAB 24:X = V * 212 / NX:A$ =
       STR$ ( INT (LX + V * NZ / NX + .5)):HT =  INT (X /
       6.975 + 5.5 -  LEN (A$) / 2)
13542  A =  VAL (A$)
13544  IF NT$ = "DATE" THEN 13547
13545  IF NT$ = "TIME" AND A > 24 THEN A = A - 24: GOTO 13545
13546  IF NT$ = "MINS" AND A > 60 THEN A = A - 60
13547  GOTO 13553
13548  RESTORE : READ J:II = 0
13549 II = II + 1: READ J: IF II < MS THEN 13549
13550  IF A > J THEN A = A - J
13551  IF A > J THEN 13549
13553 HT = HT + 1
13554  HTAB HT: PRINT A: NEXT
13555  VTAB 23: HTAB 37: PRINT NT$;: IF NT$ = "DATE" THEN
       VTAB 24: HTAB 38: PRINT "/";LM$;
```

```
13600 RETURN
16000 REM * RESP *
16010 HPLOT 30,20 TO 245,20
16020 VTAB 2: HTAB 3: PRINT "10": VTAB 4: HTAB 4: PRINT "0":
      VTAB 3: HTAB 37: PRINT "PEEP"
16030 HPLOT 30,70 TO 245,70: VTAB 9: HTAB 37: PRINT "VT"
16050 HCOLOR= 1:Z0 = 24:Z1 = 2:Z2 =  - 160:Z3 = 2: GOSUB
      13280: REM PEEP
16060 VTAB 8: HTAB 2: PRINT "700": VTAB 11: HTAB 2: PRINT
      "500"
16070 HCOLOR= 5:Z0 = 13:Z1 = 4:Z2 =  - 60:Z3 = .1: GOSUB
      13280: REM VEXP
16080 VTAB 17: HTAB 3: PRINT "30": VTAB 19: HTAB 3: PRINT
      "20"
16090 HCOLOR= 3: HPLOT 30,130 TO 242,130: HPLOT 30,150 TO
      242,150
16100 VTAB 16: HTAB 37: PRINT "PK P": VTAB 20: HTAB 37: PRINT
      "COMP"
16110 HCOLOR= 2:Z0 = 22:Z1 = 2:Z2 = 0:Z3 = 2: GOSUB 13280:
      HPLOT 255,130 TO 271,130 TO 271,131 TO 255,131
16130 HCOLOR= 6:Z0 = 26:Z1 = 2:Z3 = 2: GOSUB 13280: HPLOT
      255,163 TO 271,163 TO 271,164 TO 255,164: REM COMP
16140 VTAB 1::Z$ = "M(ENU T(ABLE X(PAND P(RINT V(OL/PRES":
      PRINT Z$
16151 RP = 1: RETURN: REM * RP = 1 DENOTES THAT THE TREND HAS
      NOT TO BE RE-PLOTTED WHEN MORE DATA HAS BEEN OBTAINED *
16160 GOSUB 21100:X = A: IF X = 77 OR X = 205 THEN XP$ =
      "N":TR$ = "N":PLT$ = "N":ME$ = "Y":X = 0: GOTO 16200
16165 IF X = 84 OR X = 212 THEN XP$ = "N":TR$ = "N":PLT$ =
      "N":ME$ = "N":X = 0: GOTO 16200
16170 IF X = 216 OR X = 88 THEN 16300
16190 RETURN
16200 GOSUB 2500: RETURN
16300 GOSUB 2500: GOTO 13001
19997 REM ****************
19998 REM * DISPLAY MENU *
19999 REM ****************
20000 VTAB 1: HTAB 1: PRINT NA$;" / ";NO$;: GOSUB 24000: VTAB
```

```
      1: HTAB 28: PRINT BR + 1;
20005 HTAB 32: PRINT  MID$ (T$,7,2);"."; MID$ (T$,10,2);":";;
      MID$ (T$,13,2)
20007 V = 3
20010 VTAB V + 1: PRINT "1. TABLE DISPLAY
20020 VTAB V + 3: PRINT "2. VOLUME/PRESSURE LOOP
20030 VTAB V + 5: HTAB 1: PRINT "3. TYPE TEXT COMMENT
20031 IF AA = 180 AND PR = 1 THEN PR = 0: GOTO 20035
20034 IF AA = 180 AND PR = 0 THEN PR = 1
20035 VTAB V + 7: HTAB 1: PRINT "4. CHANGE OUTPUT (";: IF PR
      = 1 THEN  PRINT "PRINTER)": GOTO 20040
20037 PRINT "SCREEN) "
20040 VTAB V + 9: PRINT "5. CALIBRATE SERVO
20050 VTAB V + 11: PRINT "6. TREND DISPLAY
20060 VTAB V + 13: PRINT "7. ALTER DATA RETRIEVAL TIME
      (=";DR;" MIN)
20070 VTAB V + 15: PRINT "8. TAKE NEW READING
20080 VTAB V + 17: PRINT "9. QUIT
20100 VTAB 23:Z$ = "TYPE NO. REQUIRED": GOSUB 7
20112 IF AA <  > 179 THEN 20140
20114 HOME :Z$ = "TYPE COMMENT": GOSUB 7
20115 VTAB 5: PRINT "TYPE IN THE LABEL YOU WISH TO PRINT OUT
      FOLLOWED BY THE <RETURN> KEY.
20116 PRINT : PRINT : PRINT "USE THE LEFT ARROW KEY '<-' TO
      CORRECT  ANY ERRORS.
20117 PRINT : PRINT : PRINT "TO RETURN TO MENU, TYPE <RETURN>
      KEY
20118 INPUT Z$: IF Z$ = "" THEN 20190
20120 GOSUB 24000:TP$ =  MID$ (T$,7,2) + "." +  MID$
      (T$,10,2)
20122 PRINT D$;"PR#";PR: PRINT : PRINT TP$; SPC( 3);Z$:Z$ =
      "":AA = 0: PRINT D$;"PR#"0: HOME : RETURN
20140 IF AA = 181 AND RO$ = "N" THEN  PRINT : GOSUB 2816:
      HOME : VTAB 10:Z$ = "LOADING SERVOCALIBRATE": GOSUB 7:
      PRINT :  PRINT D$;"RUN SERVOCALIBRATE"
20142 IF AA = 181 THEN  HOME : VTAB 10:Z$ = "CALIBRATE FROM
      ROCHE PROGRAM": GOSUB 7: RETURN
20145 IF AA = 183 THEN  HOME : VTAB 10: PRINT " TYPE TIME
```

```
      (MIN) BETWEEN MEASUREMENTS": VTAB 13: HTAB 19: INPUT
      DR: HOME
20150 IF AA = 184 THEN 20190
20160 IF AA = 185 THEN  HOME : VTAB 10:Z$ = "TYPE 'RUN
      <RETURN> TO RE-START": GOSUB 7: END
20190 AA = 0: RETURN
21001 REM * KEYPRESS *
21100 AA =  PEEK ( - 16384): POKE  - 16368,0
21105 X = AA
21110 RETURN
23990 REM ************************
24000 REM * GET DATE/TIME IN MINS *
24005 REM ************************
24010 D$ =  CHR$ (4): PRINT D$:SLOT = 3
24030 PRINT D$;"IN#";SL: PRINT D$;"PR#";SL: INPUT T$: PRINT
      D$;"IN#0": PRINT D$;"PR#0"
24040 MT =  VAL ( MID$ (T$,1,2))
24050 D =  VAL ( MID$ (T$,4,2))
24060 H =  VAL ( MID$ (T$,7,2))
24070 M =  VAL ( MID$ (T$,10,2))
24080 S =  VAL ( MID$ (T$,13,6))
24090 RESTORE
24100 DTD = 0
24110 FOR I = 1 TO MT: READ J:DTD = DTD + J: NEXT
24120 DATA    0,31,28,31,30,31,30,31,31,30,31,30,31
24130 DTD = DTD + D
24140 IF MT > 2 AND L = 1 THEN  TD = DTD + 1
24150 STD = DTD * 86400 + H * 3600 + M * 60 + S
24160 S = STD
24170 RETURN
25000 REM *****************
25010 REM * MINS->DAYS/HRS *
25020 REM *****************
25040 DY =  INT (ET / 1440)
25050 ET = ET - DY * 1440
25060 H =  INT (ET / 60)
25070 M = ET - H * 60
25080 RETURN
```

```
26000 REM **************
26005 REM * DAYS->DATE *
26007 REM **************
26010 IF L = 1 AND DY = 60 THEN MT = 2:DY = 29: GOTO 26070
26015 IF L = 1 AND DY > 60 THEN DY = DY - 1
26017 RESTORE
26018 READ J
26020 FOR II = 1 TO 12: READ J:MT = MT + 1
26030 IF DY - J < 1 THEN ND = DY: GOTO 26070
26035 DY = DY - J
26040 NEXT
26070 RETURN
```

```
1  REM ******************
2  REM * SERVOCALIBRATE *
3  REM ******************
5  LOMEM: 32100:D$ =  CHR$ (4): TEXT : GOSUB 8: GOTO 9
7  HTAB (40 -  LEN (Z$)) / 2: PRINT Z$;: RETURN
8  HOME :Z$ = "SEIMENS SERVO 900B": GOSUB 7: PRINT : PRINT :Z$
   = "CALIBRATION PROGRAM": GOSUB 7: RETURN
9  VTAB 20:Z$ = "ASSEMBLER DATA COLLECTION": GOSUB 7
10  D$ =  CHR$ (4): PRINT D$:X$ = "NEXTPROG"
11  PRINT D$;"OPEN";X$: PRINT D$;"READ";X$: INPUT N$: PRINT
    D$;"CLOSE"
13  IF N$ = "R2000" THEN N$ = "R2000CALIBR"
18  PRINT D$;"NOMONI,C,O"
20  DIM Z$(200)
25  NP = 2: REM  DIV FAC
60  PRINT D$;"BLOAD RESP.OBJO"
85  SI = 12.02:SE = 16.76:SP = .262: REM * DEFAULT VALUES *
90  HOME : GOSUB 8
100  VTAB 8: HTAB 1: PRINT "1. USE 1 LITRE CALIBRATION SYRINGE
110  VTAB 10: PRINT "2. ENTER MEASURED TIDAL VOLUME
120  VTAB 15: PRINT "TYPE <RETURN> KEY TO RUN NEXT PROGRAM
130  VTAB 20: PRINT "TYPE NUMBER REQIRED ";: GET A$
140  IF A$ = "1" THEN ACTUAL = 1000: GOTO 190
150  IF A$ = "2" THEN  VTAB 20: HTAB 1: CALL  - 868: PRINT
     "ENTER MEASURED TIDAL VOLUME ";: INPUT ACTUAL: IF AC >
     2000 OR AC < 200 THEN 90
152  IF A$ = "2" THEN 200
160  IF A$ =  CHR$ (13) THEN 2050
170  GOTO 100
190  HOME : VTAB 5: PRINT "CONNECT THE 1 LITRE SYRINGE TO THE
     SERVO": PRINT "AND SET IT TO IMV MODE AT A FREQUENCY OF":
     PRINT "ZERO."
192  PRINT : PRINT : PRINT "MOVE THE SYRINGE OUT AND IN UNTIL
     THE": PRINT : PRINT "CALIBRATION PROCEDURE IS COMPLETED."
194  VTAB 20: HTAB 1: PRINT "WHEN READY TYPE THE <SPACEBAR>
     ";: GET A$: IF A$ < > " " THEN 194
```

```
195  REM ************************************
196  REM * CALL SERVO MACHINE CODE PROGRAM *
197  REM ************************************
200  POKE 27798,0: CALL 31000:N = N + 1
210  IF  PEEK (27798) <  > 34 THEN 225
212  HOME : VTAB 10: FLASH :Z$ = "ARRAY EXCEEDED": GOSUB 3020:
     GOSUB 223: GOTO 200
212  HOME : VTAB 10: FLASH :Z$ = "ARRAY EXCEEDED": GOSUB 7:
     GOSUB 223: GOTO 200
220  FOR I = 1 TO 10: PRINT "": NEXT
223  PRINT "","": VTAB 15:Z$ = "?  DISCONNECTED": GOSUB 7:
     NORMAL : VTAB 24: HTAB 8: PRINT "TYPE SPACEBAR TO RESET
     ";: GET A$: IF A$ = " ""          THEN  HOME : RETURN
224  FOR I = 1 TO 1000: NEXT : FLASH : VTAB 10: GOTO 223
225  REM ******************************************
226  REM * CALCULATE LENGTH OF INSPIRATORY ARRAY *
227  REM ******************************************
230  L1 = 27800: REM * START OF DATA *
240  L2 =  PEEK (27776) * 256 +  PEEK (27777) + 9 * 16 + 8
242  IC = L2 - L1
249  REM *****************************************
250  REM * CALCULATE LENGTH OF EXPIRATORY ARRAY *
251  REM *****************************************
255  REM * EXP *
260  L3 = L2 + 1
262  L4 =  PEEK (27778) * 256 +  PEEK (27779) + 9 * 16 + 8
264  EC = L4 - L3
269  REM ******************************
270  REM * OBTAIN INSP AND EXP TIMES *
271  REM ******************************
275  T1$ = "":T2$ = "":T3$ = ""
280  FOR I = 0 TO 5:T1$ = T1$ +  CHR$ ( PEEK (27780 + I) -
     128): NEXT
290  FOR I = 0 TO 5:T2$ = T2$ +  CHR$ ( PEEK (27786 + I) -
     128): NEXT
295  FOR I = 0 TO 5:T3$ = T3$ +  CHR$ ( PEEK (27792 + I) -
     128): NEXT
305  T1 =  VAL (T1$):T2 =  VAL (T2$):T3 =  VAL (T3$)
```

```
307  IF T2 < T1 THEN T1 = T1 - 60
308  IF T3 < T2 THEN T3 = T3 + 60
310  TI = T2 - T1:TE = T3 - T2
315  RTE =  INT (600 / (TI + TE) + .5) / 10: GOTO 318
316  FOR I = 1 TO IC:FI(I) = FI(I) * (PI(I) + 559) * SP / 559
318  CI = TI / IC * NP:CE = TE / EC * NP
460  Z$ = "   CALCULATING RESULTS    ": VTAB 24: GOSUB 7: GOTO
     800
799  REM ****************
800  REM * CALCULATIONS *
801  REM ****************
803  IW = 0:EW = 0:PP = 0:IV = 0:VI = 0:EV = 0:CO = 0:IR =
     0:IP = 0:EM = 0:ZZ = 0:YY = 0
805  REM ***************
810  REM * INSPIRATORY *
815  REM ***************
820  F = 27799:P = F + 1
830  FOR I = 1 TO IC STEP 2 * NP
835  VI = VI +  PEEK (F + I)
837  IF I > 20 AND  PEEK (P + I) > PP THEN PP =  PEEK (P + I)
840  NEXT
844  V1 = VI * CI
845  VI = V1 * SI
849  REM **************
850  REM * EXPIRATORY *
851  REM **************
860  F = 27799 + IC:P = F + 1
870  FOR E = 1 TO EC STEP 2 * NP
880  EV = EV +  PEEK (F + E)
885  V2 = EV * CE
890  NEXT
895  EV = V2 * SE
897  GOTO 1000
900  IF VI < 500 OR VI > 1500 THEN 200
910  IF EV < 500 OR EV > 1500 THEN 200
```

```
996  REM *******************
997  REM * DISPLAY RESULTS *
998  REM *******************
1001  IF N1 < 1 THEN  HOME
1002  VTAB 1: HTAB 1:Z$ = "CALIBRATION ROUTINE": GOSUB 7:
      PRINT :Z$ = "===================": GOSUB 7
1003  VTAB 5: HTAB 1: PRINT "AVERAGED VALUES";: HTAB 25: PRINT
      "SINGLE VALUES"
1004  VTAB 6: HTAB 1: PRINT "---------------";: HTAB 25: PRINT
      "-------------"
1005  VTAB 8: HTAB 1
1007  N1 = N1 + 1:II = II + V1:EE = EE + V2
1008  I1 = II-/ N1 * SI:I2 = EE / N1 * SE
1009  PRINT "INSP VOL="; INT (I1 + .5);"  ";: HTAB 27: PRINT
      INT (V1 * SI + .5);"  "
1010  VTAB 10: HTAB 1: PRINT "EXP. VOL="; INT (I2 + .5);"  ";:
      HTAB 27: PRINT  INT (V2 * SE + .5);"  "
1011  VTAB 14: PRINT "SI=";SI;: HTAB 27: PRINT "IC=";IC: PRINT
      : PRINT "SE=";SE;: HTAB 27: PRINT "EC=";EC
1012  IF N1 = 10 THEN 2000
1013  VTAB 20:Z$ = "BREATH NUMBER ": GOSUB 7: PRINT N1
1015  SI = SI * ACTUAL / I1:SE = SE * ACTUAL / I2: GOTO 200
1999  REM *************************************
2000  REM * WRITE CALIBRATION FACTORS TO DISC *
2002  REM *************************************
2010  X$ = "SERVOVARIABLES"
2020  PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$
2030  PRINT SI: PRINT SE: PRINT SP
2040  PRINT D$;"CLOSE"
2044  REM **************************
2045  REM * RETURN TO MAIN PROGRAM *
2046  REM **************************
2050  X$ = "NEXTPROG"
2060  PRINT D$;"OPEN";X$: PRINT D$;"READ";X$
2070  INPUT N$
2080  PRINT D$;"CLOSE"
2090  IF N$ = "R2000" THEN N$ = "R2000CALIBR"
2100  PRINT D$;"RUN";N$
```

```
                 *************
                 * RESP.OBJO *
                 *************

         ORG $7918        START AT DECIMAL 31,000
INITIAL  LDA #$00         ZERO ARRAY
         STA $6C96        EXCEEDED POINTER
         LDX #$02
         LDA #$6C         ARRAY STARTS AT $6C(98); HIGH BYTE IS
         NOP              INDEXED BY 'X'
         NOP
         STA STORE,X      SET UP STORE+2 FOR SELF-MODIFYING
         NOP              CODE
         NOP
I->E     JSR IECHAN       READ I/E CHANNEL
         CMP #$64         CHECK STILL IN
         BCC I->E         INSPIRATION
         NOP
         NOP
E->I     JSR IECHAN       READ I/E CHANNEL
         CMP #$64         CHECK STILL IN
         BCS E->I         EXPIRATION
         JSR TIMEO        BEGIN DATA COLLECTION : READ CLOCK
         LDY #$00         ZERO Y REGISTER
         NOP
         NOP
INSPIRN  JSR GETDATA      GET INSPIRATORY PRES AND FLOW DATA
         CMP #$64         CHECK STILL INSPIRATION
         BCC INSPIRN
         LDA STORE,X      STORE HIGH BYTE OF
         STA $6C80        INSPIR ARRAY LENGTH
         STY $6C81        STORE LOW BYTE
         JSR TIME1
         NOP
         NOP
         LDY $6C81        RELOAD 'Y' REG TO STORE EXPIRATORY
```

```
         NOP                DATA
EXPIRN   JSR GETDATA
         CMP #$64           GET EXPIRATORY DATA
         BCS EXPIRN
         LDA STORE,X        STORE HIGH BYTE OF
         STA $6C82          END OF EXPIR ARRAY
         STY $6C83          STORE LOW BYTE
         JSR TIME2
         RTS                RETURN TO BASIC
         NOP
         NOP
         NOP
         NOP
ADC      STA $C401          'A' CONTAINS CHANNEL NO. FOR
         PHA                CONVERSION AND STARTS ADC
         PLA                NEED APPROX 7 MICROS DELAY TO RESET
         NOP                'EOC' FLAG
BIT      BIT $C402
         BPL BIT            TEST EOC FLAG
         LDA $C400          STORE VALUE FROM ADC IN 'A'
STORE    STA $6C98,Y        STORE RESULTS FROM $6C98 ON, INDEXED
         INY                BY 'Y'
         TYA
         BNE CHECKLEN
         INC STORE,X        'Y' RETURNED TO ZERO, ANOTHER 256
         NOP                 VALUES STORED IN MEMORY.
         NOP                 INCREMENT 'STORE'
         NOP
CHECKLEN LDA STORE,X        CHECK LENGTH OF DATA ARRAY
         CMP #$78           ARRAY EXCEEDED IF >$78
         BCC RETURN
         LDA #$22           $22 STORED IN $6C96 IF ARRAY IS
         STA $6C96          EXCEEDED
         NOP
RETURN   RTS                RETURN TO BASIC
         NOP
WAIT     LDA #$FF           DELAY LOOP
         STA $6C97
```

```
DEC      DEC $6C97
         LDA $6C97
         BNE DEC
         RTS
         NOP
         NOP
IECHAN   LDA #$07       COLLECT VALUE
         STA $C401      FROM I/E CHANNEL
         PHA            OF ADC
         PLA
BIT2     BIT $C402
         BPL BIT2
         LDA $C400
         RTS
         NOP
         NOP
GETDATA  LDA #$0B       SELECT CHANNEL 11 OF ADC
         JSR ADC
         LDA #$03       SELECT CHANNEL 03 OF ADC
         JSR ADC
         JSR WAIT       DELAY LOOP
         JSR WAIT
         JSR IECHAN     SELECT I/E/ CHANNEL OF ADC
         RTS
         NOP
         NOP
TIME0    LDX #$00       'X' IS USED TO INDEX STORAGE
         JMP CLOCK      LOCATIONS  FOR CLOCK DATA
         NOP
TIME1    LDX #$06
         JMP CLOCK
         NOP
TIME2    LDX #$0C
         NOP
         NOP
CLOCK    LDA $45        SAVE CONTENTS IN LOCATION $45
         PHA
         TXA            SAVE X REG
```

```
         PHA
         TYA              SAVE Y REG
         PHA
         LDA $38          SAVE LOCATION $38
         PHA
         LDA $39          SAVE LOCATION $39
         PHA
         LDA #$C3         SET I/O FOR CLOCK IN SLOT 3
         STA $39
         JSR $C300
         PLA              RESTORE LOCATION $39
         STA $39
         PLA              RESTORE LOCATION $38
         STA $38
         PLA              RESTORE 'Y' REGISTER
         TAY
         PLA              RESTORE 'X' REGISTER
         TAX
         PLA              RESTORE 'A' REGISTER
         NOP
         NOP
         LDY #$05         LOAD 'Y' WITH 05 FOR COLLECTION OF
         NOP              CLOCK DATA BEGINNING AT LOCATION
LOADCLK  LDA $282,Y       $287
         NOP
         STA $6C84,X      STORE CLOCK DATA FROM LOCATION
         INX              $6C84-$6C90
         DEY
         BPL LOADCLK
         LDX #02          RELOAD 'X' REGISTER TO ALLOW
         RTS              SELF-MODIFYING CODE AT 'STORE'
```

SECTION 3

FILE SUITE

```
1 REM **************
2 REM * HELLO-FILE *
3 REM **************
4 POKE 50944,5: LOMEM: 27800
6 DIM SM$(10),IN$(36),Z$(150),V(15): GOTO 10
7 FOR I = 1 TO 500: NEXT : POKE  - 16383,0: RETURN
10  TEXT : HOME
11  POKE 216,0: GOSUB 13: GOTO 15
13  Z$ = "GLASGOW ROYAL INFIRMARY": GOSUB 14:Z$ = "RESPIRATORY
    INTENSIVE CARE UNIT": GOSUB 14:Z$ = "FILE PROGRAM": GOSUB 14:
    RETURN
14  PRINT : HTAB (40 -  LEN (Z$)) / 2: PRINT Z$: RETURN
15  D$ =  CHR$ (4): PRINT D$;"BLOAD CHARGEN A$6000 L$100"
16  PRINT D$;"BLOAD COLOSSAL.SET A$6900 L$300"
17  VTAB 20: HTAB 4: PRINT "DO YOU WISH INSTRUCTIONS? Y/N  ";:
    GET A$: PRINT A$: IF A$ = "N" THEN 20
18  IF A$ = "Y" THEN 100
19  GOTO 17
20  HOME : GOSUB 13: VTAB 19:Z$ = "   THE PROGRAM IS NOW
    LOADING   ": GOSUB 14
25  N$ = "":X$ = ""
30  PRINT D$;"BLOAD CHAIN,A520"
40  CALL 520"PART1"
90  RESTORE
100  HOME : PRINT "   THIS PROGRAM ENABLES PATIENT DATA TO BE
     TYPED INTO THE COMPUTER AND TO BE    DISPLAYED SUBSEQUENTLY
     IN A GRAPHICAL ORTABULAR FORM."
110  PRINT : PRINT : PRINT "   THE PROGRAM IS DRIVEN BY
     SELECTING   ONE FROM SEVERAL CHOICES DISPLAYED ON   THE
     SCREEN.
115  FOR I = 1 TO 3000: NEXT
117  PRINT
120  PRINT : PRINT "EG. ONE PATIENT CAN BE SELECTED FROM A
     LIST OF SEVERAL PATIENTS.
125  FOR I = 1 TO 4: READ N$(I): NEXT
130  VTAB 15: FOR I = 1 TO 4: PRINT I;".  ";N$(I): PRINT :
     NEXT
140  PRINT "SELECT PATIENT NO. FOR FILE RETRIEVAL ";: GET A$
```

