DEVELOPING MENTAL MODELS: A CHILD'S CONCEPTION OF A COMPUTER SYSTEM BETWEEN THE AGES OF NINE AND FOURTEEN

PEARL PATRICIA DENHAM, B.Ed(Hons).

Thesis submitted for the degree of Doctor of Philosophy, in the Department of Psychology, Faculty of Social Science, University of Glasgow.

DEDICATION

TO JOHN

TO ALL MY PUPILS PAST AND PRESENT

ACKNOWLEDGEMENTS

I would like to express my thanks to a number of individuals without whose help and encouragement this thesis might never have been undertaken or completed.

First and foremost, I wish to thank my supervisor, Professor Simon Garrod for his advice, support and extensive help in all aspects of this thesis.

Second, I wish to thank the Schools which have taken part in these studies and their Headmasters,
D.S.C. Arthur M.A, Lomond School, Helensburgh, Scotland.
M.J.KEMP, M.A. City of London Freemen's School, Ashtead Surrey until July,1987
D.C.HAYWOOD, M.A. City of London Freemen's School

To members of staff, my colleagues, who gave their total support with the Mental Models drawing studies:

Elizabeth MacIntyre, Lomond School
Jane Emburey, Frances Stevenson and Janet Haslum,
City of London Freemen's School.

To all my Information Technology and Computer Studies pupils, past and present without whom this Thesis would not have been possible.

To John, my grateful thanks for his patience, encouragement, practical help and for 'being there' throughout.

CONTENTS

SECTION ONE

CONTENTS

Page

SECTION ONE

Chapter 1

The Educational Motivation

Chapter 2

Intellectual Development and Mental Models

Chapter 3

Mental Models, Theory and Practice

ABSTRACT

Developing Mental Models: A child's conception of a computer system between ages 9 and 14 years.

The purpose of this research was to determine whether a child novice-user of a computer system possessed a mental model of that system. If such evidence existed, to establish its nature, causally or terminologically related to the model.

A necessary element of the research was to devise a method of data capture appropriate to the nature of the data to be collected and to the youth of the population. Four studies established the effectiveness of subject drawings as a method of data collection.

Initial studies on the internal appearance of the computer identified a working model with a common set of five components. Three of them were fundamental functions; Communication-links, Memory and Input/output. More of the common set of five components are found to be included in the model as age increases, although the 'preferred' or ranked order of component choice was reliably maintained across three age-bands.

Communication, a pre-requisite for an interactive system, was perceived by virtually all the subjects. 'Communication-links' were identified in three modes. The most important mode, reflecting component inter-dependence, was identified in the drawings of a substantial and increasing number of subjects across the three year age-range. Whilst it was shown that a significant variation of choice pattern existed within each age-band, age itself was a significant influence on mode choice.

A subsequent study of 122 subjects, to identify modes of data transmission between keyboard and visual display unit, revealed four forms. The most advanced mode proposed a conduit together with evidence of data transmission using keyboard characters, electrical current and/or impulses.

It was found that as the subject became older he is more likely to explain effectively the hidden system processes between input and output by including an intermediate step. Furthermore, the intermediate step is more likely to be a memory component and the subject is more likely to illustrate the explanation with visible transmission of data.

The conceptual model of memory was found to be weak with no sense of its fundamental role in accommodating system and user needs of data retrieval. Throughout the study memory was generally recognised by the novice only when the use was explicit, implying that it is 'empty' most of the time and, because it is perceived as a formless space, no structures existed for subsequent data retrieval. In both cases it was shown that the perception could be modified by experience.

Recognition of memory as an indespensible component was found to be age and/or experience related and, at post-test only, that a relationship existed between year-band and the model of memory storage adopted. There were also reliable indications that different age groups had different beliefs about how data was stored in memory, particularly in terms of addressed memory locations. There were also indications that intermediate experience modified the subject responses positively and that the combinatorial effect of age and interim experience had a positive effect on the results. Finally, a relationship between subject age and the use of equations when allocating memory space to arithmetic results was found.

Memory, as the hidden process, was contrasted with an observable process, the visual display unit. Here it was found that experience and the combined effects of experience and age-group had a positive effect on whether subjects believed that visual display screens were 'blank' during processing, that is, blank screens were predicted less often.