```
145  IF A$ < "1" OR A$ > "4" THEN 150
147  GOTO 160
150  HOME : VTAB 10: HTAB 1: PRINT "YOU TYPED ";A$: VTAB 12:
     PRINT "THIS IS AN INVALID CHARACTER"
155  VTAB 15: PRINT "THIS WILL NOT MATTER BECAUSE THE APPLE
     COMPUTER WILL REJECT SELECTIONS WHICH   ARE OUTWITH THE
     REQUIRED RANGE.": FOR I = 1 TO 6000: NEXT : GOTO 90
160  HOME : PRINT "GOOD, YOU TYPED A '";A$;"'."
170  PRINT : PRINT "    YOU WILL ALSO BE ASKED TO ENTER
     DATAINTO THE COMPUTER.  THE NUMBERS ARE     TYPED ONE AFTER
     THE OTHER.  IF A WRONG  NUMBER IS TYPED, IT CAN BE DELETED
     BY TYPING THE 'BACKSPACE' OR LEFT ARROW KEY"
180  VTAB 10: PRINT "    ONCE THE CORRECT NUMBER HAS BEEN
     TYPED, YOU MUST ENTER THIS DATA INTO THECOMPUTER BY PRESSING
     THE <RETURN> KEY."
185  N$ = "134"
190  VTAB 17: HTAB 1: PRINT "TYPE IN THIS VALUE FOR SERUM NA+"
195  VTAB 18: HTAB 8: PRINT "->";N$
200  VTAB 20: HTAB 10: PRINT "NA+
210  VTAB 21: HTAB 9: INPUT NA$
220  VTAB 23: IF NA$ = N$ THEN  PRINT "CORRECT!": FOR I = 1 TO
     3000: NEXT : VTAB 23: FOR I = 1 TO 20: PRINT " ";: NEXT :
     VTAB 21: HTAB 1: FOR I = 1 TO 20: PRINT " ";: NEXT
230  IF NA$ <  > N$ THEN  PRINT "NO - THAT IS INCORRECT": FOR
     I = 1 TO 3000: NEXT : VTAB 23: HTAB 1: FOR I = 1 TO 30:
     PRINT " ";: NEXT : VTAB 21: HTAB 1: FOR I = 1 TO 20: PRINT
     " ";:  NEXT : GOTO 190
235  IF N$ = "146" THEN 241
240  N$ = "146": GOTO 190
241  HOME : FOR I = 1 TO 3: PRINT "": NEXT
242  VTAB 5:Z$ = "REMEMBER": FLASH : GOSUB 14: NORMAL : VTAB
     10: PRINT "   YOU CANNOT UNWITTINGLY DESTROY DATA  OR
     DAMAGE THE SYSTEM."
244  VTAB 13: PRINT "   IF YOU ARE IN DIFFICULTY AT ANY
     POINTTYPE THE <RETURN> KEY AND YOU WILL BE   RETURNED TO A
     SELECTION MENU.
246  VTAB 20: HTAB 1: PRINT "WHEN READY TYPE 1 ";: GET A$: IF
     A$ <  > "1" THEN 246
```

```
250  HOME : VTAB 5: PRINT "    THESE ARE THE ONLY OPERATIONS
     WHICH  YOU WILL HAVE TO PERFORM TO USE THIS    PROGRAM.
255  PRINT : PRINT
260  PRINT : PRINT "1. SELECTION OF A NUMBER OR A LETTER
     FROM A MENU DISPLAYED ON THE SCREEN.
265  PRINT
270  PRINT : PRINT "2. ENTRY OF DATA INTO THE COMPUTER. THIS
     REQUIRES THE <RETURN> KEY TO BE         PRESSED ONCE THE
     CORRECT NUMBERS HAVE   BEEN TYPED ON TO THE SCREEN."
280  VTAB 22: HTAB 1: PRINT "TO LOAD THE FILE PROGRAM TYPE '1'
      ": GOSUB 7: GET A$: PRINT A$: IF A$ < > "1" THEN 280
290  HOME : GOTO 20
1000  DATA  BROWN J,SMITH PJ, MACDONALD H,PETERS DG
```

```
1  REM ***************
2  REM *  FILEPART1  *
3  REM * NEEDS HELLO *
4  REM ***************
5  R(0) = 8:R(1) = 15:R(2) = 20:R(3) = 25:R(4) = 31:R(5) = 37
6  GOSUB 7: TEXT :D$ =  CHR$ (4): GOTO 10
7  POKE  - 16368,0: FOR I = 1 TO 500: NEXT : RETURN
8  HTAB (40 -  LEN (Z$)) / 2: PRINT Z$;: RETURN
10  DEF  FN TIME(X) =  VAL ( LEFT$ (Z$(X),6)):
20  PLT = 0: GOTO 5000:PLT = 1
97  REM ****************
98  REM * DISPLAY MENU *
99  REM ****************
100  I = 0: HOME : PRINT "MENU";: HTAB 20: PRINT X$: PRINT
     "===="
105  IF N$ < > "" THEN  VTAB 1: HTAB 1: PRINT N$;: HTAB 34:
     PRINT NO$:A$ = N$: GOSUB 108: HTAB 20:A$ = X$: GOSUB 108:
     HTAB 34:A$ = NO$: GOSUB 108
106  GOTO 110
108  FOR I = 1 TO  LEN (A$): PRINT "=";: NEXT : RETURN
110  V = 7: VTAB V: HTAB 1: PRINT "1. LIST NAMES ON FILE
112  VTAB V + 3: HTAB 1: PRINT "2. ADD / ALTER NAME ON FILE
114  IF X$ = "" THEN ZZ$ = "2": GOTO 140
116  ZZ$ = "5"
120  VTAB V + 6: HTAB 1: PRINT "3. DISPLAY / ENTER DATA
130  VTAB V + 9: HTAB 1: PRINT "4. TREND PLOT
132  VTAB V + 12: HTAB 1: PRINT "5. PRINT OUT DATA
134  VTAB V + 15: HTAB 1: PRINT "6. SELECT DIFFERENT SYSTEM
140  VTAB 24: GOSUB 7
145  GET A$: VTAB 3: PRINT : IF A$ = "1" THEN 1000
147  IF A$ = "2" THEN 1200
148  IF A$ = "3" THEN 5000
150  IF A$ = "4" AND X$ < > " TEXT " THEN  PRINT : PRINT
     D$;"BLOAD CHAIN,A520": CALL 520"FILEPLOT"
152  IF A$ = "5" THEN  GOSUB 5010: GOSUB 6000
154  IF A$ = "6" THEN 1095
160  GOTO 100
```

```
997  REM ********************
998  REM * OBTAIN NAME LIST *
999  REM ********************
1000  ON ERR GOTO 1014
1012  GOTO 1018
1014  HOME :Z$ = "NO NAMES ON LIST": VTAB 10: HTAB (40 -  LEN
      (Z$)) / 2: PRINT Z$: FOR I = 1 TO 2000: NEXT
1016  POKE 216,0: GOTO 100
1018  NA$ = "NAMELISTLEN": PRINT D$;"OPEN";NA$
1019  PRINT D$;"READ";NA$: INPUT NI
1020  PRINT D$;"CLOSE";NA$
1021  NA$ = "NAMELIST"
1022  PRINT D$;"OPEN";NA$: PRINT D$;"READ";NA$
1024  FOR I = 1 TO NI: INPUT Z$(I): NEXT : PRINT
      D$;"CLOSE";NA$
1030  HOME : PRINT "FILE";
1035  HTAB 8: PRINT "NAME";: HTAB 25: PRINT "HOSPITAL NO.":
      PRINT "====";: HTAB 8: PRINT "====";: HTAB 25: PRINT
      "============": PRINT
1040  FOR J = 1 TO NI: PRINT : PRINT " ";J;".   ";:NA$ =  MID$
      (Z$(J),2,20): PRINT NA$;: HTAB 28:NO$ =  MID$
      (Z$(J),22,6): PRINT NO$: NEXT
1050  VTAB 22: HTAB 1: PRINT "SELECT FILE NUMBER FOR DATA
      RETRIEVAL"
1060  VTAB 24: HTAB 1:Z$ = "TYPE <RETURN> TO RETURN TO MENU ":
      GOSUB 8
1065  GOSUB 7: GET A$: IF A$ =  CHR$ (13) THEN 100
1070  IF  VAL (A$) < 1 OR  VAL (A$) > NI THEN 1065
1075  A =  VAL (A$)
1090  NA$ =  MID$ (Z$(A),2,20):NO$ =  MID$ (Z$(A),22,6)
1092  N$ = "": FOR I = 1 TO 20: IF  MID$ (NA$,I,1) < > "."
      THEN N$ = N$ +  MID$ (NA$,I,1): NEXT
1093  REM ***************
1094  REM * SYSTEM LIST *
1095  REM ***************
1096  HOME : PRINT N$;: HTAB 30: PRINT NO$
1097  FOR I = 1 TO  LEN (N$): PRINT "=";: NEXT : HTAB 30: FOR
      I = 1 TO 6: PRINT "=";: NEXT
```

```
1100  VTAB 6: HTAB 1: PRINT "1. CARDIOVASCULAR";: HTAB 24:
       RINT "6. BLOOD GASES
1110  VTAB 8: PRINT "2. RESPIRATORY";: HTAB 24: PRINT "7. UREA
      & ELECTS
1120  VTAB 10: PRINT "3. FLUID INPUT";: HTAB 24: PRINT "8.
      LFT'S
1130  VTAB 12: PRINT "4. FLUID OUTPUT";: HTAB 24: PRINT "9.
      HAEMATOLOGY
1140  VTAB 14: PRINT "5. FLUID BALANCE";: HTAB 24: PRINT "T.
      TEXT COMMENTS
1142  Z$ = "TYPE <RETURN> TO RETURN TO MENU": VTAB 17: GOSUB 8
1148  FOR I = 1 TO 10:SM$(I) = "": NEXT
1160  VTAB 23: HTAB 1: PRINT "SELECT SYSTEM REQUIRED    ";:
      GOSUB 7: GET A$: IF A$ =  CHR$ (13) THEN 100
1162  PRINT A$: IF A$ = "T" THEN X$ = " TEXT ": GOTO 1190
1165  IF A$ < "1" OR A$ > "9" THEN 1160
1170  ON  VAL (A$) GOTO
      1171,1172,1173,1174,1175,1176,1177,1178,1179
1171  X$ = " CVS ": GOTO 1190
1172  X$ = " RS ": GOTO 1190
1173  X$ = " FL IN ": GOTO 1190
1174  X$ = " FL OUT ": GOTO 1190
1175  GOSUB 12000: GOTO 100
1176  X$ = " BL GAS ": GOTO 1190
1177  X$ = " SER EL ": GOTO 1190
1178  X$ = " LFT'S ": GOTO 1190
1179  X$ = " HAEM "
1190  NI = 0: GOTO 3000
1200  REM ***********
1201  REM * ADDNAME *
1202  REM ***********
1210  PRINT D$;"BLOAD CHAIN,A520": CALL 520"PART1ADDNAME"
1220  PRINT D$;"RUN ADDNAME
1999  REM ******************************
2000  REM * STORE DATA STRINGS ON DISC *
2001  REM ******************************
2010  Z$ = N$ + X$ + "DATALEN
2050  PRINT D$;"OPEN";Z$: PRINT D$;"WRITE";Z$
```

```
2060  PRINT NI
2070  PRINT D$;"CLOSE";Z$
2080  Z$ = N$ + X$ + "DATA"
2090  PRINT D$;"OPEN";Z$
2100  PRINT D$;"POSITION";Z$;",R";NI - 1
2110  PRINT D$;"WRITE";Z$: PRINT Z$(NI): PRINT D$;"CLOSE"
2120  GOTO 100
2999  REM ***************************
3000  REM * RETRIEVE DATA FROM DISC *
3001  REM ***************************
3006  FOR I = 1 TO NI:Z$(I) = "": NEXT :NI = 0
3010  Z$ = N$ + X$ + "DATALEN
3012  ONERR  GOTO 3014
3013  GOTO 3020
3014  POKE 216,0: PRINT D$;"CLOSE": PRINT D$;"DELETE";Z$: HOME
      :Z$ = "NO DATA AVAILABLE": VTAB 10: HTAB (40 -  LEN
      (Z$)) / 2: PRINT Z$: FOR I = 1 TO 2000: NEXT : GOTO 100
3020  PRINT D$;"OPEN";Z$: PRINT D$;"READ";Z$: INPUT NI: PRINT
      D$;"CLOSE";Z$
3030  POKE 216,0:Z$ = N$ + X$ + "DATA
3040  PRINT D$;"OPEN";Z$: PRINT D$;"READ";Z$: FOR I = 1 TO NI:
      INPUT Z$(I): NEXT
3042  PRINT D$;"CLOSE";Z$: GOTO 100
3340  VTAB 18: HTAB 1: PRINT "PLAT'S ";IT(7);"ML
3999  REM ***********************************
4000  REM * DISPLAY AND ENTER TEXT COMMENTS *
4001  REM ***********************************
4005  HOME :PN = 6
4010  IF NI = 0 THEN 4100
4020  LINE = 0: IF PN > NI THEN PN = NI
4030  FOR I = NI TO NI - PN + 1 STEP  - 1
4040  LINE = LI + 1: VTAB 1 + (LI * 3)
4050  SM$(1) =  MID$ (Z$(I),1,6):ET =  VAL (SM$(1)): GOSUB
      25000: GOSUB 26000
4060  HTAB 1: PRINT SM$(0);: HTAB 8: PRINT SM$(1): PRINT  MID$
      (Z$(I),7,40)
4070  NEXT
4100  NI = NI + 1
```

```
4110  VTAB 24: HTAB 1:Z$ = "ENTER TEXT COMMENT? Y/N ": GOSUB
      8: GET A$: IF A$ = "N" THEN 100
4120  IF A$ < > "Y" THEN 4110
4125  VTAB 1:Z$ = "** NO DATE OR TIME REQUIRED **": GOSUB 8
4130  VTAB 23: HTAB 1: PRINT "ENTER COMMENT REQUIRED AND TYPE
      <RETURN>";
4135  VTAB 24:Z$ = "** DO NOT USE MORE THAN ONE LINE **":
      GOSUB 8: VTAB 1: HTAB 40: INPUT Z$(NI)
4140  GOSUB 27000
4150  ZZ$ =  STR$ ( INT (S / 60)): FOR II = 1 TO 6: IF  LEN
      (ZZ$) < 6 THEN ZZ$ = "0" + ZZ$: NEXT
4160  Z$(NI) = ZZ$ + Z$(NI)
4170  GOTO 2000
4997  REM ****************
4998  REM * DISPLAY DATA *
4999  REM ****************
5000  HOME : IF X$ = " TEXT " THEN NV = 2:S(1) = 7:L(1) = 46:
      GOTO 4000
5001  IF X$ < > "" THEN  GOSUB 5010: GOTO 5070
5002  IF PLT = 0 THEN 100
5005  HOME : VTAB 10: PRINT "NO DATA IN MEMORY YET----TYPE 1
      OR 2": FOR I = 1 TO 2000: NEXT :100
5010  N2 = 0:PN = 6:S(1) = 1:L(1) = 6:S(2) = 7:L(2) = 3:S(3) =
      10:L(3) = 3:S(4) = 13:L(4) = 3:S(5) = 16:L(5) = 3:S(6) =
      19:L(6) = 3:S(7) = 22:L(7) = 3:S(8) = 25:L(8) = 4:S(9) =
      29:L(9) = 3:NV = 9:N1 = 6
5015  IF X$ = " CVS " THEN  PRINT "DATE   TIME    HR  SYS  DIA
      CVP  PAWP":PN = 6:N2 = 3: RETURN
5020  IF X$ = " SER EL " THEN  PRINT "DATE   TIME    NA   K
      CL   UREA  CREAT":PN = 5:N2 = 3:NVINSTRING = 9:L(5) =
      4:S(6) = 20:L(6) = 3:S(7) = 23:L(7) = 3:S(8) = 26:L(8) = 3
      :S(9) = 29:L(9) = 4: RETURN
5030  IF X$ = " HAEM " THEN  PRINT "DATE   TIME   HB   PCV
      WBC  PLATLETS":N1 = 5:NV = 5:L(2) = 4:S(3) = 11:L(3) =
      2:L(4) = 4:S(5) = 17:L(5) = 3: RETURN
5040  IF X$ = " BL GAS " THEN PN = 3: PRINT "DATE   TIME  FIO2
      PO2 PCO2  PH    BE":L(3) = 4:S(4) = 14:L(4) = 4:S(5) =
      18:L(5) = 4:S(6) = 22:L(6) = 3: RETURN
```

```
5050  IF X$ = " RS " THEN  PRINT "DATE   TIME   RR   INVT EXVT
      VMIN PAU":PN = 6:N1 = 6:N2 = 3:NV = 9:L(2) = 2:S(3) =
      9:L(3) = 4:L(4) = 4:S(5) = 17:L(5) = 5:S(6) = 22:L(6) = 2:
      S(7) = 24:L(7) = 2:S(8) = 26:L(8) = 2:S(9) = 28:L(9) = 5:
       RETURN
5055  IF X$ = " FL IN " THEN  PRINT "DATE   TIME    N/G  IV1
      IV2   IV3 DRUG":PN = 6:N1 = 6:N2 = 3:NV = 9
5056  IF X$ = " FL IN " THEN L(2) = 4:S(3) = 11:L(3) = 4:S(4)
      = 15:L(4) = 4:S(5) = 19:L(5) = 4:S(6) = 23:L(6) = 3:S(7)
      = 26:L(7) = 3:S(8) = 30:L(7) = 4:S(9) = 34:L(9) = 2:
      RETURN
5060  IF X$ = " FL OUT " THEN  PRINT "DATE   TIME   N/G URINE
      -L-   -R- CHEST":PN = 6:N1 = 5:N2 = 4:NV = 8
5062  IF X$ = " FL OUT " THEN S(6) = 19:L(6) = 4:S(7) =
      23:L(7) = 4:S(8) = 27:L(8) = 4: RETURN
5068  IF X$ = " LFT'S " THEN  PRINT "DATE   TIME  ALB  GLOB
      BIL    ASP   ALA":PN = 5:N2 = 3:NVINSTRING = 9:S(5) =
      16:L(5) = 4:S(6) = 20:L(6) = 4:S(7) = 24:L(7) = 3:S(8) =
      27:L(8) = 3:S(9) = 30:L(9) = 3
5069  RETURN
5070  IF NI = 0 THEN 5110
5072  LINE = 0:LX = 0:VV = 0: IF PN > NI THEN PN = NI
5080  FOR I = NI TO NI - PN + 1 STEP  - 1:VV = VV + 1
5085  LINE = LI + 1:V(VV) = 3 + LI + LX: VTAB V(VV)
5090  FOR AB = 1 TO NVINSTR
5100  SM$(AB) =  MID$ (Z$(I),S(AB),L(AB)): IF SM$(AB) = "-99"
      THEN SM$(AB) = " --"
5104  NEXT
5106  ET =  VAL (SM$(1)): GOSUB 25000: GOSUB 26000
5107  HTAB 1: PRINT SM$(0);: FOR J = 0 TO N1 - 1: GOSUB 5490:
      HTAB H: PRINT SM$(J + 1);: NEXT J
5108  IF N2 = 0 THEN 5117
5109  HTAB 1
5110  IF X$ = " FL OUT " THEN  VTAB 13: PRINT "DATE   TIME
       -1-  -2-  -3- ABDOM DRAINS
5111  IF X$ = " FL IN " THEN  VTAB 13: PRINT "DATE   TIME
      PLAT PLAS UNITS BLOOD
5112  IF X$ = " CVS " THEN  VTAB 13: PRINT "DATE   TIME  MEAN
```

```
     T CORE T DIFF
5113 IF X$ = " LFT'S " THEN  VTAB 13: PRINT "DATE   TIME ALK
     PH CALC PHOS
5114 IF X$ = " SER EL " THEN  VTAB 13: PRINT "DATE   TIME  P
     OS  U OS  GLUC
5115 IF X$ = " RS " THEN  VTAB 13: PRINT "DATE   TIME  PEEP
     COMP   WORK
5116 IF X$ = "BG" THEN  VTAB 13: PRINT "DATE   TIME IPPV
     PEEP
5117 IF NI = 0 THEN 5130
5118 IF N2 = 0 THEN 5125
5119 VTAB 15 + LI + LX
5120 HTAB 1: PRINT SM$(0);: HTAB 8: PRINT SM$(1);: FOR J = 1
     TO N2: GOSUB 5490: HTAB H: PRINT SM$(J + N1);: NEXT J
5122 FOR J = 1 TO 10:SM$(J) = "": NEXT
5125 NEXT I
5130 FOR J = 22 TO 24: VTAB J: HTAB 1: CALL  - 868: NEXT J
5132 REM ****************
5133 REM * CORRECT DATA *
5134 REM ****************
5135 HTAB 1: VTAB 24: PRINT N$;: HTAB 20: PRINT "C(ORRECT OR
     ANY KEY ";: GET A$: IF A$ <  > "C" THEN 5170
5136 IF NI = 0 THEN 5130
5137 FOR II = 1 TO PN: VTAB V(II): FLASH : HTAB 13: PRINT II:
     NEXT : NORMAL
5138 VTAB 24: HTAB 13: PRINT "   TYPE LINE NO TO ALTER  ";:
     GET A$: IF A$ < "1" OR A$ >  STR$ (PN) THEN  IF A$ <  >
     "M" OR A$ <  >  CHR$ (13) THEN 5138
5139 IF A$ = "M" THEN  RETURN
5140 VTAB 24: HTAB 1: CALL  - 868: HTAB 5: PRINT "D(ELETE
     A(LTER  L(EAVE LINE "; VAL (A$);" ";: GET Z$: IF Z$ =
     "A" THEN 5164
5141 IF Z$ = "L" OR Z$ =  CHR$ (13) THEN 100
5142 IF Z$ <  > "D" THEN 5140
5143 II = NI + 1 -  VAL (A$)
5144 FOR JJ = II TO NI:Z$(JJ) = Z$(JJ + 1): NEXT :Z$(NI) =
     "":NI = NI - 1
5145 VTAB 24: HTAB 1: PRINT "   WRITE CORRECTED FILE TO DISC
```

```
     Y/N ";: GET A$: IF A$ = "N" THEN 100
5146 IF A$ < > "Y" THEN 5145
5148 Z$ = N$ + X$ + "DATALEN": PRINT A$
5149 PRINT D$;"OPEN";Z$: PRINT D$;"WRITE";Z$: PRINT NI: PRINT
     D$;"CLOSE";Z$
5150 Z$ = N$ + X$ + "DATA
5152 PRINT D$;"OPEN";Z$: PRINT D$;"WRITE";Z$
5154 FOR I = 1 TO NI: PRINT Z$(I): NEXT
5156 PRINT D$;"CLOSE";Z$:Z$ = "":AL$ = "": GOTO 100
5164 ALTER$ = "Y":AL = NI + 1 -  VAL (A$): GOTO 5260
5165 GET A$: VTAB 24: HTAB 18: FOR II = 18 TO 38: PRINT " ";:
     NEXT
5170 VTAB 24: HTAB 20: PRINT "    U(PDATE M(ENU   ";: GET A$:
     IF A$ = "U" THEN 5260
5180 IF A$ < > "M" THEN 5130
5190 GOTO 100
5240 GOTO 5130
5245 REM ***************
5250 REM * UPDATE DATA *
5255 REM ***************
5260 VTAB 3: HTAB 1: PRINT "DD/MM  HH.MM
5262 HTAB 3: VTAB 22: PRINT "ENTER DATA, THEN PRESS <RETURN>
     KEY"
5265 HTAB 1: VTAB 24: PRINT "'<-' TO DELETE ENTRY   '--' IF
     NO ENTRY";: VTAB 1: HTAB 1: GOSUB 5010
5270 J = 0
5272 II = 1:H = 1:ZZ = 1
5274 IF N2 < > 0 THEN 5293
5280 V = 3: GOSUB 5330:J = J + 1: IF J <  = N1 THEN 5280
5291 GOTO 5294
5293 IF E = 0 THEN V = 3: GOSUB 5330::J = J + 1: IF J <  = N1
     THEN 5293
5294 J = 2
5295 IF N2 < > 0 THEN V = 15:ZZ = 15:E = 5:H = 15
5297 IF N2 < > 0 THEN  GOSUB 5330:J = J + 1: IF J + 5 <  =
     N1 + N2 THEN 5297
5300 E = 0: VTAB 22: HTAB 1: CALL  - 868: VTAB 24: CALL  -
     868:  HTAB 10: PRINT "DATA OK? Y/N ->";: GET A$: IF A$ =
```

```
     "N" THEN 5000
5302 IF A$ < > "Y" THEN 5300
5303 GOSUB 5304: GOTO 5315
5304 HOME : VTAB 10: HTAB 1: PRINT "ENTER TEXT COMMENT? Y/N
     ";: GET A$: IF A$ = "N" THEN  RETURN
5305 IF A$ < > "Y" THEN 5304
5306 HOME : VTAB 5: PRINT "ENTER COMMENT REQUIRED AND TYPE
     <RETURN>": PRINT : PRINT "DO NOT USE MORE THAN ONE
     LINE": VTAB 10: HTAB 40: INPUT T$
5307 RETURN
5312 IF  VAL (SM$(7)) = 0 THEN SM$(10) = "    ": GOTO 5315
5314 IF X$ = " SER EL " THEN SM$(10) =  STR$ ( INT ( VAL
     (SM$(8)) /  VAL (SM$(7)) * 1E2) / 1E2): FOR JJ = 1 TO 4:
     IF  LEN (SM$(10)) < 4 THEN SM$(10) = " " + SM$(10): NEXT
5315 IF ALTER$ = "Y" THEN NV = NV + 1: GOSUB 24094::Z$(AL) =
     "": FOR I = 1 TO NV:Z$(AL) = Z$(AL) + SM$(I): NEXT
     :Z$(AL) = Z$(AL) + T$: GOTO 5145
5317 PRINT
5318 NV = NV + 1: GOSUB 24094:NI = NI + 1:Z$(NI) = "": FOR I
     = 1 TO NV:Z$(NI) = Z$(NI) + SM$(I):SM$(I) = "": NEXT
     :Z$(NI) = Z$(NI) + T$: GOTO 2000
5319 REM **************
5320 REM * ENTER DATA *
5321 REM **************
5330 VTAB V: HTAB ZZ: GET IN$(II):ZZ = ZZ + 1: PRINT IN$(II);
5332 IF IN$(II) < > CHR$ (13) THEN 5365
5340 IF ZZ = 2 THEN  POP : GOTO 5130
5344 IN$ = "": FOR JJ = 1 TO II - 1:IN$ = IN$ + IN$(JJ): NEXT
     : FOR JJ = 1 TO L(J + E): IF J + E > 1 AND  LEN (IN$) <
     L(J + E) THEN IN$ = " " + IN$: NEXT
5345 IF J + E > 1 AND  LEN (IN$) > L(J + E) THEN  VTAB V:
     GOTO 5380
5346 IF J = 0 AND  MID$ (IN$,2,1) = "/" OR  MID$ (IN$,3,1) =
     "/" THEN  IF  LEN (IN$) < 6 THEN 5348
5347 IF J = 0 THEN  VTAB V: HTAB 1: PRINT "DATE SHOULD BE
     DD/MM";: FOR JJ = 1 TO 3000: NEXT : VTAB V: HTAB 1: FOR
     JJ = 1 TO 20: PRINT " ";: NEXT : GOTO 5380
5348 IF J = 1 AND  MID$ (IN$,2,1) = "." OR  MID$ (IN$,3,1) =
```

```
     "." THEN  IF  LEN (IN$) < 6 THEN 5350
5349 IF J = 1 THEN  VTAB V: HTAB 8: PRINT "TIME SHOULD BE
     HH.MM";: FOR JJ = 1 TO 3000: NEXT : VTAB V: HTAB 8: FOR
     JJ = 1 TO 20: PRINT " ";: NEXT : GOTO 5380
5350 VTAB V: HTAB H: PRINT IN$:SM$(J + E) = IN$:IN$ = "":II =
     1: GOTO 5490
5365 IF IN$(II) =  CHR$ (8) AND H = 1 THEN  HTAB H: PRINT
     SPC( 39): POP : GOTO 5270
5366 IF IN$(II) =  CHR$ (8) THEN 5375
5367 IF IN$(II) = "-" OR IN$(II) = "/" OR IN$(II) = "." THEN
     5370
5368 IF IN$(II) < "0" OR IN$(II) > "9" THEN ZZ = ZZ - 1: HTAB
     ZZ: PRINT " ": GOTO 5330
5370 II = II + 1: GOTO 5330
5375 IF V = 15 AND ZZ = 16 THEN ZZ = 37:H = 37:E = 0:J = 7:
     VTAB 3: HTAB H: PRINT  SPC( 3):J = J - 1: GOTO 5293
5380 IF II > 1 THEN II = 1:ZZ = H: HTAB ZZ: PRINT  SPC( 5):
     GOTO 5330
5387 REM ****************
5388 REM * DATA TAB POS *
5389 REM ****************
5390 IF H = R(0) THEN H = 1
5400 IF H = R(1) THEN H = R(0)
5410 IF H = R(2) THEN H = R(1)
5420 IF H = R(3) THEN H = R(2)
5430 IF H = R(4) THEN H = R(3)
5440 IF H = R(5) THEN H = R(4)
5460 ZZ = H
5470 HTAB H: PRINT  SPC( 6)
5480 J = J - 2: RETURN
5490 K = 0
5500 IF J = K THEN H = R(K):ZZ = H
5510 IF K = 5 THEN  RETURN
5520 K = K + 1: GOTO 5500
5997 REM *****************
5998 REM * PRINT ROUTINE *
5999 REM *****************
6000 HOME : VTAB 5:Z$ = "USE A NEW SHEET OF PAPER": GOSUB 8:
```

```
     VTAB 10:Z$ = "FOR EACH PATIENT": GOSUB 8: PRINT  CHR$
     (7); CHR$ (7)
6005 VTAB 20:Z$ = "** TYPE <SPACEBAR> WHEN READY **": GOSUB
     8: VTAB 22: HTAB 20: GET A$: PRINT A$: IF A$ < > " "
     THEN 6005
6008 PRINT D$;"PR#1": PRINT : PRINT : PRINT NA$;: HTAB 20:
     PRINT NO$: PRINT : PRINT
6010 DT$ = "": GOSUB 6020: GOTO 6120
6020 IF X$ = " CVS " THEN  PRINT " DATE    TIME    HR   SYS
     DIA   CVP   PAWP  MEAN  CORE  DIFF"
6030 IF X$ = " CVS " THEN Y(7) = 22:Z(7) = 3:Y(8) = 25:Z(8) =
     4:Y(9) = 29:Z(9) = 3:NV = 9:N1 = 9: RETURN
6040 IF X$ = " RS " THEN  PRINT " DATE    TIME  RATE  IN VT
     EX VT  V MIN PAUSE PEEP COMP   WORK   RESIST"
6060 IF X$ = " SER EL " THEN  PRINT "DATE    TIME     NA    K
     CL    UREA  CREAT  P OS  U OS   GLUC"
6070 IF X$ = " HAEM " THEN  PRINT "DATE    TIME     HB    PCV
     WBC  PLATLETS":NV = 5
6080 IF X$ = " BL GAS " THEN  PRINT "DATE    TIME     FIO2
     PO2    PCO2   PH     BE":NV = 6
6100 IF X$ = " LFT'S " THEN  PRINT "DATE    TIME     ALB
     GLOB   BIL    ASP   ALA  ALK PHOS":NV = 7
6110 IF X$ = " TEXT " THEN  PRINT "TEXT COMMENTS":NV = 2:L(2)
     = 40
6112 IF X$ = " FL IN " THEN  PRINT "DATE     TIME     N/G
     IV1    IV2    IV3   DRUG   PLAT  PLAS  UNITS BLOOD":NV =
     9
6114 IF X$ = " FL OUT " THEN  PRINT "DATE    TIME     N/G
     URINE  L-CHEST-R    -1-     -2-    -3- ABDOM DRAINS":NV =
     8
6115 RETURN
6120 FOR X = NI TO 1 STEP  - 1
6130 FOR AB = 1 TO NVINSTR
6140 SM$(AB) =  MID$ (Z$(X),S(AB),L(AB)): IF SM$(AB) = "-99"
     THEN SM$(AB) = "---"
6150 NEXT AB
6160 ET =  VAL (SM$(1)): GOSUB 25000: GOSUB 26000
6170 HTAB 1: IF DT$ < > SM$(0) THEN  PRINT SM$(0);:DT$ =
```

```
     SM$(0): GOTO 6190
6180 PRINT "     ";
6190 FOR J = 0 TO NV: PRINT "    ";: PRINT SM$(J + 1);: NEXT
     J:T$ =  MID$ (Z$(X),S(NV) + L(NV)): IF  ASC (T$) < 46
     THEN 6200
6195 PRINT : HTAB 17: PRINT T$
6200 PRINT : NEXT X
6210 PRINT : PRINT : PRINT : PRINT D$;"PR#0": RETURN
12000 REM **********************
12010 REM * CALCULATE FLUID BAL *
12015 REM **********************
12020 X$ = " FL IN ": GOSUB 12030
12022 X$ = " FL OUT ": GOSUB 12030
12024 RETURN
12030 Z$ = N$ + X$ + "DATALEN
12035 FOR I = 1 TO NI:Z$(I) = "": NEXT :NI = 0
12040 ONERR  GOTO 3014
12050 PRINT D$;"OPEN";Z$: PRINT D$;"READ";Z$: INPUT NI: PRINT
      D$;"CLOSE";Z$
12060 POKE 216,0:Z$ = N$ + X$ + "DATA
12070 PRINT D$;"OPEN";Z$: PRINT D$;"READ";Z$: FOR I = 1 TO
      NI: INPUT Z$(I): NEXT
12080 PRINT D$;"CLOSE";Z$
12100 IF X$ = " FL OUT " THEN 13015
12998 REM * CALCULATE DATA STRINGS NEEDED *
13000 FILE = NI:EX = 0:NH = 0:NM = 0:ND = 0: HOME :F2 = NI:PI
      = 1
13001 HOME : PRINT "HOW FAR BACK DO YOU WISH TO START?":
      PRINT : PRINT "TYPE THE ELAPSED TIME": GOSUB 13003:NM =
      Z: GOTO 13012
13003 VTAB 5: PRINT "FOLLOWED BY THE UNITS AND <RETURN>":
      VTAB 8: PRINT "EG  10M=10MINS  5H=5HOURS  2D=2DAYS":
      VTAB 12: HTAB 19: INPUT TI$: IF  VAL (TI$) <  = 0 THEN
      13003
13004 HOME : IF  RIGHT$ (TI$,1) = "D" THEN Z =  VAL ( LEFT$
      (TI$, LEN (TI$) - 1)) * 1440: RETURN
13006 IF  RIGHT$ (TI$,1) = "H" THEN Z =  VAL ( LEFT$ (TI$,
      LEN (TI$) - 1)) * 60: RETURN
```

```
13008 IF  RIGHT$ (TI$,1) = "M" THEN Z =  VAL ( LEFT$ (TI$,
      LEN (TI$) - 1)): RETURN
13010 POP : GOTO 13001
13012 Z9 = 43200: REM 1 MONTH
13015 START =  FN TIME(FI) - NM
13017 FOR I = 1 TO FILE: IF START >  FN TIME(I) THEN  NEXT
13019 START =  FN TIME(I):PI = I
13021 LAST = ST + Z9: FOR I = FI TO 1 STEP  - 1: IF LAST <
      FN TIME(I) THEN  NEXT
13022 F2 = I
13023 LAST =  FN TIME(F2): GOTO 13025
13024 START =  FN TIME(PI):LAST =  FN TIME(FILE): HOME
13025 NM = LAST - START
13027 ET = ST: GOSUB 25000: GOSUB 26000:SD = DY:MS = MT:SH =
      H:SM = M
13028 ET = LA: GOSUB 25000: GOSUB 26000:LM$ =  STR$ (MT): IF
      LEN (LM$) < 2 THEN LM$ = "0" + LM$
13030 DY$ =  STR$ (DY): IF  LEN (DY$) < 2 THEN DY$ = "0" +
      DY$
13050 IF NM > 2 THEN 13080
13070 IF PLT = 1 THEN  HOME : PRINT " NOT ENOUGH VALUES TO
      PLOT": FOR II = 0 TO 1000: NEXT : POP : RETURN
13080 GOSUB 5010: HOME : VTAB 15: PRINT "OBTAINING DATA"
13090 IF X$ = " FL IN " THEN 13110
13100 IF X$ = " FL OUT " THEN 13160
13110 FOR I = 1 TO 10:IT(I) = 0: NEXT
13115 FOR I = PI TO F2
13120 FOR AB = 1 TO NV
13130 IT(AB) = IT(AB) +  VAL ( MID$ (Z$(I),S(AB),L(AB)))
13135 NEXT AB,I
13140 RETURN
13160 FOR I = 1 TO 10:OT(I) = 0: NEXT
13165 FOR I = 1 TO NI
13170 FOR AB = 1 TO NV
13180 OT(AB) = OT(AB) +  VAL ( MID$ (Z$(I),S(AB),L(AB)))
13190 NEXT AB,I
13200 HOME : PRINT NA$;: HTAB 17: PRINT "TOTALS/BALANCE OVER
      ";TI$
```

```
13210  VTAB 3: PRINT "INPUT(ML)";: HTAB 16: PRINT
       "OUTPUT(ML)";: HTAB 29: PRINT "BALANCE(ML)
13215  VTAB 4: PRINT "---------";: HTAB 16: PRINT
       "----------";: HTAB 29: PRINT "-----------
13220  VTAB 5: PRINT "N/G   ";IT(2);: HTAB 16: PRINT "N/G
       ";OT(2);: HTAB 29: PRINT "N/G   ";IT(2) - OT(2)
13230  VTAB 7: PRINT "IV1   ";IT(3): VTAB 8: PRINT "IV2
       ";IT(4): VTAB 9: PRINT "IV3   ";IT(5)
13240  VTAB 10: PRINT "DRUG  ";IT(6)
13250  VTAB 11: PRINT "---------
13260  T = IT(3) + IT(4) + IT(5) + IT(6): VTAB 12: PRINT
       "CRYST";T;
13270  HTAB 16: PRINT "URINE ";OT(3);: HTAB 29: PRINT "CRYST
       ";T - OT(3)
13280  VTAB 13: HTAB 29: PRINT "----------
13290  VTAB 14: HTAB 35:B = T + IT(2) - OT(3) - OT(2): PRINT B
13310  VTAB 15: HTAB 16: PRINT "CHEST L ";OT(4): VTAB 16: HTAB
       16: PRINT "CHEST R ";OT(5): PRINT
13312  VTAB 18: PRINT "PLAT'S  ";IT(7);"ML";
13315  VTAB 18: HTAB 16: PRINT "ABDOM 1 ";OT(6)
13320  VTAB 19: PRINT "PLASMA ";IT(8);"ML";: HTAB 16: PRINT
       "ABDOM 2 ";OT(7)
13330  VTAB 20: PRINT "BLOOD  ";IT(9);"UNITS";: HTAB 16: PRINT
       "ABDOM 3 ";OT(8)
13380  VTAB 21: PRINT "-------------";: HTAB 16: PRINT
       "------------";: HTAB 34: PRINT "------"
13390  IT = IT(7) + IT(8) + IT(9) * 400: VTAB 22: PRINT "TOTAL
       ";IT;
13400  OT = OT(4) + OT(5) + OT(6) + OT(7) + OT(8): HTAB 16:
       PRINT "TOTAL   ";OT;
13410  HTAB 34: PRINT IT - OT
13420  VTAB 24: HTAB 20: INVERSE : PRINT "TOTAL BALANCE ";:
       NORMAL : PRINT B + IT - OT;
13500  VTAB 23: PRINT : PRINT D$;"PR#0"
14000  VTAB 24: HTAB 1: PRINT "PRINT OUT Y/N ";: GET A$: IF A$
       = "N" THEN 14030
14010  IF A$ = "Y" THEN  VTAB 22: PRINT : PRINT D$;"PR#1":
       PRINT : PRINT : GOTO 13200
```

```
14020  GOTO 14000
14030  VTAB 24: HTAB 1: PRINT "FURTHER BALANCES FOR THIS
       PATIENT? Y/N ";: GET A$: IF A$ = "N" THEN  RETURN
14040  IF A$ = "Y" THEN  VTAB 22: PRINT : GOTO 12000
14050  GOTO 14000
24090  REM **************
24094  REM * DATE->MINS *
24096  REM **************
24110  IF  MID$ (SM$(0),2,1) = "/" THEN DTD$ =  LEFT$
       (SM$(0),1): GOTO 24130
24120  DTD$ =  LEFT$ (SM$(0),2)
24130  DTD =  VAL (DTD$)
24135  POKE 216,0
24140  MT$ =  RIGHT$ (SM$(0), LEN (SM$(0)) -  LEN (DTD$) - 1):
       IF  LEN (MT$) = 1 THEN MT$ = "0" + MT$
24145  MT =  VAL (MT$)
24150  IF  MID$ (SM$(1),2,1) = "." THEN H$ =  LEFT$
       (SM$(1),1): GOTO 24170
24160  H$ =  LEFT$ (SM$(1),2)
24170  H =  VAL (H$)
24180  M$ =  RIGHT$ (SM$(1), LEN (SM$(1)) -  LEN (H$) - 1): IF
       LEN (M$) = 1 THEN M$ = "0" + M$
24185  M =  VAL (M$)
24190  D = 0: IF MT = 1 THEN 24220
24210  RESTORE
24215  FOR I = 1 TO MT: READ J:DTD = DTD + J: NEXT
24220  DTD = DTD + D
24230  IF MT = 2 AND L = 1 THEN DTD = DTD + 1
24240  SM$(1) =  STR$ (DTD * 1440 + H * 60 + M)
24245  FOR II = 1 TO 6: IF  LEN (SM$(1)) < 6 THEN SM$(1) = "0"
       + SM$(1): NEXT
24250  RETURN
25000  REM *****************
25010  REM * MIN->DAYS/HRS *
25015  REM *****************
25020  MT = 0
25040  DY =  INT (ET / 1440):ET = ET - DY * 1440
25060  H =  INT (ET / 60):M = ET - H * 60: RETURN
```

```
26000 REM **************
26005 REM * DAYS->DATE *
26015 REM **************
26010 IF L = 1 AND DY = 60 THEN MT = 2:DY = 29: GOTO 26070
26015 IF L = 1 AND DY > 60 THEN DY = DY - 1
26017 RESTORE : READ J
26020 FOR II = 1 TO 12: READ J:MT = MT + 1
26030 IF DY - J < 1 THEN ND = DY: GOTO 26070
26040 DY = DY - J: NEXT
26070 ND$ =  STR$ (ND): IF  LEN (ND$) < 2 THEN ND$ = " " +
      ND$
26072 MT$ =  STR$ (MT): IF  LEN (MT$) < 2 THEN MT$ = "0" +
      MT$
26080 SM$(0) = ND$ + "/" + MT$:H$ =  STR$ (H): IF  LEN (H$) <
      2 THEN H$ = " " + H$
26082 M$ =  STR$ (M): IF  LEN (M$) < 2 THEN M$ = "0" + M$
26085 SM$(1) = H$ + "." + M$:ND$ = "":MT$ = "":H$ = "":M$ =
      "": RETURN
26087 DATA  0,31,28,31,30,31,30,31,31,30,31,30,31
26999 REM *****************
27000 REM * GET CLOCK TIME *
27001 REM *****************
27010 D$ =  CHR$ (4): PRINT D$:SLOT = 3
27030 PRINT D$;"IN#";SL: PRINT D$;"PR#";SL: INPUT T$: PRINT
      D$;"IN#0": PRINT D$;"PR#0"
27040 MT =  VAL ( MID$ (T$,1,2))
27050 D =  VAL ( MID$ (T$,4,2))
27060 H =  VAL ( MID$ (T$,7,2))
27070 M =  VAL ( MID$ (T$,10,2))
27080 S =  VAL ( MID$ (T$,13,6))
27090 RESTORE
27100 DTD = 0
27110 FOR II = 1 TO MT: READ J:DTD = DTD + J: NEXT
27130 DTD = DTD + D
27140 IF MT > 2 AND L = 1 THEN DTD = DTD + 1
27150 STD = DTD * 86400 + H * 3600 + M * 60 + S
27160 S = STD: RETURN
```

```
10  REM ***************
12  REM * FILEADDNAME *
15  REM ***************
30  D$ =  CHR$ (4): GOTO 100
39  REM ********************
40  REM * ADD LEADING ZEROS *
41  REM ********************
42  IF  LEN (TT$) < TT THEN TT$ = "0" + TT$: GOTO 42
44  RETURN
49  REM ********************
50  REM * ADD TRAILING DOTS *
51  REM ********************
52  IF  LEN (TT$) < TT THEN TT$ = TT$ + ".": GOTO 52
54  RETURN
97  REM ********
98  REM * MENU *
99  REM ********
100  HOME :Z$ = "ADD / ALTER NAME FILE": PRINT Z$: FOR I = 1
     TO  LEN (Z$): PRINT "=";: NEXT
102  PRINT
105  VTAB 5: PRINT "1. ADD NAME TO FILE
110  VTAB 7: PRINT "2. ALTER / DELETE NAME ON FILE
120  VTAB 9: PRINT "3. RETURN TO MAIN PROGRAM
125  VTAB 15: PRINT "SELECT REQUIRED NUMBER  ";
130  GET A$: PRINT A$: IF A$ < "1" OR A$ > "3" THEN 100
140  IF A$ = "1" THEN 1200
150  IF A$ = "3" THEN  PRINT D$;"BLOAD CHAIN,A520": CALL
     520"PART1NAMESORT"
1000  REM **************
1001  REM * LIST NAMES *
1002  REM **************
1010  ONERR  GOTO 1014
1012  GOTO 1018
1014  HOME :Z$ = "NO NAMES ON LIST": VTAB 10: HTAB (40 -  LEN
      (Z$)) / 2: PRINT Z$: FOR I = 1 TO 2000: NEXT
1016  POKE 216,0: GOTO 100
1018  NA$ = "NAMELISTLEN": PRINT D$;"OPEN";NA$
1019  PRINT D$;"READ";NA$: INPUT NI
```

```
1020  PRINT D$;"CLOSE";NA$
1021  NA$ = "NAMELIST"
1022  PRINT D$;"OPEN";NA$
1023  PRINT D$;"READ";NA$
1024  FOR I = 1 TO NI: INPUT Z$(I): NEXT
1026  PRINT D$;"CLOSE";NA$
1030  POKE 216,0
1035  HOME : PRINT "  NAME";: HTAB 23: PRINT "HOSPITAL NO. ":
      PRINT "  ====";: HTAB 23: PRINT "===========
1038  PRINT
1040  FOR J = 1 TO NI: PRINT : PRINT J;". ";:NA$ =  MID$
      (Z$(J),2,20): PRINT NA$;: HTAB 25:NO$ =  MID$
      (Z$(J),22,6): PRINT NO$: NEXT
1050  VTAB 22: HTAB 1: PRINT "SELECT PATIENT NO. FOR
      CORRECTION"
1060  VTAB 24: HTAB 1: PRINT "TYPE 'M' TO RETURN MENU ";
1065  GET A$: IF A$ = "M" THEN 100
1070  IF A$ < "1" OR A$ >  STR$ (NI) THEN 1065
1080  GOTO 5000
1200  REM ************
1201  REM * ADD NAME *
1202  REM ************
1210  HOME : VTAB 1: HTAB 1: PRINT "TYPE 'SURNAME' <SPACE>
      'INITIAL'": PRINT : PRINT "THEN TYPE <RETURN>   ": VTAB
      5: INPUT NA$
1214  IF  LEN (NA$) > 20 THEN 1220
1215  FOR I = 1 TO 20:A$ =  MID$ (NA$,I,1): IF A$ < "0" OR A$
      > "9" THEN  NEXT : GOTO 1217
1216  GOTO 1220
1217  FOR I = 3 TO 20: IF  MID$ (NA$,I,1) = " " THEN 1225
1218  NEXT
1220  FOR I = 1 TO  LEN (NA$): PRINT " ";: PRINT : PRINT "ONLY
      20 CHARS INCLUDING 1 SPACE ALLOWED": FOR I = 1 TO 2000:
      NEXT : GOTO 1210
1225  TT$ = NA$:TT = 20: GOSUB 50:NA$ = TT$
1230  VTAB 10: HTAB 33: FOR I = 1 TO  LEN (NO$) + 1: PRINT "
      ";: NEXT : PRINT
1235  VTAB 10: PRINT "TYPE HOSPITAL NO. THEN <RETURN> ";:
```

```
      INPUT NO$
1240  IF  LEN (NO$) > 6 OR  LEN (NO$) < 3 THEN  VTAB 10: HTAB
      32: FOR I = 1 TO 1 +  LEN (NO$): PRINT " ";: NEXT :
      PRINT : VTAB 10: PRINT "MIN OF 3 & MAX OF 6 CHARACTERS
      ALLOWED": FOR I = 1 TO 3000: NEXT : GOTO 1230
1245  TT$ = NO$:TT = 6: GOSUB 40:NO$ = TT$
1250  VTAB 15: HTAB 33: FOR I = 1 TO  LEN (BW$) + 2: PRINT "
      ";: NEXT : PRINT
1255  VTAB 15: HTAB 1: PRINT "TYPE BODY WEIGHT THEN <RETURN>
      ";: INPUT BW$
1260  IF  LEN (BW$) > 3 OR  LEN (BW$) < 2 THEN  VTAB 15: HTAB
      33: FOR I = 1 TO 1 +  LEN (NO$): PRINT " ";: NEXT :
      PRINT : VTAB 15: PRINT "MIN OF 3 & MAX OF 6 CHARS ALLOWED"
      : FOR I = 1 TO 2000: NEXT : GOTO 1250
1265  TT$ = BW$:TT = 3: GOSUB 40:BW$ = TT$
1270  N$ = "": FOR I = 1 TO 20: IF  MID$ (NA$,I,1) < > "."
      THEN N$ = N$ +  MID$ (NA$,I,1): NEXT
1280  Z$ = "NAMELISTLEN"
1285  X$ = ""
1287  ONERR  GOTO 1320
1290  PRINT D$;"OPEN";Z$
1300  PRINT D$;"READ";Z$
1305  INPUT NI
1310  PRINT D$;"CLOSE";Z$
1320  NI = NI + 1
1325  PRINT D$;"OPEN";Z$
1330  PRINT D$;"WRITE";Z$
1332  PRINT NI
1334  PRINT D$;"CLOSE";Z$
1336  POKE 216,0
1338  Z$ = "NAMELIST"
1340  PRINT D$;"OPEN";Z$
1345  PRINT D$;"POSITION";Z$",R";NI-1
1350  PRINT D$;"WRITE";Z$
1355  PRINT "X" + NA$ + NO$: REM * 'X' USED FOR INTER-APPLE
      COMMUNICATION *
1360  PRINT D$;"CLOSE";Z$
1370  GOTO 100
```

```
5000  VTAB 22: HTAB 1: FOR I = 1 TO 38: PRINT " ";: NEXT
5010  VTAB 24: HTAB 1: PRINT "TYPE 'D' TO D(ELETE PATIENT
      NUMBER "; VAL (A$);" ";: GET Z$: IF Z$ < > "D" THEN 100
5020  II =  VAL (A$)
5030  FOR JJ = II TO NI:Z$(JJ) = Z$(JJ + 1): NEXT :Z$(NI) =
      "":NI = NI - 1
5040  VTAB 24: HTAB 1: PRINT "    WRITE CORRECTED FILE TO
      DISC? Y/N ";: GET A$: IF A$ = "N" THEN 100
5050  IF A$ < > "Y" THEN 5040
5055  PRINT D$
5060  Z$ = "NAMELISTLEN"
5070  PRINT D$;"OPEN";Z$: PRINT D$;"WRITE";Z$: PRINT NI: PRINT
      D$;"CLOSE"
5080  Z$ = "NAMELIST"
5085  PRINT D$;"DELETE";Z$
5090  PRINT D$;"OPEN";Z$: PRINT D$;"WRITE";Z$: FOR I = 1 TO
      NI: PRINT Z$(I): NEXT : PRINT D$;"CLOSE"
6000  GOTO 100
```

```
2  REM ****************
3  REM * FILENAMESORT *
4  REM ****************
10  D$ =  CHR$ (4)
84  HOME
85  VTAB 15: HTAB 10: PRINT "OBTAINING DATA FILE "
87  Z$ = "NAMELISTLEN"
90  PRINT D$;"OPEN";Z$
95  PRINT D$;"READ";Z$
100  INPUT NI
110  PRINT D$;"CLOSE";Z$
112  Z$ = "NAMELIST"
115  PRINT D$;"OPEN";Z$
120  PRINT D$;"READ";Z$
130  FOR I = 1 TO NI: INPUT Z$(I): NEXT
180  PRINT D$;"CLOSE";Z$
190  GOSUB 15000
270  HOME
280  VTAB 15: HTAB 10: PRINT "WRITING REARRANGED FILES"
290  PRINT D$;"OPEN";Z$
300  PRINT D$;"WRITE"Z$
310  FOR I = 1 TO NI: PRINT Z$(I): NEXT
360  PRINT D$;"CLOSE"
370  TEXT
375  X$ = ""
380  HOME :Z$ = "GLASGOW ROYAL INFIRMARY": GOSUB 400:Z$ =
     "RESPIRATORY INTENSIVE CARE UNIT": GOSUB 400
390  Z$ = "FILE PROGRAM ": GOSUB 400: GOTO 410
400  PRINT : HTAB (40 -  LEN (Z$)) / 2: PRINT Z$: RETURN
410  PRINT D$;"BLOAD CHAIN,A520": CALL 520"PART1"
15000  REM ************
15001  REM * SORT SUB *
15002  REM ************
15010  HOME : VTAB 15: HTAB 15: PRINT "SORTING"
15050  N = NI
15060  REM ** SORT **
15070  F = 0:I = 0
15080  IF Z$(I) <  = Z$(I + 1) THEN 15130
```