Perceptions of the hidden process between keyboard and visual display unit fell into three categories. One group included an intermediate step using 'memory', sorting/ordering and 'others', the incidence increasing between years 1 and 3.

It was predicted that the model type was not unrelated to the year of the subject and also, that where an intermediate process was shown, the choice of process model was related to the yearband.

Language, is believed to play a role in mental model development and when early work with subjects revealed confusion between the everyday and esoteric roles of computer terminology, it was explored through several studies.

When asked to classify a list of computer related items according to their perceived system role no group observation was as expected, whether the items were given as individual words or as a sentence. It was also shown that subjects from the different age-bands acted similarly in the way they classified the task roles of the items with clear direction to the 'Command' category. However, the single Memory category revealed that the average number of allocations to the memory category is not unaffected by the year group of the subject. The results generally indicated a growing awareness of system memory requirements.

The subjects were also asked to classify computer related words as 'picture', 'word' or 'nothing' evoking and the rankings across the age-range 1st to L4 for 181 subjects, were found to be not unrelated.

Finally, a study examined whether the novice possessed an adequate and helpful model which would enable the correct analysis and predicted outcome for a 'bugged' computer program. A high percentage of the subjects anticipated a correctly running program and the predicted program effect for the three age-groups was not dissimilar.

Conclusions

Drawings by research subjects are a valuable means of collecting research data whilst avoiding the transmission of directional cues; the method provides a wealth of new data and further insight into naive perception.

The research demonstrated a usable mental model of the computer system modified with age and experience. Its nature, in several fundamental aspects, formed the basis of a conceptual model and was seen to undergo changes, some of which were cause and effect related. However, computer-related terminology is seen to play an important part in many of the misconceptions held about the computer. It was clearly demonstrated by the three main studies that there is no significant age-related increase in understanding about the 'vocabulary-system model' relationship.

The recognition of a mental model and the development of appropriate constraints lay the foundations for future, specific model research as well as providing a valid contribution to future teaching methods.

DEVELOPING MENTAL MODELS:
A Child's Conception of a Computer System Between the Ages of Nine and Fourteen

CHAPTER 1 :EDUCATIONAL MOTIVATION

"The ultimate value of research whatever its form lies in the production of information that enables people to change".

Anon.

Introduction

The motivation for this study stemmed directly from my ten years as a practising teacher and close involvement with novice computer users. During this time a broad two-step transition has taken place which has changed the perceived role and implementation of computers within education.

Initially, the computer was seen in the narrow context of a 'specialist' subject which was offered at public examination, for example, at the General Certificate of Education levels 'O' and 'A'. The first advancement to blur the division between specialist and non-specialist computer user was seen when the 16+, vocational courses were introduced for the less academic pupils, for example, the SCOTVEC qualification in Scotland. The second major change began with the introduction of courses for younger non-examination pupils. These were largely under the heading of Information Technology, and required greater flexibility and the provision of a much wider variety and level of skills, not least to accommodate students of wider ages and abilities.

During the earliest phase, computers were generally to be found in the mathematics department and computing candidates were often the brightest in that subject. Such pupils were well able to accommodate the theoretical, machine orientated approach and whilst misconceptions were observed the effect tended to be minor. With the transition from specialist to non-specialist 16+ users particular areas of learning difficulty became apparent which often created persistent barriers to success in the subject.

The second broad phase of transitional change has been developing for five or six years and is not yet complete. Today, the use of computers as a resource is being made available to most of our children, starting in the primary school. The Educational Reform Act(1988) is beginning to effect changes for the advancement of Information Technology and the use of computers through the Technology component, at present via non-statutory guidance. Within this structure suggestions are offered to teachers and IT coordinators on ways of building and implementing Information Technology across the curriculum.

The guidelines appear, however, biassed toward structured 'application' of the resource tool, at the expense of knowledge and understanding of the tool itself, a pre-requisite underpinning its full and sensible use. This thesis set out to examine how the novice can be encouraged to develop a better understanding of the computer as a processing system and thereby effectively increase its efficiency as a tool. The thesis is written in three sections which reflect a natural pattern of development.