```
15090  T$ = Z$(I + 1)
15100  Z$(I + 1) = Z$(I)
15110  Z$(I) = T$
15120  F = 1
15130  I = I + 1: IF I <  = N THEN 15080
15140  IF F = 1 THEN 15070
15150  FOR I = 2 TO NI + 1:Z$(I - 1) = Z$(I): NEXT
15160  RETURN
```

```
1  REM ************
2  REM * FILEPLOT *
3  REM ************
5  GOSUB 7: TEXT : GOTO 30
7  POKE  - 16368,0: FOR I = 1 TO 500: NEXT : RETURN
30  D$ =  CHR$ (4)
40  DEF  FN TIME(X) =  VAL ( LEFT$ (Z$(X),6)): DEF  FN PLT(X)
    =  VAL ( MID$ (Z$(X),Z0,Z1))
50  PRINT D$;"BLOAD CHARGEN A$6000 L$100
60  PRINT D$;"BLOAD COLOSSAL.SET A$6900 L$300
70  PLT = 0: GOSUB 13000:PLT = 1
97  REM  ********
98  REM  * MENU *
99  REM  ********
100  I = 0: HOME : PRINT "MENU";: HTAB 20: PRINT X$: PRINT
     "===="
105  IF N$ <  > "" THEN  VTAB 1: HTAB 1: PRINT N$;: HTAB 34:
     PRINT NO$:A$ = N$: GOSUB 108: HTAB 20:A$ = X$: GOSUB 108:
     HTAB 34:A$ = NO$: GOSUB 108
106  GOTO 110
108  FOR I = 1 TO  LEN (A$): PRINT "=";: NEXT : RETURN
110  V = 7: VTAB V: HTAB 1: PRINT "1. LIST NAMES ON FILE"
112  VTAB V + 3: HTAB 1: PRINT "2. ADD / ALTER NAME ON FILE"
114  IF X$ = "" THEN ZZ$ = "2": GOTO 140
116  ZZ$ = "5"
120  VTAB V + 6: HTAB 1: PRINT "3. DISPLAY / STORE DATA"
130  VTAB V + 9: HTAB 1: PRINT "4. TREND PLOT"
132  VTAB V + 12: HTAB 1: PRINT "5. SELECT DIFFERENT SYSTEM"
140  VTAB 24: GOSUB 7
145  GET A$: VTAB 3: PRINT : IF A$ = "1" THEN 1000
147  IF A$ = "2" THEN 1200
149  IF A$ = "3" THEN  PRINT : PRINT D$;"BLOAD CHAIN,A520":
     CALL 520"PART1"
150  IF A$ = "4" THEN  GOSUB 13000: GOTO 1095
152  IF A$ = "5" THEN 1095
160  GOTO 100
```

```
997  REM *************
998  REM * NAME LIST *
999  REM *************
1000  ONERR  GOTO 1014
1012  GOTO 1018
1014  HOME :Z$ = "NO NAMES ON LIST": VTAB 10: HTAB (40 -  LEN
      (Z$)) / 2: PRINT Z$: FOR I = 1 TO 2000: NEXT
1016  POKE 216,0: GOTO 100
1018  NA$ = "NAMELISTLEN": PRINT D$;"OPEN";NA$
1019  PRINT D$;"READ";NA$: INPUT NI
1020  PRINT D$;"CLOSE";NA$
1021  NA$ = "NAMELIST"
1022  PRINT D$;"OPEN";NA$: PRINT D$;"READ";NA$
1024  FOR I = 1 TO NI: INPUT Z$(I): NEXT : PRINT
      D$;"CLOSE";NA$
1030  HOME : PRINT "FILE";
1035  HTAB 8: PRINT "NAME";: HTAB 25: PRINT "HOSPITAL NO.":
      PRINT "====";: HTAB 8: PRINT "====";: HTAB 25: PRINT
      "============": PRINT
1040  FOR J = 1 TO NI: PRINT : PRINT " ";J;".   ";:NA$ =  MID$
      (Z$(J),2,20): PRINT NA$;: HTAB 28:NO$ =  MID$
      (Z$(J),22,6): PRINT NO$: NEXT
1050  VTAB 22: HTAB 1: PRINT "SELECT FILE NUMBER FOR DATA
      RETRIEVAL"
1060  VTAB 24: HTAB 1: PRINT "TYPE 'M' TO RETURN MENU ";
1065  GOSUB 7: GET A$: IF A$ = "M" THEN 100
1070  IF A$ < "1" OR A$ > "9" THEN 1065
1075  A =  VAL (A$)
1090  NA$ =  MID$ (Z$(A),2,20):NO$ =  MID$ (Z$(A),22,6)
1092  N$ = "": FOR I = 1 TO 20: IF  MID$ (NA$,I,1) <  > "."
      THEN N$ = N$ +  MID$ (NA$,I,1): NEXT
1093  REM ***************
1094  REM * SYSTEM LIST *
1095  REM ***************
1096  HOME : PRINT N$;: HTAB 30: PRINT NO$
1097  FOR I = 1 TO  LEN (N$): PRINT "=";: NEXT : HTAB 30: FOR
      I = 1 TO 6: PRINT "=";: NEXT
1100  VTAB 6: HTAB 1: PRINT "1. CARDIOVASCULAR";: HTAB 24:
```

```
      PRINT "6. BLOOD GASES
1110  VTAB 8: PRINT "2. RESPIRATORY";: HTAB 24: PRINT "7. UREA
      & ELECTS
1120  VTAB 10: PRINT "3. FLUID INPUT";: HTAB 24: PRINT "8.
      LFT'S
1130  VTAB 12: PRINT "4. FLUID OUTPUT";: HTAB 24: PRINT "9.
      HAEMATOLOGY
1140  VTAB 14: PRINT "5. FLUID BALANCE";: HTAB 24: PRINT "M.
      RETURN TO MENU
1148  FOR I = 1 TO 10:SM$(I) = "": NEXT
1160  VTAB 20: HTAB 1: PRINT "SELECT SYSTEM REQUIRED    ";:
      GOSUB 7: GET A$: IF A$ = "M" THEN 100
1165  IF A$ < "1" OR A$ > "9" THEN 1160
1170  PRINT A$: ON  VAL (A$) GOTO
      1171,1172,1173,1173,1173,1176,1177,1178,1179
1171  X$ = " CVS ": GOTO 1190
1172  X$ = " RS ": GOTO 1190
1173  HOME : VTAB 10: PRINT "TREND PLOTS OF FLUIDS NOT
      AVAILABLE": FOR I = 1 TO 2000: NEXT : GOTO 100
1176  X$ = " BL GAS ": GOTO 1190
1177  X$ = " SER EL ": GOTO 1190
1178  X$ = " LFT'S ": GOTO 1190
1179  X$ = " HAEM "
1190  NI = 0: GOTO 3000
1200  REM * ADDNAME *
1210  PRINT D$;"BLOAD CHAIN,A520": CALL 520"FILEPLOTADDNAME"
3000  REM  * RETRIEVE DATA *
3006  FOR I = 1 TO NI:Z$(I) = "": NEXT :NI = 0
3010  Z$ = N$ + X$ + "DATALEN
3012  ONERR  GOTO 3014
3013  GOTO 3020
3014  POKE 216,0: PRINT D$;"DELETE";Z$: HOME :Z$ = "NO DATA
      AVAILABLE": VTAB 10: HTAB (40 -  LEN (Z$)) / 2: PRINT
      Z$: FOR I = 1 TO 2000: NEXT : GOTO 100
3020  PRINT D$;"OPEN";Z$: PRINT D$;"READ";Z$: INPUT NI: PRINT
      D$;"CLOSE";Z$
3030  POKE 216,0:Z$ = N$ + X$ + "DATA
3040  PRINT D$;"OPEN";Z$: PRINT D$;"READ";Z$: FOR I = 1 TO NI:
```

```
      INPUT Z$(I): NEXT
3042  PRINT D$;"CLOSE";Z$: GOTO 100
12997 REM ******************
12998 REM * TREND GRAPHICS *
12999 REM ******************
13000 FILE = NI:EX = 0:NH = 0:NM = 0:ND = 0: HOME :F2 = NI:PI
      = 1: GOTO 13024
13001 HOME : PRINT "HOW FAR BACK DO YOU WISH TO START PLOT?":
      PRINT : PRINT "TYPE THE ELAPSED TIME": GOSUB 13003:NM =
      Z: GOTO 13012
13003 VTAB 5: PRINT "FOLLOWED BY THE UNITS AND <RETURN>":
      VTAB 8: PRINT "EG  10M=10MINS  5H=5HOURS  2D=2DAYS":
      VTAB 12: HTAB 19: INPUT TI$: IF  VAL (TI$) <  = 0 THEN
      13003
13004 HOME : IF  RIGHT$ (TI$,1) = "D" THEN Z =  VAL ( LEFT$
      (TI$, LEN (TI$) - 1)) * 1440: RETURN
13006 IF  RIGHT$ (TI$,1) = "H" THEN Z =  VAL ( LEFT$ (TI$,
      LEN (TI$) - 1)) * 60: RETURN
13008 IF  RIGHT$ (TI$,1) = "M" THEN Z =  VAL ( LEFT$ (TI$,
      LEN (TI$) - 1)): RETURN
13010 POP : GOTO 13001
13012 HOME : PRINT "TYPE TIME INTERVAL TO BE PLOTTED": GOSUB
      13003:Z9 = Z
13015 START =  FN TIME(FI) - NM
13017 FOR I = 1 TO FILE: IF START >  FN TIME(I) THEN  NEXT
13019 START =  FN TIME(I):PI = I
13021 LAST = ST + Z9: FOR I = FI TO 1 STEP  - 1: IF LAST <
      FN TIME(I) THEN  NEXT
13022 F2 = I
13023 LAST =  FN TIME(F2): GOTO 13025
13024 START =  FN TIME(PI):LAST =  FN TIME(FILE): HOME
13025 NM = LAST - START
13027 ET = ST: GOSUB 25000: GOSUB 26000:SD = DY:MS = MT:SH =
      H:SM = M
13028 ET = LA: GOSUB 25000: GOSUB 26000:LM$ =  STR$ (MT): IF
      LEN (LM$) < 2 THEN LM$ = "0" + LM$
13030 DY$ =  STR$ (DY): IF  LEN (DY$) < 2 THEN DY$ = "0" +
      DY$
```

```
13050  IF NM > 2 THEN 13250
13070  IF PLT = 1 THEN  HOME : PRINT " NOT ENOUGH VALUES TO
       PLOT": FOR II = 0 TO 1000: NEXT
13080  RETURN
13250  REM  * HI-RES PLOTS *
13260  GOSUB 13420
13262  IF X$ = " CVS " THEN 14700
13263  IF X$ = " RS " THEN 20000
13264  IF X$ = " SER EL " THEN 13610
13266  IF X$ = " HAEM " THEN 19000
13270  IF X$ = " BL GAS " THEN 19400
13275  IF X$ = " LFT'S " THEN 15610
13276  STOP
13280  AX = 0:CX = 0:PP = PI
13281  IF BX = 0 THEN BA = 30: GOTO 13290
13282  BA = 30 + 212 / NM * (BX + CX): REM   * BASE ADDRESS *
13290  X1 = BA
13295  IF PP = FI OR PP = F2 THEN  RETURN
13300  IF  FN PLT(PP) < LO THEN PP = PP + 1:CX = CX + ( FN
       TIME(PP) -  FN TIME(PP - 1)): GOTO 13282
13305  IF LG$ = "Y" THEN Y1 = 174 -  LOG ( FN PLT(PP)) /  LOG
       (10) * 164 / 3: GOTO 13312: REM * LOG CALCULATIONS *
13307  IF X$ = " CVS " OR X$ = " RS " THEN Y1 = 174 - ( FN
       PLT(PP) - BS) * 164 / (TS - BS): GOTO 13312: REM *
       ALTERED SCALING SYSTEM *
13310  Y1 = 190 - Z3 *  FN PLT(PP) + Z2
13312  IF Y1 < 0 THEN Y1 = 0
13313  IF Y1 > 270 THEN Y1 = 270
13314  IF X1 > 278 THEN X1 = 278
13316  IF X1 < 30 THEN X1 = 30
13320  FOR X = PP + 1 TO F2
13322  IF  FN PLT(X) < LO THEN AX = AX + 212 / NM * ( FN
       TIME(X) -  FN TIME(X - 1)): GOTO 13410
13325  IF LG$ = "Y" THEN Y2 = 174 -  LOG ( FN PLT(X)) /  LOG
       (10) * 164 / 3: GOTO 13349: REM * LOG CALCULATIONS *
13327  IF X$ = " CVS " OR X$ = " RS " THEN Y2 = 174 - ( FN
       PLT(X) - BS) * 164 / (TS - BS): GOTO 13349: REM *
       ALTERED SCALING *
```

```
13330  Y2 = 190 - Z3 *  FN PLT(X) + Z2
13349  X2 = X1 + 212 / NM * ( FN TIME(X) -  FN TIME(X - 1)) +
       AX
13375  IF X2 > 278 THEN X2 = 278
13376  IF Y2 < 0 THEN Y2 = 0
13377  IF X2 < 30 THEN X2 = 30
13378  AX = 0
13380  IF LBL$ = "" THEN 13396
13382  HCOLOR= 3: GOSUB 13384: GET A$: GOTO 13385
13384  HPLOT X2,177 TO X2,182 TO X2 + 1,182 TO X2 + 1,177 TO
       X2 - 2,179 TO X2 + 3,179 TO X2,177: RETURN
13385  IF A$ = "M" THEN MARK(LM + 1) = X2:LM = LM + 1: GOTO
       13400
13386  HCOLOR= 4: GOSUB 13384
13387  GOTO 13400
13396  HPLOT X1,Y1 TO X2,Y2
13398  HPLOT X1 + 1,Y1 + 1 TO X2 + 1,Y2 + 1
13400  X1 = X2:Y1 = Y2
13410  NEXT X: RETURN
```

```
13417  REM *************
13418  REM * AXES PLOT *
13419  REM *************
13420  GOTO 13430
13422  FOR I = 3 TO 7: IF N / I -  INT (N / I) < .01 THEN NX =
       2 * I: RETURN
13424  NEXT I:N = N + 1: GOTO 13422
13430  IF NM < 121 THEN NT$ = "MINS":N = NM: GOSUB 13422:BX =
       N - NM:NM = N: GOTO 13505
13450  IF NM < 2160 THEN BX = SM: IF NM + BX < 2160 THEN NT$ =
      "TIME": GOTO 13498
13470  NT$ = "DATE":BX = SH * 60 + SM
13490  ND =  INT ((NM + BX) / 1440) + 1
13492  N = ND: GOSUB 13422:ND = N:NM = ND * 1440
13493  NT$ = "DATE": GOTO 13505
13498  NH = (NM + BX) / 60: IF (NM + BX) / 60 -  INT ((NM +
       BX) / 60) <  > 0 THEN NH =  INT ((NM + BX) / 60) + 1
13500  N = NH: GOSUB 13422:NH = N:NM = NH * 60
13505  HGR2
13506  HCOLOR= 3: POKE 54,0: POKE 55,96: FOR II = 1 TO 39:
       VTAB 23: HTAB II: PRINT " ";: VTAB 24: HTAB II: PRINT "
       ";: NEXT
13508  FOR V = 0 TO NX:X = 30 + V * 212 / NX: HPLOT X,174 TO
       X,177 TO X + 1,177 TO X + 1,174: NEXT
13510  FOR V = 0 TO NX STEP 2:X = 30 + V * 212 / NX: HPLOT
       X,174 TO X,180 TO X + 1,180 TO X + 1,174: NEXT
13520  HPLOT 30,10 TO 30,180 TO 31,180 TO 31,10
13525  IF NT$ = "MINS" THEN LX = SM:NZ = NM: FOR V = 0 TO NX
       STEP 2: VTAB 24:X = V * 212 / NX:A$ =  STR$ ( INT (LX +
       V * NZ / NX + .5)):HT =  INT (X / 6.975 + 5.5 -  LEN (A$)
       / 2): GOTO 13542
13530  IF NT$ = "TIME" THEN LX = SH:BX = SM:NZ = NH: GOTO
       13540
13532  LX = SD:BX = SH * 60 + SM:NZ = ND
13540  FOR V = 0 TO NX STEP 2: VTAB 24:X = V * 212 / NX:A$ =
       STR$ ( INT (LX + V * NZ / NX + .5)):HT =  INT (X /
       6.975 + 5.5  -  LEN (A$) / 2)
13542  A =  VAL (A$)
```

```
13544  IF NT$ = "DATE" THEN 13548
13545  IF NT$ = "TIME" AND A > 24 THEN A = A - 24: GOTO 13545
13546  IF NT$ = "MINS" AND A > 60 THEN A = A - 60
13547  GOTO 13553
13548  RESTORE : READ J:II = 0
13549  II = II + 1: READ J: IF II < MS THEN 13549
13550  IF A > J THEN A = A - J
13551  IF A > J THEN 13549
13553  HT = HT + 1
13554  HTAB HT: PRINT A: NEXT
13555  VTAB 23: HTAB 37: PRINT NT$;: IF NT$ = "DATE" THEN
       VTAB 24: HTAB 38: PRINT "/";LM$;
13557  IF NT$ = "TIME" THEN  VTAB 24: HTAB 38: PRINT "/";DY$;
13560  RETURN
13600  REM ***********************
13610  REM * UREA & ELECTROLYTES *
13615  REM ***********************
13620  VTAB 5: HTAB 38: PRINT "NA";: VTAB 4: HTAB 2: PRINT
       "145": VTAB 6: HTAB 2: PRINT "135
13630  HCOLOR= 7: HPLOT 30,30 TO 245,30: HPLOT 245,40 TO 30,40
13636  Z0 = 7:Z1 = 3:Z2 =  - 15:Z3 = 1:LO = 100: GOSUB 13280
13638  HCOLOR= 5: HPLOT 30,70 TO 245,70: HPLOT 245,80 TO 30,80
13640  VTAB 10: HTAB 38: PRINT "K+";: VTAB 9: HTAB 2: PRINT
       "5.0": VTAB 11: HTAB 2: PRINT "4.0"
13670  Z0 = 10:Z2 =  - 70:Z3 = 10:LO = 1: GOSUB 13280
13700  HCOLOR= 2: HPLOT 30,110 TO 245,110: HPLOT 245,120 TO
       30,120
13710  VTAB 15: HTAB 38: PRINT "CL";: VTAB 14: HTAB 2: PRINT
       "105": VTAB 16: HTAB 3: PRINT "95"
13740  Z0 = 13:Z2 = 25:Z3 = 1:LO = 70: GOSUB 13280
13770  HCOLOR= 1: HPLOT 30,160 TO 245,160: HPLOT 245,170 TO
       30,170
13780  VTAB 21: HTAB 38: PRINT "GLU";: VTAB 21: HTAB 2: PRINT
       "8.0": VTAB 22: HTAB 2: PRINT "4.0
13810  Z0 = 29:Z1 = 4:Z2 =  - 10:Z3 = 2.5:LO = .5: GOSUB 13280
13840  VTAB 1: HTAB 1:Z$ = "R(ENAL  M(ENU  X(PAND": PRINT N$;:
       HTAB 40 -  LEN (Z$): PRINT Z$
13850  GOSUB 7: GET A$: IF A$ = "R" THEN 13870
```

```
13851 IF A$ = "C" THEN 20200
13852 IF A$ = "M" THEN 13861
13853 IF A$ = "X" THEN 13863
13854 IF A$ =  CHR$ (12) THEN  GOSUB 21000
13855 IF A$ =  CHR$ (16) THEN  GOSUB 30000
13856 IF A$ = "E" THEN  GOSUB 13505: GOTO 13610
13857 IF A$ = "T" THEN 15000
13858 IF A$ = "F" THEN  GOSUB 13505: GOTO 19400
13859 IF A$ = "P" THEN 19570
13860 GOTO 13850
13861 CALL 1002: PRINT D$;"PR#0": PRINT D$;"IN#0": TEXT : RETURN
13863 CALL 1002: PRINT D$;"PR#0": PRINT D$;"IN#0": TEXT : GOTO 13001
13870 GOSUB 13505
13878 REM  * RENAL *
13880 HCOLOR= 5: HPLOT 30,50 TO 30,60
13900 VTAB 8: HTAB 1: PRINT "MMOL/L"
13910 REM  UREA
13920 HCOLOR= 2: HPLOT 30,110 TO 245,110: HPLOT 245,120 TO 30,120
13930 VTAB 15: HTAB 37: PRINT "UREA": VTAB 14: HTAB 2: PRINT "8.0": VTAB 16: HTAB 2: PRINT "4.0"
13960 Z0 = 16:Z1 = 4:Z2 =  - 60:Z3 = 2.5:LO = 1: GOSUB 13280
13990 HCOLOR= 1: HPLOT 30,160 TO 245,160: HPLOT 245,170 TO 30,170
14000 VTAB 21: HTAB 37: PRINT "CREA": VTAB 22: HTAB 3: PRINT "40": VTAB 21: HTAB 2: PRINT "120"
14020 VTAB 21: HTAB 2: PRINT "120"
14030 Z0 = 20:Z1 = 3:Z2 =  - 15:Z3 = 1 / 8:LO = 10: GOSUB 13280
14070 HCOLOR= 7: HPLOT 30,40 TO 245,40: HPLOT 245,20 TO 30,20
14080 VTAB 4: HTAB 37: PRINT "U/P": VTAB 3: HTAB 2: PRINT "2.0": VTAB 6: HTAB 2: PRINT "1.0"
14110 Z0 = 33:Z1 = 4:Z2 =  - 130:Z3 = 20:LO = 1: GOSUB 13280
14130 VTAB 1: HTAB 1: PRINT "MMOL/KG ";N$;:Z$ = "E(LEC M(ENU X(PAND": HTAB 40 -  LEN (Z$): PRINT Z$
14140 GOTO 13850
```

```
14700  REM ******************
14705  REM * CARDIOVASCULAR *
14707  REM ******************
14710  JJ = 160: VTAB 2: PRINT JJ: VTAB 7: PRINT JJ - 40: VTAB
       12: HTAB 2: PRINT JJ - 80: VTAB 17: HTAB 2: PRINT JJ -
       120
14720  TS = 160:BS =  - 10:Y = 120: GOSUB 15037: HPLOT 30,Y TO
       245,Y:Y = 80: GOSUB 15037: HPLOT 30,Y TO 245,Y
14730  HCOLOR= 5: VTAB 6: HTAB 37: PRINT "BP": HPLOT 250,48 TO
       266,48 TO 266,49 TO 250,49
14750  ZO = 10:Z1 = 3: GOSUB 13280
14770  ZO = 13: GOSUB 13280: REM  DIA
14800  HCOLOR= 6
14805  VTAB 10: HTAB 37: PRINT "HR": HPLOT 250,80 TO 266,80 TO
       266,81 TO 250,81
14810  ZO = 7: GOSUB 13280
14850  HCOLOR= 1: REM  CVP
14855  TS = 60:BS =  - 5:Y = 0: GOSUB 15037: HPLOT 30,Y TO
       245,Y
14857  VTAB 14: HTAB 37: PRINT "CVP": HPLOT 252,116 TO 273,116
       TO 273,115 TO 252,115
14858  VTAB 19: HTAB 3: PRINT "+5": VTAB 22: HTAB 3: PRINT
       "-5"
14860  ZO = 16:Z1 = 3:LO =  - 20: GOSUB 13280
14870  HCOLOR= 2: VTAB 18: HTAB 37: PRINT "PAWP": HPLOT
        252,146 TO 278,146 TO 278,147 TO 252,147
14880  ZO = 19: GOSUB 13280
14890  VTAB 1: HTAB 1: PRINT N$;:Z$ = "T(EMPS  M(ENU  X(PAND":
       HTAB 40 -  LEN (Z$): PRINT Z$
14900  GOTO 13850
15000  REM ****************
15010  REM * TEMPERATURES *
15015  REM ****************
15020  GOSUB 13505
15030  VTAB 2: HTAB 3: PRINT 45: VTAB 6: HTAB 3: PRINT 40:
       VTAB 10: HTAB 3: PRINT 35: VTAB 18: HTAB 4: PRINT 5:
       VTAB 22: HTAB 4: PRINT 0
15035  HCOLOR= 5:TS = 45:BS = 20:Y = 37: GOSUB 15037: HPLOT
```

```
       30,Y TO 245,Y:Y = 40: GOSUB 15036:Y = 45: GOSUB 15036:Y
       = 35: GOSUB 15036: GOTO 15040
15036  HCOLOR= 3: GOSUB 15037: HPLOT 30,Y TO 35,Y TO 35,Y + 1
       TO 30,Y + 1: RETURN
15037  Y = 174 - (Y - BS) * 164 / (TS - BS): RETURN
15040  HCOLOR= 5:Z0 = 26:Z1 = 4:LO = 10: GOSUB 13280
15045  VTAB 6: HTAB 37: PRINT "CORE": HPLOT 252,48 TO 278,48
       TO 278,49 TO 252,49
15050  HCOLOR= 2:TS = 25:BS = 0:Y = 5: GOSUB 15037: HPLOT 30,Y
       TO 245,Y:Y = 0: GOSUB 15037: HPLOT 30,Y TO 245,Y
15055  VTAB 19: HTAB 37: PRINT "GRAD": HPLOT 252,154 TO
       278,154 TO 278,155 TO 252,155
15060  Z0 = 29:Z1 = 3:LO = 0: GOSUB 13280
15080  VTAB 1: HTAB 1: PRINT N$;:Z$ = "C(VS  M(ENU  X(PAND":
       HTAB 40 -  LEN (Z$): PRINT Z$
15090  GET A$: IF A$ = "C" THEN  GOSUB 13505: GOSUB 14700
15095  GOTO 13850
15600  REM *********
15610  REM * LFT'S *
15615  REM *********
15620  VTAB 4: HTAB 37: PRINT "ALB": VTAB 3: HTAB 3: PRINT
       "50": VTAB 6: HTAB 3: PRINT "30
15630  HCOLOR= 6: HPLOT 30,30 TO 245,30: HPLOT 30,20 TO 245,20
15636  Z0 = 7:Z1 = 3:Z2 =  - 120:Z3 = 1:LO = 10: GOSUB 13280
15638  HCOLOR= 7: HPLOT 30,40 TO 245,40: HPLOT 30,50 TO 245,50
15640  VTAB 7: HTAB 37: PRINT "GLOB"
15670  Z0 = 10: GOSUB 13280
15710  VTAB 12: HTAB 37: PRINT "BIL": VTAB 11: HTAB 3: PRINT
       "30": VTAB 14: HTAB 4: PRINT "0"
15720  HCOLOR= 5: HPLOT 30,90 TO 245,90: HPLOT 245,100 TO
       30,100
15740  Z0 = 13:Z1 = 3:Z2 =  - 90:Z3 = .35:LO = .1: GOSUB 13280
15770  VTAB 17: HTAB 2: PRINT "3.0": VTAB 19: HTAB 2: PRINT
       "2.0": VTAB 18: HTAB 37: PRINT "CALC";
15780  HCOLOR= 2: HPLOT 30,140 TO 245,140: HPLOT 30,150 TO
       245,150: HPLOT 30,130 TO 245,130
15810  Z0 = 27:Z1 = 3:Z2 = 0:Z3 = 20:LO = 1: GOSUB 13280
15820  VTAB 21: HTAB 2: PRINT "1.0": VTAB 21: HTAB 37: PRINT
```

```
      "PHOS";
15830 HCOLOR= 1: HPLOT 30,160 TO 245,160
15835 Z0 = 30:Z1 = 3:Z2 =  - 20:Z3 = 10:L0 = 0: GOSUB 13280
15840 VTAB 1: HTAB 1:Z$ = "E(NZYMES M(ENU X(PAND": PRINT N$;:
      HTAB 40 -  LEN (Z$): PRINT Z$
15850 GET A$: IF A$ = "E" THEN 15900
15860 GOTO 13852
15900 REM * ENZYMES *
15910 LG$ = "Y"
15920 GOSUB 13505
15930 VTAB 7: HTAB 37: PRINT "ALK": VTAB 8: HTAB 37: PRINT
      "PHOS";:: VTAB 2: HTAB 1: PRINT "1000": VTAB 9: HTAB 2:
      PRINT "100": VTAB 16: HTAB 3: PRINT "10": VTAB 22: HTAB 4
      : PRINT "0"
15931 GOTO 15935
15932 Y = 174 -  LOG (Y) /  LOG (10) * 164 / 3: RETURN
15935 HCOLOR= 3:Y = 10: GOSUB 15932: HPLOT 30,Y TO 245,Y:Y =
      50: GOSUB 15932: HPLOT 30,Y TO 245,Y: HCOLOR= 1:Y = 80:
      GOSUB 15932: HPLOT 30,Y TO 245,Y:Y = 280: GOSUB 15932:
      HPLOT 30,Y TO 245,Y
15950 VTAB 12: HTAB 37: PRINT "AST": HCOLOR= 2: HPLOT 255,97
      TO 271,97 TO 271,98 TO 255,98
15960 Z0 = 16:Z1 = 4:Z2 = 1:Z3 = 1:L0 = .1: GOSUB 13280
15965 VTAB 17: HTAB 37: PRINT "ALA": HCOLOR= 5: HPLOT 255,137
      TO 271,137 TO 271,138 TO 255,138
15970 Z0 = 20: GOSUB 13280
15980 HCOLOR= 1:Z0 = 24:Z1 = 3:L0 = 9: GOSUB 13280:LG$ = "":
      REM :ALK PHOS
16000 VTAB 1: HTAB 1:Z$ = "L(FT'S  M(ENU  X(PAND": PRINT N$;:
      HTAB 40 -  LEN (Z$): PRINT Z$
16010 GET A$: IF A$ = "L" THEN  GOSUB 13505: GOTO 15600
18020 GOTO 13852
19000 REM ***************
19005 REM * HAEMATOLOGY *
19007 REM ***************
19010 HCOLOR= 7: HPLOT 30,30 TO 245,30: HPLOT 245,40 TO 30,40
19020 VTAB 5: HTAB 37: PRINT "HB": VTAB 4: HTAB 3: PRINT
      "12": VTAB 6: HTAB 3: PRINT "11"
```

```
19050  Z0 = 7:Z1 = 4:Z2 =  - 40:Z3 = 10: GOSUB 13280
19080  HCOLOR= 5: HPLOT 30,70 TO 245,70: HPLOT 245,80 TO 30,80
19090  VTAB 10: HTAB 37: PRINT "PCV": VTAB 9: HTAB 3: PRINT
       "45": VTAB 11: HTAB 3: PRINT "35"
19120  Z0 = 11:Z1 = 2:Z2 =  - 75:Z3 = 1: GOSUB 13280
19150  HCOLOR= 2: HPLOT 30,110 TO 245,110: HPLOT 245,120 TO
       30,120
19160  VTAB 15: HTAB 37: PRINT "WBC": VTAB 14: HTAB 3: PRINT
       "10": VTAB 16: HTAB 4: PRINT "5"
19190  Z0 = 13:Z1 = 4:Z2 =  - 61:Z3 = 2: GOSUB 13280
19220  HCOLOR= 1: HPLOT 30,150 TO 245,150: HPLOT 245,170 TO
       30,170
19230  VTAB 21: HTAB 37: PRINT "PLTS": VTAB 19: HTAB 2: PRINT
       150: VTAB 22: HTAB 3: PRINT 50
19260  Z0 = 17:Z1 = 3:Z2 =  - 10:Z3 = .2:L0 = 2: GOSUB 13280
19280  VTAB 1: HTAB 1: PRINT N$;:Z$ = "M(ENU  X(PAND": HTAB 40
       - LEN (Z$): PRINT Z$: GOTO 13850
19400  REM ***************
19405  REM * BLOOD GASES *
19407  REM ***************
19410  HCOLOR= 5: HPLOT 30,160 TO 245,160: HPLOT 30,170 TO
       245,170
19420  VTAB 21: HTAB 37: PRINT "FIO2": VTAB 20: HTAB 3: PRINT
       "60": VTAB 22: HTAB 3: PRINT "40"
19450  Z0 = 7:Z1 = 3:Z2 =  - 0:Z3 = .5: GOSUB 13280
19470  JJ = 160: VTAB 2: PRINT JJ: VTAB 7: PRINT JJ - 40: VTAB
       12: HTAB 2: PRINT JJ - 80: VTAB 17: HTAB 2: PRINT JJ -
       120
19480  HCOLOR= 7: HPLOT 30,90 TO 242,90: HPLOT 30,130 TO
       242,130
19490  VTAB 10: HTAB 37: PRINT "PO2"
19495  HCOLOR= 1: HPLOT 255,85 TO 271,85: HPLOT 255,84 TO
       271,84
19500  Z0 = 10:Z1 = 4:Z2 =  - 20:Z3 = 1: GOSUB 13280
19525  VTAB 15: HTAB 37: PRINT "PCO2": HCOLOR= 6: HPLOT
       255,125 TO 275,125: HPLOT 255,124 TO 275,124
19530  Z0 = 14:Z1 = 4:Z2 =  - 20:Z3 = 1: GOSUB 13280
19540  VTAB 1: HTAB 1:Z$ = "P(H/PCO2/BE  M(ENU  X(PAND": PRINT
```

```
        N$;: HTAB 40 -  LEN (Z$): PRINT Z$
19550   GOTO 13850
19570   GOSUB 13505
19580   VTAB 3: HTAB 1: PRINT "7.45": VTAB 4: HTAB 37: PRINT
        "PH": VTAB 6: HTAB 1: PRINT "7.35"
19598   Z0 = 18:Z1 = 4:Z2 = 1393:Z3 = 210
19599   HCOLOR= 1: HPLOT 30,30 TO 240,30: GOSUB 13280
19600   VTAB 12: HTAB 2: PRINT "50": VTAB 13: HTAB 37: PRINT
        "PCO2": VTAB 14: HTAB 2: PRINT "30"
19620   HCOLOR= 6: HPLOT 30,100 TO 240,100
19630   Z0 = 14:Z1 = 4:Z2 =  - 50:Z3 = 1: GOSUB 13280
19780   VTAB 19: HTAB 3: PRINT "+5": VTAB 22: HTAB 3: PRINT
        "-5": VTAB 20: HTAB 37: PRINT "B.E."
19790   HCOLOR= 5: HPLOT 30,160 TO 242,160
19820   Z0 = 22:Z1 = 4:Z2 =  - 30:Z3 = 2:LO =  - 10: GOSUB
13280
19840   VTAB 1: HTAB 1:Z$ = "F(IO2/PO2  M(ENU  X(PAND": PRINT
        N$;: HTAB 40 -  LEN (Z$): PRINT Z$
19850   GOTO 13850
20000   REM ***************
20005   REM * RESPIRATORY *
20007   REM ***************
20010   VTAB 1: HTAB 1: PRINT N$;:Z$ = "C(OMP M(ENU X(PAND":
        HTAB 40 -  LEN (Z$): PRINT Z$
20020   HCOLOR= 3:TS = 1200:BS = 0:Y = 600: GOSUB 15037: HPLOT
        30,Y TO 245,Y: VTAB 2: PRINT 1200: VTAB 12: HTAB 2:
        PRINT 600
20025   VTAB 10: HTAB 37: PRINT "INVT": VTAB 5: HTAB 37: PRINT
        "EXVT"
20030   HCOLOR= 6: HPLOT 252,80 TO 278,80 TO 278,81 TO 252,81
20040   Z0 = 9:Z1 = 4:HI = 1500:LO = 0: GOSUB 13280
20050   HCOLOR= 2: HPLOT 252,40 TO 278,40 TO 278,41 TO 252,41
20060   Z0 = 13: GOSUB 13280
20065   VTAB 20: HTAB 37: PRINT "RATE"
20070   HCOLOR= 5: HPLOT 252,160 TO 278,160 TO 278,161 TO
        252,161
20080   TS = 100:BS = 0:Y = 10: GOSUB 15037: HPLOT 30,Y TO
        245,Y: VTAB 20: HTAB 3: PRINT 10
```

```
20090  Z0 = 7:Z1 = 2: GOSUB 13280
20100  HCOLOR= 1:TS = 100:BS = 0:Y = 20: GOSUB 15037: HPLOT
       30,Y TO 245,Y: VTAB 18: HTAB 3: PRINT 20
20110  Z0 = 20:ZI = 2: GOSUB 13280
20120  VTAB 17: HTAB 37: PRINT "PAUS":
20125  HCOLOR= 1: HPLOT 252,137 TO 278,137 TO 278,136 TO
       252,136
20140  Z0 = 22: GOSUB 13280
20170  GOTO 13850
20200  REM * COMP *
20205  GOSUB 13505
20210  VTAB 1: HTAB 1: PRINT N$;:Z$ = "V(OLS M(ENU X(PAND":
       HTAB 40 -  LEN (Z$): PRINT Z$
20220  VTAB 12: HTAB 37: PRINT "COMP": HCOLOR= 2: HPLOT 252,96
       TO 278,96 TO 278,97 TO 252,97
20225  TS = 100:BS =  - 10:Y = 40: GOSUB 15037: HPLOT 30,Y TO
       245,Y: VTAB 13: HTAB 3: PRINT 40
20230  Z0 = 26:Z1 = 2: GOSUB 13280
20240  VTAB 18: HTAB 37: PRINT "PEEP": HCOLOR= 6: HPLOT
       252,146 TO 278,146 TO 278,147 TO 252,147: VTAB 20: HTAB
       3: PRINT 10: VTAB 23: HTAB 4: PRINT 0
20250  TS = 80:BS = 0:Y = 10: GOSUB 15037: HPLOT 30,Y TO 245,Y
20260  Z0 = 24:Z1 = 2: GOSUB 13280
20270  VTAB 4: HTAB 37: PRINT "WORK": HCOLOR= 5: HPLOT 252,34
       TO 278,34 TO 278,35 TO 252,35: VTAB 5: HTAB 4: PRINT 2:
       VTAB 7: HTAB 4: PRINT 1
20280  TS = 4:BS =  - 8:Y = 2: GOSUB 15037: HPLOT 30,Y TO
       245,Y:Y = 1: GOSUB 15037: HPLOT 30,Y TO 245,Y
20290  Z0 = 28:Z1 = 5: GOSUB 13280
20300  GET A$: IF A$ = "V" THEN  GOSUB 13505: GOTO 20000
20310  GOTO 13851
21000  REM *********************
21005  REM * LABEL TREND GRAPH *
21007  REM *********************
21010  LBL$ = "Y": GOSUB 13280
21065  GOSUB 13506: IF LM = 0 THEN LB$ = "": RETURN
21070  FOR I = 1 TO LM:X2 = MA(I): GOSUB 13384: VTAB 24: HTAB
       X2 / 7 + 1: PRINT  CHR$ (96 + I);: NEXT
```

```
21080  A$ = "":LB$ = "": FOR I = 1 TO LM:MA(I) = 0: NEXT :LM =
       0: RETURN
25000  REM *****************
25005  REM * MIN->DAYS/HRS *
25010  REM *****************
25020  MT = 0
25040  DY =  INT (ET / 1440):ET = ET - DY * 1440
25060  H =  INT (ET / 60):M = ET - H * 60: RETURN
25070  VTAB 1: HTAB 1:Z$ = "M(ENU  X(PAND": PRINT N$;: HTAB 40
       -  LEN (Z$): PRINT Z$
25080  GOTO 13850
26000  REM **************
26005  REM * DAYS->DATE *
26007  REM **************
26010  IF L = 1 AND DY = 60 THEN MT = 2:DY = 29: GOTO 26070
26015  IF L = 1 AND DY > 60 THEN DY = DY - 1
26017  RESTORE : READ J
26020  FOR II = 1 TO 12: READ J:MT = MT + 1
26030  IF DY - J < 1 THEN ND = DY: GOTO 26070
26040  DY = DY - J: NEXT
26070  ND$ =  STR$ (ND): IF  LEN (ND$) < 2 THEN ND$ = " " +
       ND$
26072  MT$ =  STR$ (MT): IF  LEN (MT$) < 2 THEN MT$ = "0" +
       MT$
26080  SM$(0) = ND$ + "/" + MT$:H$ =  STR$ (H): IF  LEN (H$) <
       2 THEN H$ = " " + H$
26082  M$ =  STR$ (M): IF  LEN (M$) < 2 THEN M$ = "0" + M$
26085  SM$(1) = H$ + "." + M$:ND$ = "":MT$ = "":H$ = "":M$ =
       "": RETURN
26086  DATA 0,31,28,31,30,31,30,31,31,30,31,30,31
29980  REM ****************************
29990  REM * PRINT GRAPH ON SILENTYPE *
29995  REM ****************************
30000  VTAB 2: PRINT : PRINT D$;"PR#1": PRINT : PRINT " ":
       POKE  - 12528,7: POKE  - 12526,83: POKE  - 12525,64:
       POKE  - 12529,255: POKE  - 12524,0: PRINT  CHR$ (17):
       PRINT D$;"PR#0":A$ = "": RETURN
```

```
10  REM ***************
12  REM * PLOTADDNAME *
15  REM ***************
30  D$ =  CHR$ (4): GOTO 100
39  REM *********************
40  REM * ADD LEADING ZEROS *
41  REM *********************
42  IF  LEN (TT$) < TT THEN TT$ = "0" + TT$: GOTO 42
44  RETURN
49  REM *********************
50  REM * ADD TRAILING DOTS *
51  REM *********************
52  IF  LEN (TT$) < TT THEN TT$ = TT$ + ".": GOTO 52
54  RETURN
97  REM ********
98  REM * MENU *
99  REM ********
100  HOME :Z$ = "ADD / ALTER NAME FILE": PRINT Z$: FOR I = 1
     TO  LEN (Z$): PRINT "=";: NEXT
102  PRINT
105  VTAB 5: PRINT "1. ADD NAME TO FILE
110  VTAB 7: PRINT "2. ALTER / DELETE NAME ON FILE
120  VTAB 9: PRINT "3. RETURN TO MAIN PROGRAM
125  VTAB 15: PRINT "SELECT REQUIRED NUMBER  ";
130  GET A$: PRINT A$: IF A$ < "1" OR A$ > "3" THEN 100
140  IF A$ = "1" THEN 1200
150  IF A$ = "3" THEN  PRINT D$;"BLOAD CHAIN,A520": CALL
     520"FILEPLOTNAMESORT"
1000  REM **************
1001  REM * LIST NAMES *
1002  REM **************
1010  ONERR  GOTO 1014
1012  GOTO 1018
1014  HOME :Z$ = "NO NAMES ON LIST": VTAB 10: HTAB (40 -  LEN
      (Z$)) / 2: PRINT Z$: FOR I = 1 TO 2000: NEXT
1016  POKE 216,0: GOTO 100
1018  NA$ = "NAMELISTLEN": PRINT D$;"OPEN";NA$
1019  PRINT D$;"READ";NA$: INPUT NI
```

```
1020  PRINT D$;"CLOSE";NA$
1021  NA$ = "NAMELIST"
1022  PRINT D$;"OPEN";NA$
1023  PRINT D$;"READ";NA$
1024  FOR I = 1 TO NI: INPUT Z$(I): NEXT
1026  PRINT D$;"CLOSE";NA$
1030  POKE 216,0
1035  HOME : PRINT "  NAME";: HTAB 23: PRINT "HOSPITAL NO. ":
      PRINT "  ====";: HTAB 23: PRINT "===========
1038  PRINT
1040  FOR J = 1 TO NI: PRINT : PRINT J;". ";:NA$ =  MID$
      (Z$(J),2,20): PRINT NA$;: HTAB 25:NO$ =  MID$
      (Z$(J),22,6): PRINT NO$: NEXT
1050  VTAB 22: HTAB 1: PRINT "SELECT PATIENT NO. FOR
      CORRECTION"
1060  VTAB 24: HTAB 1: PRINT "TYPE 'M' TO RETURN MENU ";
1065  GET A$: IF A$ = "M" THEN 100
1070  IF A$ < "1" OR A$ >  STR$ (NI) THEN 1065
1080  GOTO 5000
1200  REM ************
1201  REM * ADD NAME *
1202  REM ************
1210  HOME : VTAB 1: HTAB 1: PRINT "TYPE 'SURNAME' <SPACE>
      'INITIAL'": PRINT : PRINT "THEN TYPE <RETURN>   ": VTAB
      5: INPUT NA$
1214  IF  LEN (NA$) > 20 THEN 1220
1215  FOR I = 1 TO 20:A$ =  MID$ (NA$,I,1): IF A$ < "0" OR A$
      > "9" THEN  NEXT : GOTO 1217
1216  GOTO 1220
1217  FOR I = 3 TO 20: IF  MID$ (NA$,I,1) = " " THEN 1225
1218  NEXT
1220  FOR I = 1 TO  LEN (NA$): PRINT " ";: PRINT : PRINT "ONLY
      20 CHARS INCLUDING 1 SPACE ALLOWED": FOR I = 1 TO 2000:
      NEXT : GOTO 1210
1225  TT$ = NA$:TT = 20: GOSUB 50:NA$ = TT$
1230  VTAB 10: HTAB 33: FOR I = 1 TO  LEN (NO$) + 1: PRINT "
      ";: NEXT : PRINT
1235  VTAB 10: PRINT "TYPE HOSPITAL NO. THEN <RETURN> ";:
```

```
     INPUT NO$
1240 IF  LEN (NO$) > 6 OR  LEN (NO$) < 3 THEN  VTAB 10: HTAB
     32: FOR I = 1 TO 1 +  LEN (NO$): PRINT " ";: NEXT :
     PRINT : VTAB 10: PRINT "MIN OF 3 & MAX OF 6 CHARACTERS
     ALLOWED": FOR I = 1 TO 3000: NEXT : GOTO 1230
1245 TT$ = NO$:TT = 6: GOSUB 40:NO$ = TT$
1250 VTAB 15: HTAB 33: FOR I = 1 TO  LEN (BW$) + 2: PRINT "
     ";: NEXT : PRINT
1255 VTAB 15: HTAB 1: PRINT "TYPE BODY WEIGHT THEN <RETURN>
     ";: INPUT BW$
1260 IF  LEN (BW$) > 3 OR  LEN (BW$) < 2 THEN  VTAB 15: HTAB
     33: FOR I = 1 TO 1 +  LEN (NO$): PRINT " ";: NEXT :
     PRINT : VTAB 15: PRINT "MIN OF 3 & MAX OF 6 CHARS ALLOWED"
     : FOR I = 1 TO 2000: NEXT : GOTO 1250
1265 TT$ = BW$:TT = 3: GOSUB 40:BW$ = TT$
1270 N$ = "": FOR I = 1 TO 20: IF  MID$ (NA$,I,1) <  > "."
     THEN N$ = N$ +  MID$ (NA$,I,1): NEXT
1280 Z$ = "NAMELISTLEN"
1285 X$ = ""
1287 ONERR  GOTO 1320
1290 PRINT D$;"OPEN";Z$
1300 PRINT D$;"READ";Z$
1305 INPUT NI
1310 PRINT D$;"CLOSE";Z$
1320 NI = NI + 1
1325 PRINT D$;"OPEN";Z$
1330 PRINT D$;"WRITE";Z$
1332 PRINT NI
1334 PRINT D$;"CLOSE";Z$
1336 POKE 216,0
1338 Z$ = "NAMELIST"
1340 PRINT D$;"OPEN";Z$
1345 PRINT D$;"POSITION";Z$;",R";NI-1
1350 PRINT D$;"WRITE";Z$
1355 PRINT "X" + NA$ + NO$: REM * 'X' USED IN INTER-APPLE
     COMMUNICATION *
1360 PRINT D$;"CLOSE";Z$
1370 GOTO 100
```

```
5000  VTAB 22: HTAB 1: FOR I = 1 TO 38: PRINT " ";: NEXT
5010  VTAB 24: HTAB 1: PRINT "TYPE 'D' TO D(ELETE PATIENT
      NUMBER "; VAL (A$);" ";: GET Z$: IF Z$ < > "D" THEN 100
5020  II = VAL (A$)
5030  FOR JJ = II TO NI:Z$(JJ) = Z$(JJ + 1): NEXT :Z$(NI) =
      "":NI = NI - 1
5040  VTAB 24: HTAB 1: PRINT "    WRITE CORRECTED FILE TO
      DISC? Y/N ";: GET A$: IF A$ = "N" THEN 100
5050  IF A$ < > "Y" THEN 5040
5055  PRINT D$
5060  Z$ = "NAMELISTLEN"
5070  PRINT D$;"OPEN";Z$: PRINT D$;"WRITE";Z$: PRINT NI: PRINT
      D$;"CLOSE"
5080  Z$ = "NAMELIST"
5085  PRINT D$;"DELETE";Z$
5090  PRINT D$;"OPEN";Z$: PRINT D$;"WRITE";Z$: FOR I = 1 TO
      NI: PRINT Z$(I): NEXT : PRINT D$;"CLOSE"
6000  GOTO 100
```

```
2  REM ****************
3  REM * PLOTNAMESORT *
4  REM ****************
10  D$ =  CHR$ (4)
84  HOME
85  VTAB 15: HTAB 10: PRINT "OBTAINING DATA FILE "
87  Z$ = "NAMELISTLEN"
90  PRINT D$;"OPEN";Z$
95  PRINT D$;"READ";Z$
100  INPUT NI
110  PRINT D$;"CLOSE";Z$
112  Z$ = "NAMELIST"
115  PRINT D$;"OPEN";Z$
120  PRINT D$;"READ";Z$
130  FOR I = 1 TO NI: INPUT Z$(I): NEXT
180  PRINT D$;"CLOSE";Z$
190  GOSUB 15000
270  HOME
280  VTAB 15: HTAB 10: PRINT "WRITING REARRANGED FILES"
290  PRINT D$;"OPEN";Z$
300  PRINT D$;"WRITE"Z$
310  FOR I = 1 TO NI: PRINT Z$(I): NEXT
360  PRINT D$;"CLOSE"
370  TEXT
375  X$ = ""
380  HOME :Z$ = "GLASGOW ROYAL INFIRMARY": GOSUB 400:Z$ =
     "RESPIRATORY INTENSIVE CARE UNIT": GOSUB 400
390  Z$ = "FILEPLOT PROGRAM": GOSUB 400: GOTO 410
400  PRINT : HTAB (40 -  LEN (Z$)) / 2: PRINT Z$: RETURN
410  PRINT D$;"BLOAD CHAIN,A520": CALL 520"FILEPLOT"
15000  REM ************
15001  REM * SORT SUB *
15002  REM ************
15010  HOME : VTAB 15: HTAB 15: PRINT "SORTING"
15050  N = NI
15060  REM ** SORT **
15070  F = 0:I = 0
15080  IF Z$(I) <  = Z$(I + 1) THEN 15130
```

```
15090  T$ = Z$(I + 1)
15100  Z$(I + 1) = Z$(I)
15110  Z$(I) = T$
15120  F = 1
15130  I = I + 1: IF I <  = N THEN 15080
15140  IF F = 1 THEN 15070
15150  FOR I = 2 TO NI + 1:Z$(I - 1) = Z$(I): NEXT
15160  RETURN
```

SECTION 4

ICU ANALYSIS SUITE

```
2  REM ***********
3  REM * ABGSTAT *
4  REM ***********
5  TEXT :D$ =  CHR$ (4): GOTO 10
7  HTAB (40 -  LEN (X$)) / 2: PRINT X$: RETURN
10  PRINT D$;"  MONI,C,O"
15  DIM Z$(500),X(500,1)
20  HOME :X$ = "GLASGOW ROYAL INFIRMARY": GOSUB 7: PRINT :X$ =
    "RESPIRATORY INTENSIVE CARE UNIT": GOSUB 7: PRINT :X$ =
    "ON-LINE 30  VTAB 20: PRINT "ENTER PATIENT NAME ";: INPUT
    NA$
30  VTAB 20: PRINT "ENTER PATIENT NAME ";: INPUT NA$
31  REM ***************************************************
32  REM * OBTAIN PATIENT DATA STRINGS FROM ALL CVS FILES *
33  REM ***************************************************
34  Y = Y + 50
35  X$ = NA$ + " CVS " +  STR$ (Y)
36  ONERR  GOTO 42
38  PRINT D$;"OPEN";X$: PRINT D$;"READ";X$: FOR I = 1 TO 50:
    INPUT Z$(I + Y - 50): NEXT
40  PRINT D$;"CLOSE": GOTO 32
42  Y = Y - 50: POKE 216,0: PRINT D$;"DELETE";X$: ONERR  GOTO
    55
48  X$ = NA$ + " CVS DATALEN"
50  PRINT D$;"OPEN";X$: PRINT D$;"READ";X$: INPUT FI: PRINT
    D$;"CLOSE": GOTO 60
55  HOME : POKE 216,0: HOME : VTAB 10: PRINT "NO FILE
    AVAILABLE ": FOR I = 1 TO 2000: NEXT : GOTO 30
60  X$ = NA$ + " CVS DATA"
65  ONERR  GOTO 85
70  PRINT D$;"OPEN";X$: PRINT D$;"READ";X$
80  FOR I = 1 TO FI: INPUT Z$(I + Y): NEXT
85  FI = I - 1 + Y: POKE 216,0
90  PRINT D$;"CLOSE"
100  REM ************************************
110  REM * SELECT VARIABLES FOR CORRELATION *
120  REM ************************************
200  HOME
```

```
205  K = 0
210  VTAB 5: PRINT "1. CVP": PRINT : PRINT "2. PAWP": PRINT :
     PRINT "3. C.O.": PRINT : PRINT "4. SATURATION"
220  VTAB 20: PRINT "TYPE TWO NUMBERS FOR CORRELATION ": PRINT
     : GET A$: PRINT A$,: GET B$: PRINT B$
225  IF A$ < "1" OR A$ > "4" THEN 200
226  IF B$ < "1" OR B$ > "4" THEN 200
230  A =  VAL (A$): GOSUB 250:S1 = START:L1 = NO:N1$ = N$
240  A =  VAL (B$): GOSUB 250:S2 = START:L2 = NO:N2$ = N$:
     GOTO 300
245  REM ************************************
244  REM * START = START POSITION IN STRING  *
246  REM * NO = NO OF CHARACTERS IN VARIABLE *
247  REM ************************************
250  IF A = 1 THEN START = 16:NO = 3:N$ = "CVP"
260  IF A = 2 THEN START = 19:NO = 3:N$ = "PAWP"
270  IF A = 3 THEN START = 32:NO = 4:N$ = "CO"
280  IF A = 4 THEN START = 25:NO = 4:N$ = "VSATN"
290  RETURN
294  REM **************************************************
295  REM * LOAD VARIABLES FOR CORRELATION INTO X(A,B) ARRAY *
296  REM **************************************************
300  FOR I = 1 TO FI:X(I - K,0) =  VAL ( MID$
     (Z$(I),S1,L1)):X(I - K,1) =  VAL ( MID$ (Z$(I),S2,L2))
303  IF N1$ = "VSATN" OR N2$ = "VSATN" AND  MID$ (Z$(I),29,3)
      < > "ATN" THEN K = K + 1: GOTO 315
305  IF X(I - K,0) =  - 99 OR X(I - K,1) =  - 99 THEN K = K +
     1: GOTO 315: REM PAWP/CVP
306  IF S1 = 16 OR S1 = 19 THEN 312
307  IF X(I - K,0) = 0 THEN K = K + 1: GOTO 315
312  IF S2 = 16 OR S2 = 19 THEN 315
313  IF X(I - K,1) = 0 THEN K = K + 1
315  NEXT
320  RN = FI - K: IF RN = 1 THEN  PRINT "RN = ";1: FOR I = 1
     TO 2000: NEXT : GOTO 200
322  TN = 1: GOSUB 3000: REM * EDITING SUBROUTINE *
326  GOSUB 1000: REM * CORRELATION CO-EFICIENT & REGRESSION
     EQUATION *
```

```
328  HOME
330  VTAB 5: PRINT "CORRELATION COEFF = ";RV: PRINT : PRINT
      RN;" RECORDS"
350  GOSUB 2000: VTAB 10: PRINT "REGRESSION EQUATION IS Y =
     ";C1;" X + ";C2
354  REM ***************************
355  REM * SAVE DATA ON STATS DISC *
356  REM ***************************
358  X$ = NA$ + " " + N1$ + "/" + N2$
360  VTAB 22:K = 0: PRINT "SAVE DATA ON 'STATS DISC' ? Y/N ";:
     GET A$: PRINT A$: IF A$ = "N" THEN 400
362  IF A$ < > "Y" THEN 360
368  HOME : VTAB 10: PRINT "PUT STATISICS DISC INTO THE
     DRIVE": PRINT : PRINT "AND TYPE <SPACEBAR> ";: GET A$:
     PRINT A$: IF A$ < > " " THEN 368
370  PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$
380  PRINT 1: PRINT RN: PRINT 500: PRINT 0: PRINT 0
390  FOR I = 1 TO RN: PRINT X(I,0): PRINT X(I,1): NEXT
395  PRINT D$;"CLOSE"
399  REM ***************************************
400  REM * SAVE DATA FOR APPLEPLOT AND HP PLOT *
401  REM ***************************************
405  HOME : VTAB 5: PRINT "SAVE DATA FOR APPLEPLOT/HP PLOT ?
     Y/N ";: GET A$: PRINT A$
406  IF A$ = "Y" THEN 417
410  IF A$ = "N" THEN  HOME : VTAB 10: PRINT "ANALYSE MORE
     DATA FROM THIS PATIENT? ";: GET A$: IF A$ = "N" THEN 20
412  IF A$ = "Y" THEN 200
415  GOTO 405
417  VTAB 10: PRINT "PUT APPLEPLOT DISC INTO THE DRIVE": PRINT
     : PRINT "AND TYPE <SPACEBAR> ";: GET A$: PRINT A$: IF A$
     < > " " THEN 417
420  PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$
430  PRINT RN: PRINT 2
435  LX = 99:HX = 0
440  FOR I = 1 TO RN: PRINT X(I,0): PRINT X(I,1): IF X(I,0) <
     LX THEN LX = X(I,0)
450  IF X(I,0) > HX THEN HX = X(I,0)
```

```
460  NEXT
470  LY = C1 * LX + C2:HY = C1 * HX + C2
480  PRINT LX: PRINT LY
490  PRINT HX: PRINT HY
492  FOR I = 1 TO 10: PRINT 1: NEXT
494  PRINT N1$: PRINT N2$:X$ = N2$ + "/" + N1$ + "  " + NA$ +
     "  R=" +  STR$ (RV): PRINT X$
500  PRINT D$;"CLOSE"
505  HOME
510  VTAB 15: HTAB 1: CALL  - 868: PRINT "DATA SAVED ON
      APPLEPLOT DISC": PRINT : PRINT "1. ANALYSE MORE DATA"
520  PRINT "2. RUN APPLEPLOT"
530  VTAB 22: GET A$:A =  VAL (A$): IF A = 2 THEN  PRINT :
     PRINT D$;"RUN HELLO"
535  IF A <  > 1 THEN 510
540  HOME : VTAB 10: PRINT "PUT DATA DISC INTO DRIVE"
545  PRINT : PRINT "AND PRESS <SPACEBAR> ";: GET A$: IF A$ <
     > " " THEN  HOME : GOTO 510
550  A$ = "N": GOTO 410
1000  REM ****************************************
1001  REM * CORRELATION CO-EFFICIENT CALCULATIONS *
1002  REM ****************************************
1005  IF RN < 2 THEN  RETURN
1010  XS = 0:XQ = 0:XY = 0:YS = 0:YQ = 0
1020  FOR I = 1 TO RN
1030  XS = XS + X(I,0)
1040  YS = YS + X(I,1)
1050  XY = XY + X(I,0) * X(I,1)
1060  XQ = XQ + X(I,0) * X(I,0)
1070  YQ = YQ + X(I,1) * X(I,1)
1080  NEXT I
1090  XM = XS / RN:YM = YS / RN
1100  RV = (XY / RN - XM * YM) /  SQR ((XQ / RN - XM * XM) *
      (YQ / RN - YM * YM))
1110  RV =  INT (RV * 1E2 + .5) / 1E2
1140  RETURN : REM * RV=COR COEF *
```

```
1997  REM ************************************
1998  REM * REGRESSION EQUATION CALCULATIONS *
1999  REM ************************************
2000  XS = 0:XQ = 0:XY = 0:YS = 0
2005  IF RN = 0 THEN  RETURN
2010  FOR I = 1 TO RN
2020  XS = XS + X(I,0)
2030  YS = YS + X(I,1)
2040  XY = XY + X(I,0) * X(I,1)
2050  XQ = XQ + X(I,0) * X(I,0)
2060  NEXT I
2070  XM = XS / RN:YM = YS / RN
2080  C1 = (XY - RN * XM * YM) / (XQ - RN * XM * XM)
2090  C2 = YM - C1 * XM
2100  C1 =  INT (C1 * 1E2 + .5) / 1E2
2110  C2 =  INT (C2 * 1E2 + .5) / 1E2
2170  RETURN : REM * MX=C1 C=C2 (Y=MX+C) *
3000  REM *******************************************************
3005  REM * EDITOR MENU -MODIFIED FROM STRATHAND STATS PACKAGE *
3000  REM *******************************************************
3010  HOME :T = 2
3020  PRINT : PRINT  TAB( T);"EDITOR MENU"
3030  PRINT  TAB( T);"------ ----"
3040  PRINT : PRINT  TAB( T + E);"1) LIST DATA"
3050  PRINT : PRINT  TAB( T + E);"2) APPEND DATA"
3060  PRINT : PRINT  TAB( T + E);"3) INSERT DATA"
3070  PRINT : PRINT  TAB( T + E);"4) DELETE DATA"
3080  PRINT : PRINT  TAB( T + E);"5) ALTER DATA"
3090  PRINT : PRINT  TAB( T + E);"6) RETURN"
3120  PRINT : PRINT : PRINT  TAB( T);"PLEASE SELECT - ";
3130  GET A$: IF A$ < "1" OR A$ > "6" THEN 3000
3140  A =  VAL (A$)
3150  ON A GOTO 3550,3160,4060,4490,4740,5000
3160  REM  ************** APPEND DATA **********
3170  HOME
3180  PRINT : PRINT "KEY <RETURN> AFTER EACH POINT ENTERED"
3190  PRINT : PRINT  TAB( T);"KEY 'END' <RETURN> WHEN ALL":
      PRINT  TAB( T);"POINTS ARE ENTERED"
```

```
3200  IF TN < > 2 GOTO 3230
3210  PRINT : PRINT "APPEND X GROUP DATA? Y OR N ..";: GOSUB
      4500: IF A$ = "N" GOTO 3450
3220  RN = RX
3230  VTAB 7: PRINT  TAB( 38): HTAB 1: PRINT "POINT NUMBER";
      TAB( 15);"X VALUE"; TAB( 26);
3240  IF TN < > 1 THEN  PRINT : PRINT : GOTO 3260
3250  PRINT "Y VALUE"
3260  POKE 34,8: HOME
3270  FOR I = RN + 1 TO MN
3280  PRINT  TAB( 2);I; TAB( 17): INPUT "";A$
3290  IF A$ < > "END" GOTO 3330
3300  RN = I - 1
3310  IF TN < > 2 THEN  POKE 34,0: GOTO 3000
3320  RX = RN: GOTO 3450
3330  X(I,0) =  VAL (A$)
3340  IF TN < > 1 THEN 3410
3350  CK = 0
3360  CV =  PEEK (37): VTAB CV
3370  HTAB 30: INPUT "";A$
3380  IF A$ = "END" THEN  PRINT G$;G$; TAB( H);"INPUT Y
      VALUE":CK =  PEEK (37): VTAB (CV): GOTO 3370
3390  X(I,1) =  VAL (A$)
3400  IF CK < > 0 THEN  VTAB CK: PRINT  TAB( 38): HTAB 1
3410  NEXT
3420  PRINT : PRINT  TAB( T);"MAXIMUM NUMBER OF ENTRIES
      REACHED"
3430  FOR J = 0 TO 2000: NEXT J
3440  RN = I - 1: GOTO 3000
3450  VTAB 7: PRINT "APPEND Y GROUP DATA? Y OR N ..";: GOSUB
4500: IF A$ = "N" THEN  POKE 34,0: GOTO 3000
3460  VTAB 7: PRINT  TAB( 38): HTAB 1: PRINT "POINT NUMBER";
      TAB( 15);"Y VALUE": POKE 34,8: HOME
3470  FOR I = RY + 1 TO MN
3480  PRINT  TAB( 2);I; TAB( 17): INPUT "";A$
3490  IF A$ = "END" THEN RY = I - 1: POKE 34,0: GOTO 3000
3500  X(I,1) =  VAL (A$)
3510  NEXT
```

```
3520 PRINT : PRINT  TAB( T);"MAXIMUM NUMBER OF ENTRIES
     REACHED"
3530 FOR J = 0 TO 2000: NEXT J
3540 RY = I - 1: GOTO 3000
3550 REM  ******* LIST DATA *********
3560 HOME
3570 PRINT : PRINT  TAB( T);"LIST TO SCREEN OR PRINTER (S/P)
     - ";: GET A$
3580 IF A$ < > "P" THEN PS = 0: GOTO 3630
3590 PRINT : PRINT "TURN ON PRINTER, ALIGN PAPER"
3600 PRINT "PRESS SPACE BAR TO CONTINUE"
3610 POKE  - 16368,0: WAIT  - 16384,128: WAIT  - 16384,1,1
3620 PS = 1
3630 HOME
3640 PRINT : PRINT "PRESS <SPACE BAR> TO STOP AND START"
3650 PRINT "LISTING,"
3660 PRINT  TAB( T);"OR <RETURN> TO RETURN TO EDIT MENU"
3670 POKE 34,6
3680 PRINT D$;"PR#";PS
3690 PRINT : PRINT "POINT NUMBER  X VALUE";
3700 IF TN = 2 THEN RN = RX: GOTO 3730
3710 IF TN = 0 THEN  GOTO 3730
3720 PRINT "    Y VALUE";
3730 PRINT
3740 FOR I = 1 TO RN
3750 PRINT  TAB( 2);I; TAB( 17);X(I,0);
3760 IF TN = 2 THEN 3790
3770 IF TN = 0 THEN  GOTO 3790
3780 HTAB 30: PRINT X(I,1);
3790 PRINT :X =  PEEK ( - 16384): IF X < 128 THEN 3840
3800 POKE  - 16368,0: WAIT  - 16384,128: WAIT  - 16384,1,1
3810 POKE  - 16368,0
3820 GET A$: IF A$ = R$ THEN  GOTO 4040
3830 GOTO 3850
3840 FOR J = 0 TO 100: NEXT J
3850 NEXT I
3860 IF TN < > 2 THEN  GOTO 4010
3870 PRINT D$;"PR#0"
```

```
3880  PRINT : PRINT  TAB( T);"PRESS <SPACE> TO CONTINUE": POKE
       - 16368,0: WAIT  - 16384,128: WAIT  - 16384,1,1
3890  POKE  - 16368,0
3900  PRINT D$;"PR#";PS:RN = RY: VTAB 5
3910  PRINT : PRINT "POINT NUMBER  Y VALUE"
3920  HOME
3930  FOR I = 1 TO RN
3940  PRINT  TAB( 2);I; TAB( 17);X(I,1)
3950  X =  PEEK ( - 16384): IF X < 128 THEN 3990
3960  POKE  - 16368,0: WAIT  - 16384,128: WAIT  - 16384,1,1
3970  POKE  - 16368,0
3980  GET A$: IF A$ = R$ THEN  GOTO 4040
3990  FOR J = 0 TO 100: NEXT J
4000  NEXT I
4010  PRINT D$;"PR#0"
4020  PRINT : PRINT : PRINT  TAB( T);"PRESS ANY KEY TO RETURN
      TO EDIT MENU"
4030  GET A$
4040  POKE 34,0
4050  GOTO 3000
4060  REM  ******** INSERT DATA ********
4070  HOME
4080  IF TN = 2 THEN RN = RX + RY
4090  IF RN < MN THEN 4120
4100  PRINT : PRINT G$; TAB( T);"NO SPACE FOR INSERTIONS"
4110  FOR T = 1 TO 2000: NEXT T: GOTO 3000
4120  IF TN = 2 THEN  GOTO 4250
4130  PRINT : PRINT  TAB( T);: INPUT "INSERT BEFORE WHICH
      NUMBER - ";NI
4140  IF NI <  = RN THEN 4170
4150  PRINT : PRINT  TAB( T);"THIS NUMBER IS TOO LARGE": FOR T
      = 1 TO 2000: NEXT T
4160  GOTO 4130
4170  FOR I = RN TO NI STEP  - 1
4180  X(I + 1,0) = X(I,0)
4190  IF TN = 1 THEN X(I + 1,1) = X(I,1)
4200  NEXT I
4210  RN = RN + 1
```

```
4220  PRINT : PRINT  TAB( T);: INPUT "NEW X VALUE - ";X(NI,0)
4230  IF TN = 1 THEN  PRINT  TAB( T);: INPUT "NEW Y VALUE -
      ";X(NI,1)
4240  GOTO 3000
4250  PRINT : PRINT  TAB( T);"INSERT IN X GROUP ? Y OR N ..";:
      GET A$: IF A$ <  > "Y" GOTO 4350
4260  PRINT : PRINT  TAB( T);: INPUT "INSERT BEFORE WHICH
      NUMBER ";NI
4270  IF NI <  = RX THEN 4300
4280  PRINT : PRINT  TAB( T);"THIS NUMBER IS TOO LARGE": FOR T
      = 1 TO 2000: NEXT
4290  GOTO 4260
4300  FOR I = RX TO NI STEP  - 1
4310  X(I + 1,0) = X(I,0)
4320  NEXT
4330  RX = RX + 1
4340  PRINT : PRINT  TAB( T);: INPUT "NEW X VALUE ";X(NI,0)
4350  PRINT : PRINT  TAB( T);"INSERT IN Y GROUP ? Y OR N ..";:
      GET A$: IF A$ <  > "Y" GOTO 4450
4360  PRINT : PRINT  TAB( T);: INPUT "INSERT BEFORE WHICH
      NUMBER ";NI
4370  IF NI <  = RY THEN  GOTO 4400
4380  PRINT : PRINT  TAB( T);"THIS NUMBER IS TOO LARGE": FOR T
      = 1 TO 2000: NEXT
4390  GOTO 4350
4400  FOR I = RY TO NI STEP  - 1
4410  X(I + 1,1) = X(I,1)
4420  NEXT
4430  RY = RY + 1
4440  PRINT : PRINT  TAB( T);: INPUT "NEW Y VALUE ";X(NI,1)
4450  GOTO 3000
4460  HOME
4470  PRINT : PRINT  TAB( T);G$;G$;"DATA LIST EMPTY": PRINT :
      PRINT  TAB( T);"CHECK FILES - RE-ENTER DATA"
4480  D = 15: GOSUB 4600: GOTO 65
4490  REM  ****** DELETE DATA ********
4500  HOME :MT = 0
4510  IF TN = 2 GOTO 4630
```

```
4520  PRINT : PRINT  TAB( T);: INPUT "DELETE WHICH POINT-";ND
4530  IF ND > RN THEN  PRINT "POINT NO. TOO LARGE": GOTO 4520
4540  FOR I = ND TO RN - 1
4550  IF MT = 0 GOTO 4570
4560  IF TN = 2 THEN X(I,1) = X(I + 1,1): GOTO 4590
4570  X(I,0) = X(I + 1,0)
4580  IF TN = 1 THEN X(I,1) = X(I + 1,1)
4590  NEXT
4600  IF TN = 2 GOTO 4660
4610  RN = RN - 1: IF RN = 0 GOTO 4460
4620  GOTO 3000
4630  PRINT : PRINT "DELETE POINT FROM X GROUP? Y OR N ..";:
      GET A$
4640  IF A$ = "Y" THEN RN = RX: GOTO 4520
4650  GOTO 4700
4660  IF MT = 0 GOTO 4690
4670  RY = RN - 1: IF RY = 0 GOTO 4460
4680  GOTO 3000
4690  RX = RN - 1: IF RX = 0 THEN  GOTO 4460
4700  RN = RY:MT = 1
4710  PRINT : PRINT "DELETE POINT FROM Y GROUP? Y OR N ..";:
      GET A$
4720  IF A$ < > "Y" GOTO 3000
4730  GOTO 4520
4740  REM  ******* EDIT DATA *******
4750  HOME :MT = 0
4760  IF TN = 2 GOTO 4880
4770  PRINT : PRINT  TAB( T);: INPUT "ENTER POINT NUMBER TO
      EDIT - ";NE
4780  IF (NE < 1) OR (NE > RN) THEN  PRINT "POINT NO. OUT OF
      RANGE": GOTO 4740
4790  PRINT : PRINT  TAB( T);"PRESENT VALUES"
4800  IF MT = 1 THEN  PRINT : PRINT "Y ";X(NE,1): PRINT :
      INPUT "NEW Y VALUE ";X(NE,1): GOTO 4860
4810  PRINT : PRINT "X ";X(NE,0);
4820  IF TN < > 1 THEN  PRINT : GOTO 4840
4830  PRINT  TAB( 20);"Y ";X(NE,1)
4840  PRINT : INPUT "NEW X VALUE ";X(NE,0)
```

```
4850  IF TN = 1 THEN  PRINT : PRINT : INPUT "NEW Y VALUE
      ";X(NE,1)
4860  IF TN = 2 GOTO 4900
4870  GOTO 3000
4880  PRINT : PRINT "ALTER GROUP X DATA? Y OR N ..";: GET A$
4890  IF A$ = "Y" THEN RN = RX: GOTO 4770
4900  IF MT = 1 GOTO 3000
4910  PRINT : PRINT ;"ALTER GROUP Y DATA? Y OR N ..";: GET A$
4920  IF A$ = "N" GOTO 3000
4930  RN = RY:MT = 1
4940  GOTO 4770
5000  RETURN
```

```
2    REM *************
3    REM * STAT>PLOT *
4    REM *************
10   TEXT
15   DIM Z$(500),X(500,1)
20   D$ =  CHR$ (4)
25   PRINT D$;"  MONI,C,O"
199  REM ***************************
200  REM * GET DATA FROM STAT FILE *
201  REM ***************************
210  HOME : VTAB 10: PRINT "ENTER FILE NAME ";: INPUT NA$
220  REM ONERR GOTO
230  PRINT D$;"OPEN";NA$: PRINT D$;"READ";NA$
240  INPUT TN,RN,MN,RX,RY
250  FOR I = 1 TO RN: INPUT X(I,0),X(I,1)
270  NEXT
280  PRINT D$;"CLOSE"
290  GOSUB 2000: REM * REGRESSION EQUATION *
300  GOSUB 1000: REM * CORRELATION CO-EFFICIENT *
350  HOME : VTAB 5: INPUT "ENTER X-AXIS ";N1$
360  VTAB 8: INPUT "ENTER Y-AXIS ";N2$
399  REM ***************************************
400  REM * SAVE DATA FOR APPLEPLOT AND HP PLOT *
401  REM ***************************************
404  HOME : VTAB 10: PRINT "PUT APPLEPLOT DISC IN DRIVE":
     PRINT : PRINT "AND TYPE <SPACEBAR> ";: GET A$
406  PRINT A$: IF A$ <  > " " THEN 404
410  X$ = NA$ + " " + N1$ + "/" + N2$
420  PRINT D$;"OPEN";X$: PRINT D$;"WRITE";X$
430  PRINT RN: PRINT 2
435  LX = 99:HX = 0
440  FOR I = 1 TO RN: PRINT X(I,0): PRINT X(I,1): IF X(I,0) <
     LX THEN LX = X(I,0): REM * OBTAIN LOWEST X VALUE *
450  IF X(I,0) > HX THEN HX = X(I,0): REM * OBTAIN HIGHEST X
     VALUE *
460  NEXT
470  LY = C1 * LX + C2:HY = C1 * HX + C2: REM CALCULATE LOWEST
     AND HIGHEST VALUE *
```

```
480  PRINT LX: PRINT LY
490  PRINT HX: PRINT HY
492  FOR I = 1 TO 10: PRINT 1: NEXT
494  PRINT N1$: PRINT N2$:X$ = N2$ + "/" + N1$ + "   " + NA$ +
     "  COR=" +  STR$ (RV): PRINT X$
500  PRINT D$;"CLOSE"
505  HOME
510  VTAB 15: HTAB 1: CALL  - 868: PRINT "DATA SAVED ON
     APPLEPLOT DISC": PRINT : PRINT "1. CONVERT MORE DATA"
520  PRINT "2. RUN APPLEPLOT"
530  VTAB 22: GET A$:A =  VAL (A$): IF A = 2 THEN  PRINT :
     PRINT D$;"RUN HELLO"
535  IF A <  > 1 THEN 510
540  HOME : VTAB 10: PRINT "PUT STATS DISC INTO DRIVE"
545  PRINT : PRINT "AND PRESS <SPACEBAR> ";: GET A$: IF A$ <
     > " " THEN  HOME : GOTO 510
550  GOTO 210
1000  REM ****************************************
1001  REM * CORRELATION CO-EFFICIENT CALCULATION *
1002  REM ****************************************
1005  IF RN = 0 THEN  RETURN
1010  XS = 0:XQ = 0:XY = 0:YS = 0:YQ = 0
1020  FOR I = 1 TO RN
1030  XS = XS + X(I,0)
1040  YS = YS + X(I,1)
1050  XY = XY + X(I,0) * X(I,1)
1060  XQ = XQ + X(I,0) * X(I,0)
1070  YQ = YQ + X(I,1) * X(I,1)
1080  NEXT I
1090  XM = XS / RN:YM = YS / RN
1100  RV = (XY / RN - XM * YM) /  SQR ((XQ / RN - XM * XM) *
      (YQ / RN - YM * YM))
1110  RV =  INT (RV * 1E2 + .5) / 1E2
1140  RETURN : REM * RV=COR COEF *
```

```
1997 REM ***********************************
1998 REM * REGRESSION EQUATION CALCUALTION *
1999 REM ***********************************
2000 XS = 0:XQ = 0:XY = 0:YS = 0
2005 IF RN = 0 THEN  RETURN
2010 FOR I = 1 TO RN
2020 XS = XS + X(I,0)
2030 YS = YS + X(I,1)
2040 XY = XY + X(I,0) * X(I,1)
2050 XQ = XQ + X(I,0) * X(I,0)
2060 NEXT I
2070 XM = XS / RN:YM = YS / RN
2080 C1 = (XY - RN * XM * YM) / (XQ - RN * XM * XM)
2090 C2 = YM - C1 * XM
2100 C1 =  INT (C1 * 1E2 + .5) / 1E2
2110 C2 =  INT (C2 * 1E2 + .5) / 1E2
2170 RETURN : REM * MX=C1 C=C2 (Y=MX+C) *
```

```
4   REM ***********
5   REM * HP PLOT *
6   REM ***********
10  TEXT : HOME
15  DIM D(400,1),X(400),Y(400)
20  D$ =  CHR$ (4)
27  REM **************************
28  REM * COLLECT DATA FROM DISC *
29  REM **************************
30  INPUT "FILE NAME ";X$
40  PRINT D$;"  MONI,C,O"
100  PRINT D$;"OPEN";X$: PRINT D$;"READ";X$
110  INPUT RN: INPUT R
120  FOR I = 1 TO RN: INPUT D(I,0): INPUT D(I,1): REM  *
     OBTAIN X,Y VALUES *
130  NEXT
135  IF X$ = "MCCORMICK SVR" THEN  FOR I = 1 TO RN:D(I,1) =
     D(I,1) * 80: NEXT
140  INPUT R(0,0): INPUT R(0,1): REM * INPUT VALUES FOR
     REGRESSION LINE *
150  INPUT R(1,0): INPUT R(1,1)
160  FOR I = 1 TO 10: INPUT S(I): NEXT : REM * INPUT APPLEPLOT
     SCALING VALUES *
165  L0 = S(8):L1 = S(7):NX = S(6)
166  H0 = S(10):H1 = S(9):NY = S(5)
170  INPUT N1$: INPUT N2$: INPUT GL$: REM * INPUT GRAPH LABELS *
175  PRINT D$;"CLOSE"
179  REM *****************************************
180  REM * OBTAIN LOWEST (X,Y) AND HIGHEST (X,Y) *
181  REM *****************************************
183  FOR I = 1 TO RN: IF D(I,0) < L0 THEN L0 = D(I,0)
184  IF D(I,0) > H0 THEN H0 = D(I,0)
185  IF D(I,1) < L1 THEN L1 = D(I,1)
186  IF D(I,1) > H1 THEN H1 = D(I,1)
187  NEXT
```

```
189  REM **************************
190  REM * SELECT PRINTER OPTIONS *
195  REM **************************
200  HOME : PRINT "LOWEST X = ";LO: PRINT "HIGHEST X = ";HO
210  VTAB 6: PRINT "LOWEST Y = ";L1: PRINT "HIGHEST Y = ";H1
220  VTAB 10
230  I = RN
234  REM *************************************
235  REM * ALLOCATE VALUES TO X AND Y ARRAYS *
236  REM *************************************
240  FOR II = 1 TO I:X(II) = D(II,0):Y(II) = D(II,1): NEXT
390  INPUT "NO OF X DIV ";NX
400  INPUT "NO OF Y DIV ";NY
410  VTAB 15
420  INPUT " MAX X ";HX
430  INPUT " MIN X ";LX
440  PRINT : INPUT " MAX Y ";HY
445  HY$ =  STR$ (HY)
450  INPUT " MIN Y ";LY
459  REM ***************************************************
460  REM * ENTER FACTORS TO COMPRESS GRAPH AND MOVE ORIGIN *
461  REM ***************************************************
480  HOME : PRINT "COMPRESSION FACTOR FOR X-AXIS ";: INPUT CX
490  PRINT "COMPRESSION FACTOR FOR Y-AXIS ";: INPUT CY
500  VTAB 10: PRINT "X-OFFSET ";: INPUT OX
510  VTAB 12: PRINT "Y-OFFSET ";: INPUT OY
550  PRINT : PRINT : PRINT : INPUT "PEN NO (1-2) ";PEN
560  PRINT : PRINT : PRINT : PRINT "JOIN POINTS? Y/N ";: GET
     JP$: PRINT
570  PRINT : PRINT : PRINT "PLOT REGRESSION LINE? Y/N ";: GET
     RE$: PRINT
580  PRINT : PRINT : PRINT "ENTER PEN SPEED (1-38) ";: INPUT SP
1000  REM ******************
1001  REM * SET UP PLOTTER *
1002  REM ******************
1010  ESC$ =  CHR$ (27):FINISH$ =  CHR$ (3)
1020  PSLOT = 4: PRINT D$;"PR#";PSLOT
1030  PRINT "IN;SP";PEN: REM * INITIALISE PLOTTER *
```

```
1040  PRINT ES$;".M0;63;0;13:";ES$;".N5:"
1050  PRINT ES$;".H80;18;49:"
1060  PRINT "VS";SP: REM * PEN SPEED *
1070  PRINT "VS";SP
1100  GOSUB 3000
1105  IF RE$ < > "Y" THEN 1200: REM * DO NOT PLOT REGRESSION
      LINE *
1110  I = 2:X(1) = R(0,0):Y(1) = R(0,1)
1120  X(2) = R(1,0):Y(2) = R(1,1)
1130  JP$ = "Y": GOSUB 3210
1200  PRINT D$;"PR#0"
1210  HOME : VTAB 10: PRINT "DO YOU WISH TO ANALYSE FURTHER
      DATA? Y/N ";: GET A$: IF A$ = "Y" THEN  RUN
1220  IF A$ < > "N" THEN 1210
1230  END
1999  REM *********
2000  REM * LABEL *
2001  REM *********
2040  PRINT "LB";L$ + FIN$: PRINT "PU"
2050  RETURN
3000  REM ********
3010  REM * AXES *
3020  REM ********
3040  PRINT "PU"
3050  PRINT "PA";9000 * CX + OX;",";779 + OY
3060  PRINT "PD"
3064  REM ***************
3065  REM * DRAW X-AXIS *
3066  REM ***************
3070  FOR L = 9000 * CX + OX TO 750 + OX STEP  - (9000 * CX -
      750) / NXTICKS
3080  PRINT "PA";L;",","779 + OY
3090  PRINT "XT"
3100  NEXT
```

```
3104 REM ***************
3105 REM * DRAW Y-AXIS *
3106 REM ***************
3110 FOR L = 7000 * CY + OY TO 779 + OY STEP  - (7000 * CY -
     779) / NYTICKS
3120 PRINT "PA";750 + OX",";L
3130 PRINT "YT"
3140 NEXT
3145 PRINT "PA";750 + OX;",";779 + OY: PRINT "YT": REM *
     LOWEST Y VALUE *
3150 PRINT "PU"
3151 REM ****************
3152 REM * LABEL Y-AXIS *
3153 REM ****************
3155 ADD = (HY - LY) / NY:II = HY + ADD
3160 FOR L = 7000 * CY + OY TO 779 + OY STEP  - (7000 * CY -
     779) / NYTICKS
3170 PRINT "PA";150 + OX;",";L - 50
3180 II = II - ADD
3182 L$ =  STR$ (II): FOR ZZ = 1 TO 4: IF  LEN (L$) <  LEN
     (HY$) THEN L$ = " " + L$: NEXT ZZ
3183 GOSUB 2000
3184 NEXT L
3185 PRINT "PA";150 + OX;",";779 + OY - 50:L$ =  STR$ (LY):
     FOR ZZ = 1 TO 4: IF  LEN (L$) < 4 THEN L$ = " " + L$:
     NEXT ZZ
3186 GOSUB 2000: REM * LOWEST Y VALUE *
3187 REM ****************
3188 REM * LABEL X-AXIS *
3189 REM ****************
3190 ADD = (HX - LX) / NX:II = HX + ADD
3192 FOR L = 9000 * CX + OX TO 750 + OX STEP  - (9000 * CX -
     750) / NXTICKS
3194 PRINT "PA";L - 50;",";479 + OY
3196 II = II - ADD
3197 L$ =  STR$ (II)
3198 GOSUB 2000
3199 NEXT L
```

```
3200 REM ****************
3201 REM * PLOT ROUTINE *
3202 REM ****************
3210 FOR II = 1 TO I
3220 PRINT "PA";((X(II) - LX) * (9000 * CX - 750) / (HX -
     LX)) + 750 + OX;",";((Y(II) - LY) * (7000 * CY - 779) /
     (HY - LY)) + 779 + OY
3225 GOSUB 3300
3230 NEXT II
3240 PRINT "PU"
3250 IF RE$ = "Y" AND GL = 1 THEN  RETURN
3255 GL = 1
3260 GOSUB 4000
3270 RETURN
3290 REM **************
3300 REM * PLOT CROSS *
3301 REM **************
3305 PRINT "PU"
3307 PRINT "PR";0;",";  - 50
3310 PRINT "PD"
3320 PRINT "PR";0;",";100
3322 PRINT "PU"
3324 PRINT "PR";  - 50;",";  - 50
3326 PRINT "PD"
3328 PRINT "PR";100;",";0
3330 PRINT "PU"
3332 PRINT "PR";  - 50;",";0
3333 IF JP$ = "N" THEN  RETURN
3334 PRINT "PD": REM * JOIN POINTS *
3340 RETURN
3999 REM **************
4000 REM * LABEL PLOT *
4001 REM **************
4010 L$ = GL$: GOSUB 4020: REM * LABEL GRAPH *
4015 L$ = N1$: GOSUB 4020: GOTO 4100: REM * LABEL X-AXIS *
4020 X = (9000 * CX - 750) / 100:X = (X -  LEN (L$)) / 2 *
     100 + 750
4030 Y = 7000 * CY + 500
```

```
4035  IF L$ = N1$ THEN Y = 279
4050  PRINT "PU"
4060  PRINT "PA";X + OX;",";Y + OY
4070  GOSUB 2000
4080  PRINT "PU"
4090  RETURN
4095  REM ****************
4100  REM * LABEL Y-AXIS *
4105  REM ****************
4110  L$ = N2$
4120  Y = (7000 * CY - 779) / 100:Y = (Y -  LEN (L$)) / 2 *
      100 + 750
4130  X = 750 - ( LEN (HY$) * 100 + 300)
4140  PRINT "PU"
4150  PRINT "DR 1,30"
4160  PRINT "PA";X + OX;",";Y + OY
4165  PRINT "DR 1,100": REM * DRAW Y-LABEL VERTICALLY *
4170  GOSUB 2000
4180  PRINT "PU"
4190  PRINT "DR 1,0"
4200  RETURN
```

SECTION 5

CO2RESPONSE SUITE

```
4    REM ****************************************************
5    REM * HELLO-CO2 : TIMER CARD IN SLOT 3 : ADC IN SLOT 4 *
6    REM ****************************************************
30   HOME
40   D$ =  CHR$ (4)
50   Z$ = "CO2 RESPONSE PROGRAMS": HTAB (40 -  LEN (Z$)) / 2:
     PRINT Z$
60   VTAB 10: PRINT "1. COLLECT DATA
70   VTAB 13: PRINT "2. PRINT OUT DATA
80   VTAB 16: PRINT "3. ANALYSE RESULTS"
90   VTAB 20: HTAB 1: PRINT "SELECT PROGRAM REQUIRED ";: GET A$:
      IF A$ < "1" OR A$ > "3" THEN 9
100  PRINT A$: ON  VAL (A$) GOTO 12,13,14
110  PRINT D$;"RUN CO2SPACE"
120  PRINT D$;"RUN CO2PRINT"
130  PRINT D$;"RUN CO2STAT"
```

```
    REM ***********************************************
    REM * CO2SPACE - MODIFICATION OF COMPUTEC UTILITY *
    REM ***********************************************
0   REM  SPACE (C) SGL 22/08/79
100 SP =  PEEK (106) * 256 +  PEEK (105) - 128
110 CALL SP
120 X =  PEEK (513)
130 IF X = 255 THEN 200
140 X = X * 256 +  PEEK (512)
150 PRINT X;" SECTORS ("; INT (X / 4.96);"%)";
152 PL =  INT (X / 4.96)
154 IF PL < 5 THEN  HOME : VTAB 10: HTAB 10: PRINT
    "";"";"DISC FULL": END
155 PRINT
158 D$ =  CHR$ (4): PRINT D$;"RUN CO2RESPONSE"
200 PRINT  CHR$ (7);"I/O ERROR "; PEEK (512);" READING VTOC";
210 END
```

```
4    REM ************
5    REM * CO2SETUP *
6    REM ************
10   D$ =  CHR$ (4)
140  ONERR  GOTO 220
144  REM **********************
145  REM * OBTAIN CURRENT NAME *
146  REM **********************
150  PRINT D$;"NOMONI,C,O"
160  PRINT D$;"OPEN NAMEFILE"
170  PRINT D$;"READ NAMEFILE"
180  INPUT NA$: INPUT NO$
190  PRINT D$;"CLOSENAMEFILE"
200  HOME : VTAB 10: PRINT "FILE FOR ";NA$;" Y/N ";: GET A$:
     IF A$ = "Y" THEN  POKE 216,0: GOTO 290
210  IF A$ <  > "N" THEN 200
220  POKE 216,0
230  GOSUB 380
240  PRINT D$;"OPEN NAMEFILE"
250  PRINT D$;"WRITENAMEFILE"
260  PRINT NA$: PRINT NO$
270  PRINT D$;"CLOSENAMEFILE"
280  HOME
290  VTAB 20: HTAB 1: PRINT "LOADING THE NEXT PROGRAMME"
300  PRINT D$;"RUN CO2RESPONSE"
320  REM * ADD LEADING ZEROS *
330  IF  LEN (TT$) < TT THEN TT$ = "0" + TT$: GOTO 330
340  RETURN
350  REM ADD TRAILING DOTS
355  IF  LEN (TT$) < TT THEN TT$ = TT$ + ".": GOTO 355
360  RETURN
377  REM ********************
378  REM * ADDNAME SUBROUTINE*
379  REM ********************
380  HOME : VTAB 1: HTAB 1: PRINT "TYPE 'SURNAME' <SPACE>
     'INITIAL'": PRINT : PRINT "THEN TYPE <RETURN>   ": VTAB 5:
     INPUT NA$
390  IF  LEN (NA$) > 20 THEN 440
```

```
400  FOR I = 1 TO 20:A$ =  MID$ (NA$,I,1): IF A$ < "0" OR A$ >
     "9" THEN  NEXT : GOTO 420
410  GOTO 440
420  FOR I = 3 TO 20: IF  MID$ (NA$,I,1) = " " THEN 460
430  NEXT
440  FOR I = 1 TO  LEN (NA$): PRINT " ";: PRINT : PRINT "ONLY
     20 CHARS INCLUDING 1 SPACE ALLOWED": FOR I = 1 TO 2000:
     NEXT :OTO 380
460  VTAB 10: HTAB 1: CALL  - 868: VTAB 11: CALL  - 868
470  VTAB 10: PRINT "TYPE REFERENCE NO. THEN <RETURN> ";:
     INPUT NO$
480  FOR I = 1 TO  LEN (NO$): IF  MID$ (NO$,I,1) < "0" OR
     MID$ (NO$,I,1) > "9" THEN NO$ = "11111111"
490  NEXT
500  IF  LEN (NO$) > 6 OR  LEN (NO$) < 3 THEN  VTAB 10: HTAB
     1: CALL  - 868: VTAB 11: CALL  - 868: VTAB 10: PRINT "MIN
     OF 3 & MAX OF 6 DIGITS ALLOWED": FOR I = 1 TO 3000: NEXT :
     GOTO 460
510  TT$ = NO$:TT = 6: GOSUB 320:NO$ = TT$
590  RETURN
```

```
1    REM ***************
2    REM * CO2RESPONSE *
3    REM ***************
10   LOMEM: 28000:D$ =  CHR$ (4): TEXT : PRINT
     D$;"NOMON,I,C,O": GOSUB 5000: REM * OPEN SHUTTER *
20   GOTO 520: REM *
180  REM *******************
190  REM * DATA COLLECTION *
195  REM *******************
200  TE = TE + 1: REM * COUNTER TO OMIT FIRST BREATH *
210  IC = IC + 1: POKE CHAN,VT:VT%(IC) =  PEEK (ADC): POKE
     CHAN,IE: IF  PEEK (ADC) > 25 AND IC < 400 THEN 210
211  REM *******************************
212  REM * OBTAIN TIME FOR INSPIRATION *
213  REM *******************************
215  IF CLOCK$ = "ST" THEN  CALL 775: GOTO 225: REM *
     STRATHAND TIMER CARD (ST) *
220  CALL 768: REM * MOUNTAIN HARDWARE CLOCK/CALENDAR CARD
     (MH) * 221  IF TE < 2 THEN 225
222  TO$ = "": FOR I = 0 TO 5:TO$ = TO$ +  CHR$ ( PEEK (27780
     + I) - 128): NEXT : REM  * GET START TIME FOR PREVIOUS
     BREATH FROM MH CLOCK *
224  REM **********************
225  REM * OBTAIN INSPIRED VT *
226  REM **********************
228  VINST = 0
230  FOR J = 1 TO IC: IF VT%(J) > VINST THEN VINST = VT%(J)
240  NEXT
245  FOR I = 1 TO 750: NEXT : REM * WAIT .75S BEFORE MEASURING
     CO2 *
250  POKE  - 16295,1: REM * CLOSE SHUTTER *
257  REM *******************************
255  REM * GET EXP DATA, P50, P100 AND *
256  REM * START TIME FOR NEXT BREATH  *
257  REM *******************************
260  IF CLOCK$ = "ST" THEN  CALL 794: GOTO 280: REM *
     STRATHAND TIMER *
270  CALL 773: REM * OBTAIN DATA *
```

```
280  IF TE < 2 THEN 200
285  REM * PRESSURES *
290  N = N + 1:P100(N) =  PEEK (ADC) * SP * .09806: REM *
     DIGITAL VALUE -> CMH2O -> KPA *
305  P50(N) =  PEEK (27809) * SP * .09806
306  REM * END-TIDAL CO2 *
310  CO2 =  PEEK (27807) * SGAS * (PB / 100) * 0.133: REM *
     DIGITAL VALUE -> %CO2 -> MMHG -> KPA *
311  REM *****************************
312  REM * OBTAIN INSP AND EXP TIMES *
313  REM *****************************
315  IF CLOCK$ = "ST" THEN 400
320  T1$ = "":T2$ = "": REM * MH CLOCK *
340  FOR I = 0 TO 5:T1$ = T1$ +  CHR$ ( PEEK (27786 + I) -
     128): NEXT
350  FOR I = 0 TO 5:T2$ = T2$ +  CHR$ ( PEEK (27792 + I) -
     128): NEXT
360  T0 =  VAL (T0$):T1 =  VAL (T1$):T2 =  VAL (T2$)
370  IF T1 < T0 THEN T0 = T0 - 60
380  IF T2 < T1 THEN T2 = T2 + 60
390  IT = T1 - T0:ET = T2 - T1: GOTO 414: REM * MH *
400  IT =  PEEK (27802) + 256 *  PEEK (27803): REM * ST *
410  T2 =  PEEK (27804) + 256 *  PEEK (27805)
411  ET = (T2 - IT) / 1E3
412  IT = IT / 1E3
413  REM ****************************
414  REM * CALCULATIONS AND DISPLAY *
415  REM ****************************
417  MFLOW(N) =  INT (VINST * SI / IT)
418  RTE(N) =  INT (600 / (IT + ET) + .5) / 10:EI(N) =  INT
     (ET / ( T + IT) * 100 + .5) / 100
419  VTAB 4: HTAB 1: PRINT "RATE = ";RTE(N),:VINST = VINST *
     SI: PRINT "TIDAL VOL ="; INT (VINST);" ML   ": PRINT
420  VTAB 6: PRINT "IT="; INT (IT * 100 + .5) / 100,"ET="; INT
     (ET * 100 + .5) / 100
422  PRINT : PRINT "MEAN FLOW RATE = ";MFLOW(N)" ML/SEC"
425  PRINT
430  VINST(N) =  INT (RTE(N) * VINST) / 1E3
```

```
450  PRINT "INSTANT MIN VOL = ";VINST(N);" L": PRINT : PRINT
452  CO2(N) =  INT (CO2 * 100 + .05) / 100
455  PRINT : PRINT "END TIDAL CO2 = ";CO2EX(N);"KPA   "
457  P100(N) =  INT (P100(N) * 1E3 + .5) / 1E3:P50(N) =  INT
     (P50(N) * 1E3 + .5) / 1E3
458  PRINT : PRINT "OCCLUSION PRES = ";P100(N);" KPA";"
     ";P50(N);"KPA  ";
461  VTAB 22:X$ = "TYPE 1 TO STOP": GOSUB 1060
465  IC = 0:VINST = 0
466  KEYPR =  PEEK ( - 16384): POKE  - 16368,0
467  IF KEYPRES = 177 THEN N = N - 1: GOTO 1130: REM * OMIT
     LAST READING *
470  IF RTE(N) > 35 OR P100(N) < .03 OR VINST(N) < 1 OR CO2(N)
     < 4 THEN N = N - 1
480  GOTO 200
499  REM **********************************
500  REM * CHANNELS/SUBROUTINES/ CALLS      *
501  REM *--------------------------------*
504  REM * CHAN 3 = CO2 (C2) (RED WIRE)     *
506  REM * CHAN 7 = VT (VT) (BROWN WIRE)    *
508  REM * CHAN 11 = I/E (IE) (ORANGE WIRE) *
510  REM * CHAN 0 = -VE PRES                *
511  REM * TIMER SLOT3 ADC SLOT 4           *
512  REM * SUBR. AT 5000 = CLOSE SHUTTER    *
513  REM * SUBR. AT 5050 = OPEN SHUTTER     *
514  REM *--------------------------------*
515  REM *       MH CLOCK/CALENDER          *
516  REM * CALL 768 RETURNS INSPIR'Y TIME   *
517  REM * CALL 775 RETURNS EXPIR'Y TIME    *
518  REM *--------------------------------*
519  REM *         STRATHAND TIMER          *
520  REM * CALL 775 OBTAINS INSPIR'Y TIME   *
521  REM * CALL 794 OBTAINS EXPIR'Y TIME    *
520  REM **********************************
523  HOME :X$ = "GLASGOW RESPIRATORY": GOSUB 1060:X$ =
     "INTENSIVE CARE UNIT": GOSUB 1060:X$ = "CO2  RESPONSE
     PROGRAMME": GOSUB 1060
```

```
524  REM *****************
525  REM * SET VARIABLES *
526  REM *****************
533  IE = 10:VT = 7:PR = 0:C2 = 3
534  ADC =  - 16384 + 256 * 4:CHAN = ADC + 1
538  IC = 0:TE = 0:N = 0
541  N = 0: DIM VT%(400),VI(250),CO2(250),RTE(250),EI(250),
     P100(250),P50(250),MFLOW(250)
546  POKE  - 16368,0: REM * CLEAR KEYBOARD STROBE *
550  REM ****************
560  REM * SET BAR PRES *
570  REM ****************
580  PB = 760
590  HOME : VTAB 5: PRINT "BAR PRESS SET AT ";PB;" MMHG"
592  PRINT : PRINT "DO YOU WISH TO ALTER THIS ?  Y/N ";: GET
     A$: IF A$ = "N" THEN 600
594  IF A$ < > "Y" THEN 590
595  HTAB 1: VTAB 15: PRINT "TYPE IN THE NEW BAR PRESS (MMHG)
     ";: INPUT PB
596  VTAB 15: HTAB 1: CALL  - 868: GOTO 590
597  REM *********************************
598  REM * SELECT CALIBRATION SUBROUTINE *
599  REM *********************************
600  HOME : PRINT "CALIBRATE  Y/N  ";: GET A$: PRINT A$: IF A$
     = "Y" THEN  GOSUB 3000: GOTO 620
602  IF A$ < > "N" THEN 600
604  ONERR  GOTO 608
606  GOTO 610
608  POKE 216,0:X$ = "NO CALIBRATION VALUES ON DISC ": VTAB
     10: GOSUB 1060: FOR I = 1 TO 2000: NEXT : GOSUB 3000: GOTO
     620
610  X$ = "CALVALUES"
613  PRINT D$;"OPEN";X$: PRINT D$;"READ";X$
615  INPUT SI: REM *VOL
616  INPUT SP: REM *PRES
617  INPUT SGAS: REM *CO2
618  PRINT D$;"CLOSE"
619  POKE 216,0
```

```
620  PRINT D$;"OPEN NAMEFILE"
630  PRINT D$;"READ NAMEFILE"
640  INPUT NA$: INPUT NO$
645  PRINT D$;"CLOSE NAMEFILE"
650  REM **********************
651  REM * SET PRES THRESHOLD *
652  REM **********************
653  PT = .02: HOME
654  VTAB 10: HTAB 1: PRINT "PRESENT PRESSURE THRESHOLD SET AT
     "; PT;"KPA    "
656  PRINT : PRINT "DO YOU WISH TO ALTER THIS ?  Y/N ";: GET
     A$: IF A$ = "N" THEN 664
658  IF A$ <  > "Y" THEN 654
660  HTAB 1: VTAB 15: PRINT "TYPE IN THE NEW THRESHOLD (KPA)
     ";: INPUT PT
662  VTAB 15: HTAB 1: CALL  - 868: GOTO 654
664  POKE 27808,PT / .09806 / SP: REM * KPA -> CMH20 ->
     DIGITAL VALUE *
665  REM ***************************
666  REM * SELECT CLOCK/TIMER CARD *
667  REM ***************************
669  HOME :X$ = "CLOCK / TIMER IN SLOT 3": GOSUB 1060
670  VTAB 5: PRINT "1. MOUNTAIN HARDWARE CLOCK/CALENDER
680  VTAB 8: PRINT "2. STRATHAND TIMER CARD"
690  VTAB 15: PRINT "TYPE IN THE NUMBER OF THE CARD USED ";:
     GET A$: PRINT A$: IF A$ = "1" THEN CLOCK$ = "MH": GOTO 720
700  IF A$ = "2" THEN CLOCK$ = "ST": GOTO 730
710  GOTO 660
720  PRINT D$;"BLOAD CO2+P50+P100CLOCK.OBJ0": GOTO 740: REM *
     MH CLOCK *
730  PRINT D$;"BLOAD CO2+P50+P100.OBJ0": REM * ST TIMER *
740  HOME : VTAB 10:X$ = "*** RESET PNEUMOTACHOGRAPH ***":
     GOSUB 1060: VTAB 14:X$ = "PRESS <RETURN> WHEN RESET IS
     COMPLETE":
     GOSUB 1060
742  VTAB 16:X$ = "AND PATIENT IS READY FOR TEST  ": GOSUB
     1060: VTAB 20: HTAB 20: GET A$: IF A$ =  CHR$ (13) THEN
     PRINT A$:  GOTO 747
```

```
744  GOTO 740
747  REM *****************************************************
748  REM * WAIT FOR CHANGEOVER FROM EXPIRATION TO INSPIRATION *
749  REM *****************************************************
750  POKE CHAN,IE: IF  PEEK (ADC) > 25 THEN 750: REM * INSP *
760  POKE CHAN,IE: IF  PEEK (ADC) < 25 THEN 760: REM * EXP *
770  GOTO 200
1030  REM *************
1040  REM * CENTRE X$ *
1050  REM *************
1060  PRINT : HTAB (40 -  LEN (X$)) / 2: PRINT X$: RETURN
1090  REM ************************
1100  REM * PRINT DATA ON SCREEN *
1110  REM ************************
1130  HOME : FOR I = 1 TO N: PRINT "MEASUREMENT NO ";I
1135  PRINT "VINS=";VI(I);: HTAB 24: PRINT "CO2=";CO2(I)
1140  PRINT "RATE=";RTE(I);: HTAB 12: PRINT "P100=";P100(I);:
      HTAB 24: PRINT "P50=";P50(I): PRINT
1145  IF  INT (I / 5) - I / 5 = 0 THEN  PRINT "TO SEE NEXT 5
      RESULTS TYPE A KEY ";: GET A$: PRINT : PRINT
1150  NEXT
1160  HOME : VTAB 10: PRINT "WRITING FILES TO DISC"
1170  PRINT D$;"MONI,C,O": GOSUB 2700
1180  HOME : VTAB 3: PRINT "1. ANOTHER RUN": VTAB 6: PRINT
      "2. LOAD PRINT PROGRAMME": VTAB 9: PRINT "3. QUIT"
1190  HTAB 1: VTAB 20: PRINT "TYPE NO. REQUIRED  ";: GET A$:
      IF A$ < "1" OR A$ > "3" THEN 1190
1200  PRINT
1210  IF A$ = "3" THEN  END
1220  VTAB 20: HTAB 1: PRINT "THE REQUIRED PROGRAMME IS
      LOADING"
1230  IF A$ = "1" THEN  PRINT D$;"RUN CO2SETUP"
1240  PRINT D$;"RUN CAMPRINT"
2699  REM  *********************
2700  REM  * WRITE DATA TO DISC *
2701  REM  *********************
2710  DA$ = "DATE":TI$ = "TIME"
2720  PRINT D$;"OPEN";NA$ + "CO2"
```

```
2730  PRINT D$;"WRITE";NA$ + "CO2"
2740  PRINT DA$: PRINT TI$: PRINT N
2750  FOR I = 1 TO N: PRINT CO2(I): PRINT VINST(I): PRINT
      P100(I): PRINT P50(I): PRINT RTE(I): PRINT EI(I): PRINT
      MFLOW(I): NEXT I
2770  RETURN
2999  REM *************
3000  REM * CALIBRATE *
3001  REM *************
3005  HOME
3007  VINST = 0
3010  VTAB 3: PRINT "1. CALIBRATE VOLUME"
3020  VTAB 10: PRINT "ATTACH 1 LITRE SYRINGE TO PATIENT
      CIRCUIT AND MOVE THE PLUNGER FULLY IN   AND OUT.
3030  IC = IC + 1: POKE CHAN,VT:VT%(IC) =  PEEK (ADC): POKE
      CHAN,IE: IF  PEEK (ADC) > 25 AND IC < 400 THEN 3030
3040  FOR J = 1 TO IC: IF VT%(J) > VINST THEN VINST = VT%(J)
3050  NEXT
3051  VTAB 18: PRINT "VM = "; INT (VINST);"    "
3052  VTAB 20: PRINT "VT = "; INT (VINST * SI);"    "
3055  IF VINST = 0 THEN 3030
3060  IF VINST * SI > 980 AND VINST * SI < 1020 THEN 3100
3070  SI = 1000 / VINST:IC = 0:VINST = 0
3080  POKE CHAN,IE: IF  PEEK (ADC) < 100 THEN 3080
3090  GOTO 3030
3094  REM *********************************
3095  REM * CALIBRATE PRESSURE TRANSDUCER *
3096  REM *********************************
3100  FOR I = 1 TO 5: PRINT  CHR$ (7): NEXT : VTAB 22:X$ =
      "VOLUME OK": GOSUB 1060: FOR I = 1 TO 3000: NEXT
3110  HOME : VTAB 3: PRINT "2. CALIBRATE PRESSURE"
3120  VTAB 10: PRINT "ATTACH MANOMETER TO PRESSURE SENSING
      CIRCUIT AND ESTABLISH 8-9 CMH2O PRESSURE
3130  VTAB 14: PRINT "TYPE IN THE CALIBRATION PRESSURE IN
      CM H2O AND PRESS THE <RETURN> KEY
3140  VTAB 18: INPUT P
3150  FOR I = 1 TO 100: POKE CHAN,PR:PO = PO +  PEEK (ADC):
      NEXT
```

```
3160  PO = PO / 100
3165  IF PO = 0 THEN 3100
3170  SP = P / PO: FOR I = 1 TO 5: PRINT  CHR$ (7): NEXT
3180  VTAB 20:X$ = "PRESSURE OK": GOSUB 1060: FOR I = 1 TO
      3000: NEXT : HOME
3184  REM ***************************
3185  REM * CALIBRATE CO2 ANALYSER *
3186  REM ***************************
3200  HOME :X$ = "CALIBRATE CO2 ANALYSER": GOSUB 1060
3202  VTAB 8:X$ = "SET ZERO ON ANALYSER": GOSUB 1060
3204  VTAB 11:X$ = "THEN TYPE ANY KEY": GOSUB 1060
3206  VTAB 18: HTAB 19: POKE CHAN,C2: PRINT  PEEK (ADC);"  ":
      IF PEEK ( - 16384) < 127 THEN 3206
3207  POKE  - 16368,0:CG = 7
3208  HOME : VTAB 5: HTAB 8: PRINT "CALIBRATION GAS (%) = ";CG
3209  PRINT : PRINT "DO YOU WISH TO ALTER THIS? Y/N ";: GET
      A$: IF A$ = "N" THEN 3228
3210  IF A$ <  > "Y" THEN 3208
3215  HTAB 1: VTAB 15: PRINT "TYPE IN THE NEW CONCENTRATION
      (%) ";: INPUT CG
3220  VTAB 15: HTAB 1: CALL  - 868: GOTO 3208
3228  HOME : VTAB 3:X$ = "ATTACH CALIBRATION GAS": GOSUB
      1060:X$ = "TO THE INPUT LINE OF THE ANALYSER": GOSUB
      1060
3230  VTAB 12: HTAB 6: PRINT "ADJUST GAIN TO GIVE "; INT (255
      / 10 * CG);"-"; INT (255 / 10 * CG + 1):X$ = "THEN PRESS
      ANY KEY": GOSUB 1060
3240  VTAB 18: HTAB 19: POKE CHAN,C2: PRINT  PEEK (ADC);"  ":
      IF  PEEK ( - 16384) < 127 THEN 3240
3250  POKE  - 16368,0
3270  SGAS = 10 / 255: REM * 10%CO2 = 255 *
3280  VTAB 20:X$ = "CO2 CALIBRATION OK": GOSUB 1060: FOR I = 1
      TO 3000: NEXT : HOME
3390  REM ***************************
3400  REM * SAVE CAL VALUES TO DISC *
3401  REM ***************************
3410  X$ = "CALVALUES"
3420  PRINT D$;"OPEN";X$
```

```
3430  PRINT D$;"WRITE";X$
3440  PRINT SI: PRINT SP: PRINT SGAS
3450  PRINT D$;"CLOSE"
3460  RETURN
4990  REM ********************************
5000  REM * TURN OFF GAMES PORT SWITCHES *
5005  REM ********************************
5010  POKE  - 16296,1: POKE  - 16294,1: POKE  - 16292,1: POKE
      - 16290,1: RETURN
5045  REM ***************************
5050  REM * CLOSE OCCLUSION SHUTTER *
5055  REM ***************************
5060  POKE  - 16295,1: RETURN
```

```
1    REM ************
2    REM * CO2PRINT *
3    REM ************
4    TEXT
5    HOME : VTAB 10: PRINT "PUT DISC IN DRIVE AND TYPE A KEY"
7    DIM CO2(300),VI(300),PO(300),RTE(300),EI(300),MF(300)
10   D$ =  CHR$ (4)
12   PRINT D$,"NOMONI,C,O"
15   GET A$
17   ONERR  GOTO 90
20   HOME : VTAB 10: PRINT "PRINTER Y/N ";: GET A$: IF A$ = "Y"
     THEN PR = 1: GOTO 50
30   IF A$ = "N" THEN PR = 0: GOTO 50
40   GOTO 20
50   HOME : VTAB 10: PRINT "TYPE THE NAME FOR FILE RETRIEVAL":
     PRINT : PRINT "FOLLOWED BY THE <RETURN> KEY".
60   VTAB 20: INPUT NA$
70   GOTO 100
90   HOME : VTAB 10: FOR I = 1 TO 5: PRINT "": NEXT : HTAB 12:
     PRINT "FILE UNAVAILABLE": FOR I = 1 TO 3000: NEXT : POKE
     216,0: PRINT D$;"CLOSE": VTAB 20: GOTO 310
100  GOSUB 2700
170  REM *************************
180  REM * DISPLAY DATA ON SCREEN *
190  REM *************************
200  IF PR = 1 THEN 270
205  HOME : FOR I = 1 TO N: PRINT "MEASUREMENT NO ";I
210  PRINT "VINS=";VI(I);: HTAB 24: PRINT "CO2=";CO2(I)
215  PRINT "RATE=";RTE(I);: HTAB 12: PRINT "P100=";P100(I);:
     HTAB 24: PRINT "P50=";P50(I): PRINT
220  IF  INT (I / 5) - I / 5 = 0 THEN  PRINT "TO SEE NEXT 5
     RESULTS TYPE A KEY ";: GET A$: PRINT : PRINT
225  NEXT
230  NEXT
240  GOTO 300
```

```
255  REM ************************
260  REM * PRINT DATA ON PRINTER *
265  REM ************************
270  PRINT D$;"PR#1"
280  FOR I = 1 TO N: PRINT "MEASUREMENT NO ";I
285  PRINT "VINS=";VI(I);: HTAB 24: PRINT "CO2=";CO2(I)
290  PRINT "RATE=";RTE(I);: HTAB 12: PRINT "P100=";P100(I);:
     HTAB 24: PRINT "P50=";P50(I): PRINT
295  PRINT: PRINT: PRINT
300  PRINT D$;"PR#0"
310  PRINT "MORE Y/N ";: GET A$: IF A$ = "Y" THEN 50
320  IF A$ = "N" THEN  PRINT A$: PRINT D$;"RUN HELLO"
330  GOTO 310
1000  X$ = "NAMEFILE"
1010  PRINT D$;"OPEN";X$
1020  PRINT D$;"READ";X$: INPUT NA$
1030  PRINT D$;"CLOSE"
1040  RETURN
2690  REM ************************
2700  REM * OBTAIN DATA FROM DISC *
2701  REM ************************
2720  PRINT D$;"OPEN";NA$ + "CO2"
2730  PRINT D$;"READ";NA$ + "CO2
2740  INPUT DA$: INPUT TI$: INPUT N
2750  FOR I = 1 TO N: INPUT CO2(I): INPUT VINST(I): INPUT
      P100(I): INPUT P50(I): INPUT RTE(I): INPUT EI(I): INPUT
      MFLOW(I): NEXT I
2750  PRINT D$;"CLOSE"
2760  RETURN
```

```
1   REM ***********
2   REM * CO2STAT *
3   REM ***********
5   GOTO 10
6   HTAB (40 -  LEN (Z$)) / 2: PRINT Z$;: RETURN
10  D$ =  CHR$ (4)
20  DIM N(5,250,1),X(250,1)
30  PRINT D$;"  MONI,C,O"
35  HOME :Z$ = "INSERT CO2 RESPONSE DATA DISC": VTAB 5: GOSUB
    6: Z$ = "AND PRESS <SPACEBAR>": VTAB 10: GOSUB 6
37  VTAB 15: HTAB 20: GET A$: IF A$ <  > " " THEN 37
40  HOME : VTAB 5: INPUT "FILENAME - ";F$
42  REM ***********************
43  REM * READ FROM DATA DISC *
44  REM ***********************
45  F$ = F$ + "CO2"
50  PRINT D$;"OPEN ";F$
60  PRINT D$;"READ ";F$
70  INPUT DAY$: INPUT TIME$: INPUT RN
80  FOR I = 1 TO RN
90  INPUT N(0,I,0): REM * CO2 *
100  FOR J = 0 TO 5: INPUT N(J,I,1): NEXT
110  NEXT I
120  PRINT D$;"CLOSE"
130  HOME :Z$ = "INSERT STATISTIC DISC": VTAB 5: GOSUB 6:
     VTAB 10:Z$ = "AND PRESS <SPACEBAR>": GOSUB 6
140  VTAB 15: HTAB 20: GET A$: IF A$ <  > " " THEN 140
150  ONERR  GOTO 300
154  REM **********************
155  REM * WRITE TO STAT DISC *
156  REM **********************
160  PRINT D$;"OPEN";F$
170  PRINT D$;"WRITE";F$
180  PRINT DAY$: PRINT TIME$: PRINT RN
190  FOR I = 1 TO RN
200  PRINT N(0,I,0)
210  FOR J = 0 TO 5: PRINT N(J,I,1): NEXT
220  NEXT I
```

```
230  PRINT D$;"CLOSE"
240  HOME : VTAB 10:Z$ = "CALCULATIONS PROGRESSING": GOSUB 6
250  GOTO 700
299  REM ********************
300  REM * DISC WRITE ERROR *
301  REM ********************
310  POKE 216,0:X =  PEEK (222): IF X = 9 THEN 330
320  HOME : VTAB 10:Z$ = "** DISC ERROR **": GOSUB 6: PRINT
     CHR$ (7); CHR$ (7): STOP
330  HOME : VTAB 10:Z$ = "** STATISTIC DISC IS FULL **": GOSUB
     6: VTAB 15:Z$ = "** PLEASE INSERT ANOTHER DISC **": GOSUB 6
340  PRINT  CHR$ (7); CHR$ (7): FOR I = 1 TO 3000: NEXT : GOTO
     130
699  REM *************************
700  REM * MAIN ANALYSIS ROUTINE *
701  REM *************************
703  FILE = RN: GOSUB 720: GOSUB 805
705  SECO2 = 6: GOSUB 720: GOSUB 805:SE = 7: GOSUB 720: GOSUB
     805: SE = 8: GOSUB 720: GOSUB 805
707  HOME : VTAB 10:Z$ = "END": GOSUB 6: END
714  REM *****************************************
715  REM * OBTAIN X-ARRAY FOR COR COEF & REGRESS *
716  REM *   N-ARRAY HAS ENTIRE DATA FROM DISC   *
717  REM *****************************************
720  FOR I = 0 TO 7:CO(I) = 0:MX(I) = 0:C(I) = 0: NEXT
722  FOR J = 0 TO 5
724  K = 0
725  FOR I = 1 TO FILE
730  X(I - K,0) = N(0,I,0):X(I - K,1) = N(J,I,1)
735  IF X(I - K,0) = 0 OR X(I - K,1) = 0 OR X(I - K,0) < SE
     THEN K = K + 1: REM * UNACCEPTABLE VALUE *
740  NEXT :RN = FI - K
745  IF I - K < 5 THEN 770
747  GOSUB 1000: REM * CORRELATION CO-EFFICIENT *
748  GOSUB 2000: REM * REGRESSION EQUATION *
750  COEF(J) = RV
760  MX(J) = C1
765  C(J) = C2
```

```
770  FOR I = 1 TO FILE:X(I,0) = 0:X(I,1) = 0: NEXT
780  NEXT J
785  E =  LOG (10):K = 0
787  FOR J = 1 TO 2: REM * 1=P100 2=P50 *
790  FOR I = 1 TO FILE
792  X(I - K,0) = N(0,I - K,0):X(I - K,1) =  LOG (N(J,I,1)) /
     E
793  IF X(I - K,0) = 0 OR X(I - K,1) = 0 OR X(I - K,0) < SE
     THEN K = K + 1: REM * UNACCEPTABLE VALUE *
794  NEXT :RN = FI - K: GOSUB 1000: GOSUB 2000
795  COEF(5 + J) = RV
796  MX(5 + J) = C1
797  C(5 + J) = C2
798  NEXT J
799  RETURN
800  REM ************************
802  REM * DISPLAY CALCULATIONS *
805  REM ************************
806  HOME : VTAB 3: PRINT "VOLUME": VTAB 5: PRINT "P 100":
     VTAB 7: PRINT "P 50"
807  VTAB 9: PRINT "RATE": VTAB 11: PRINT "IE RATIO": VTAB 13:
     PRINT "MEAN FL"
808  VTAB 16: PRINT "LOG P100": VTAB 18: PRINT "LOG P50"
810  VTAB 1:X = 1
820  HTAB 12: PRINT "COR COEF";: HTAB 25: PRINT "REGRESSION "
830  FOR I = 1 TO 8: VTAB I * 2 + X: HTAB 12: PRINT CO(I);:
     HTAB 25: PRINT MX(I);: HTAB 32: PRINT C(I): IF I = 6
     THEN X = X + 1
840  NEXT
900  VTAB 1: HTAB 1: PRINT RN;" ";"RECORDS"
920  VTAB 24: PRINT "TYPE A KEY FOR ";SE + 1;"KPA CO2 ";:
     GET A$: RETURN
1000  REM ****************************************
1001  REM * CORRELATION CO-EFFICIENT CALCULATION *
1002  REM ****************************************
1005  IF RN = 0 THEN  RETURN
1010  XS = 0:XQ = 0:XY = 0:YS = 0:YQ = 0
1020  FOR I = 1 TO RN
```

```
1030  XS = XS + X(I,0)
1040  YS = YS + X(I,1)
1050  XY = XY + X(I,0) * X(I,1)
1060  XQ = XQ + X(I,0) * X(I,0)
1070  YQ = YQ + X(I,1) * X(I,1)
1080  NEXT I
1090  XM = XS / RN:YM = YS / RN
1100  RV = (XY / RN - XM * YM) /  SQR ((XQ / RN - XM * XM)
      * (YQ / RN - YM * YM))
1110  RV =  INT (RV * 1E2 + .5) / 1E2
1120  HOME : VTAB 24: PRINT "CC=";RV,"RN=";RN
1140  RETURN : REM * RV=COR COEF *
1997  REM ***********************
1998  REM * REGRESSION EQUATION *
1999  REM ***********************
2000  XS = 0:XQ = 0:XY = 0:YS = 0
2005  IF RN = 0 THEN  RETURN
2010  FOR I = 1 TO RN
2020  XS = XS + X(I,0)
2030  YS = YS + X(I,1)
2040  XY = XY + X(I,0) * X(I,1)
2050  XQ = XQ + X(I,0) * X(I,0)
2060  NEXT I
2070  XM = XS / RN:YM = YS / RN
2080  C1 = (XY - RN * XM * YM) / (XQ - RN * XM * XM)
2090  C2 = YM - C1 * XM
2100  C1 =  INT (C1 * 1E2 + .5) / 1E2
2110  C2 =  INT (C2 * 1E2 + .5) / 1E2
2170  RETURN : REM * MX=C1 C=C2 (Y=MX+C) *
```

```
                *************************
                * CO2+P50+P100CLOCK.OBJ0 *
                *************************

      PROGRAM FOR MOUNTAIN HARDWARE CLOCK/CALENDAR CARD

            ORG $300        START AT DECIMAL 768
            JSR TIME1       GET TIME AT END OF INSPIRATION
            RTS

            LDA #$00        SET LOCATION FOR STORAGE OF
            STA $6C9F       END-TIDAL CO2 TO ZERO
INCHECK     LDY #$00        SELECT CHAN 0 ON ADC (-VE PRES)
            STY $C401       BEGIN CONVERSION
            PHA             DELAY
            PLA             LOOP
BIT1        BIT $C402
            BPL BIT1        WAIT FOR EOC
            LDA $C400       OBTAIN CONVERTED DATA
            CMP $6CA0       COMPARE WITH THRESHOLD IN 27808
            BCS INSPN       IF LESS, INSPIRN HAS STARTED
            LDY #$03        SELECT CO2 CHAN IF STILL EXPIRN
            STY $C401
            PHA
            PLA
BIT2        BIT $C402
            BPL BIT2
            LDA $C400
            CMP $6C9F       COMPARE WITH PREVIOUS MAXIMUM CO2
            BCS STORECO2    AND STORE IF GREATER
            JMP INCHECK

STORECO2    LDA $C400
            STA $6C9F       STORE CO2 MAX
            JMP INCHECK
```

```
INSPN        JSR TIME2        GET TIME AT END OF EXPIRATION
             LDA $COB6        STOP CLOCK AND ZERO MSEC TIME BITS
             LDA $COB5        START CLOCK
             JSR TIMEO        GET START TIME OF INSPIRATION
             LDA $C058        SEND SIGNAL TO OPEN SHUTTER
                              (AN 0 = PIN 15 GAMES PORT)
READ50MS     JSR READCLOCK
             CMP #$50         COMPARE TIME WITH 50 MSEC
             BCS P50          GET -VE PRES SINCE 50 MSEC ELAPSED
             JMP READ50MS     50 MSEC NOT YET ELAPSED

P50          JSR READPRES
             STA $6CA1        STORE P50 IN 27809

READ100MS    JSR READCLOCK
             CMP #$00         COMPARE WITH 00 (100 MSEC)
             BEQ P100         GET -VE PRES SINCE 100 MSEC ELAPSED
             JMP READ100MS    100 MSEC NOT YET ELAPSED

P100         JSR READPRES
             STA $6C9E        STORE P100 IN 27806
             RTS              RETURN TO BASIC

READCLOCK    LDA $COB4        READ 1-10 MSEC TIME BITS (BCD)
 RTS                          RETURN FROM READING CLOCK

READPRES     LDY #$00         SELECT -VE PRES CHAN
             STY $C401
             PHA
             PLA
BIT3         BIT $C402
             BPL BIT3
             LDA $C400
             RTS              RETURN FROM READING -VE PRES
```

```
TIME0       LDX #$00        TIMING ROUTINES
            JMP CLOCK
TIME1       LDX #$06
            JMP CLOCK
TIME2       LDX #$0C

CLOCK       LDA $45         SAVE A REG
            PHA
            TXA             SAVE X REG
            PHA
            TYA             SAVE Y REG
            PHA
            LDA $38
            PHA
            LDA $39
            PHA
            LDA #$C3        SET I/O FOR CLOCK IN SLOT 3
            STA $39
            JSR $C300
            PLA             RESTORE $39
            STA $39
            PLA             RESTORE $38
            STA $38
            PLA             RESTORE Y
            TAY
            PLA             RESTORE X
            TAX
            PLA             RESTORE A
            LDY #$05

LOADCLK     LDA $282,Y
            STA $6C84,X     STORE CLOCK TIMES
            INX
            DEY
            BPL LOADCLK
            RTS             RETURN FROM CLOCK
```

```
********************
* CO2+P50+P100.OBJ0 *
********************

   PROGRAM FOR STRATHAND TIMER CARD


            ORG $300        START AT DECIMAL 768
            LDA $C0B2       START TIMER
            LDA $C0B4       RESET TIMER TO ZERO
            RTS

            LDA $C0B3       STOP TIMER
            LDA $C0B0       LOAD LOW BYTE OF INSPIR TIME
            STA $6C9A       AND STORE IN 27802
            LDA $C0B1       LOAD HIGH BYTE AND
            STA $6C9B       STORE IN 27803
            LDA $C0B2       RE-START TIMER
            RTS

            LDA #$00        SET LOCATION FOR STORAGE OF
            STA $6C9F       END-TIDAL CO2 TO ZERO
            STA $6C9F

INCHECK     LDY #$00        SELECT CHAN 0 ON ADC (-VE PRES)
            STY $C401       BEGIN CONVERSION
            PHA             DELAY
            PLA             LOOP
BIT1        BIT $C402
            BPL BIT1        WAIT FOR EOC
            LDA $C400       OBTAIN CONVERTED DATA
            CMP $6CA0       COMPARE WITH THRESHOLD IN 27808
            BCS INSPN       IF GREATER, INSPIRN HAS STARTED
```

```
            LDY #$03         SELECT CO2 CHAN IF STILL EXPIRN
            STY $C401
            PHA
            PLA
BIT2        BIT $C402
            BPL BIT2
            LDA $C400
            CMP $6C9F        COMPARE WITH PREVIOUS MAXIMUM CO2
            BCS STORECO2     AND STORE IF GREATER
            JMP INCHECK

STORECO2    LDA $C400
            STA $6C9F        STORE CO2 MAX
            JMP INCHECK

INSPN       LDA $COB3        STOP TIMER
            LDA $COBO        OBTAIN LOW BYTE OF EXPIR TIME
            STA $6C9C        AND STORE IN 27804
            LDA $COB1        OBTAIN HIGH BYTE AND
            STA $6C9D        STORE IN 27805
            LDA $COB4        RESET TIMER
            LDA $COB2        START TIMER
            LDA $C058        SEND SIGNAL TO OPEN SHUTTER
                             (AN 0 = PIN 15 GAMES PORT)
READ50MS    JSR READCLK
            CMP #$32         COMPARE TIME WITH 50 MSEC
            BCS P50          GET -VE PRES SINCE 50 MSEC ELAPSED
            JMP READ50MS     50 MSEC NOT YET ELAPSED

P50         JSR READPRES
            STA $6CA1        STORE P50 IN 27809

READ100MS   JSR READCLK
            CMP #$64         COMPARE TIME WITH 100 MSEC
            BCS P100         GET -VE PRES SINCE 100 MSEC ELAPSED
            JMP READ100MS    100 MSEC NOT YET ELAPSED
```

```
P100        JSR READPRES
            STA $6C9E       STORE P100 IN 27806
            RTS             RETURN TO BASIC

READCLK     LDA $C0B0       READ TIMER
            RTS

READPRES    LDY #$00        SELECT -VE PRES CHAN
            STY $C401
            PHA
            PLA
BIT3        BIT $C402
            BPL BIT3
            LDA $C400
            RTS              RETURN FROM READING -VE PRES
```

GLASGOW UNIVERSITY LIBRARY