Section one lays the foundations for the empirical studies and is in three chapters. The task of this, the first chapter, is to present a statement of the problem, that is, an examination of the educational motivation. It introduces the varied influences and constraints at work in the environment surrounding the novice. Chapter two examines the novice and the constraints which are implicit in his intellectual development. In particular, clues about the young novice's ability to consider processes he is unable to witness and his ability to hypothesise about cause and effect relationships are sought. The concepts of imaging and mental models are examined in a problem-solving context and, in chapter three, examples of relevant research carried out on the theory and practice of mental models is examined.

The second section of the thesis progresses to a series of practical studies undertaken with young computer users. The mental models held of the computer system are interpreted from novice drawings and analysed according to their components and structure. A further dimension is added to selected components by a study of predicted system changes during processing. Finally, novice interpretation of computer-related terminology is examined as a possible factor in misconceptions about the system affecting perceived structure and, thereby, choice of model component.

The objective of the concluding section, is to bring together the research findings in an outline of implications affecting the design of a system model. A general model is proposed from which a schematic task-related, first-level model is suggested which illustrates the major conclusions about the 'user' model. The final part of the thesis proposes and discusses a set of educational constraints based on the findings and the problem stated in the opening chapter.

The Background

As Head of a Computer/Information Technology department, I hold the responsibility for development and implementation of these areas in our Junior and Senior schools, as well as for providing a support service to subject teachers. The department is well equipped with two computer laboratories which I designed and developed. There are two networks, an Econet /SJ fileserver managed system in the Junior school running a mixture of 21 BBC(B) and MASTER 128 machines, whilst the Senior school network runs 22 Apple MAC+ computers using Appleshare backed by two hard disks. There are also laser and dot-matrix printers, an Apple Mac II computer used for Desk Top Publishing, communications and document scanning facilities. To contrast the user-friendly 'Apple' operating environment, further equipment includes an IBM PS/2 model 50 and an Archimedes offering students alternative operating system experience. In addition, almost all subject departments in the school have at least one computer as part of the developing Information Technology implementation.

Most secondary and many junior schools have been caught up in the tidal rush of micro-computer development which began about ten years ago and has been gaining momentum ever since. Money has been, and still is, poured into schools all over the country and my own school illustrates the current trend.

Much talk is heard now about faster processors, 16 or 32 bit machines and IBM compatibility. Anxiety is voiced about the lack of qualified teaching and advisory personnel, usually in the context of 'knowing how to use' the new machines and their growing complexities. The new technology demands improved methods of teaching and learning and these issues are clearly not being addressed. No other subject in the history of education in schools matches the capital invested in computer installation, maintainance and upgrading but we are failing to conduct the necessary research into learning and teaching that has taken place elsewhere. A fundamental framework of understanding about the child/computer learning relationship is an essential on which plans for the future can be based. The integrity of any such studies must be rigorously maintained because of the rate and nature of technological change.

A critical point in the history of human education has been reached. We have seen a fundamental change in the way people learn; this change is based on modern computer technology. Early in the eighteenth century, as a direct result of the printing press technology, but some 250 years after the invention of the printing press in Western Europe, textbooks appeared in schools. The first two digital computers were built in 1939, separately, one by Stibitz and the other by Zuse; the changes that have occurred in less than 50 years have been phenomenal. The current change, due to advances in micro-electronics, is yet more dramatic in its implications and repercussions. The reasons are based on the fact that the world population today is much greater, everything happens at an ever increasing rate and, because of the rapidly increasing pace of life, changes spread more rapidly. Finally, everything we do, including the way we learn, is driven by powerful global commercial interests.

Changes in society effect changes in education which in turn effect changes on society as a result of education; the two are inextricably bound. Whilst computers have the potential for improving education their misuse produces poor or negative results. Likewise, our curriculum development cannot be left to market forces or the whims of fashion. The question posed is, not whether the changes will take place but whether they will be beneficial or detrimental; do we choose to use the computer with better understanding, imagination and determination?

The situation now

Computers already play an important role in our lives and environment. Recognition of their devoloping role and a growing educational need have been reflected in such government activities as:

1. Department of Trade and Industry schemes 1980-88: Introducing Micros in schools, Micros in Primary Schools, Peripherals Schemes, Supported Software and Telecommunications Schemes.
2. D.E.S. Micro-electronics Education and Support Schemes
3. Development of local support and advisory centres
4. The training of new subject teachers and in-service courses.
5. The National Curriculum: ERA

The all too obvious weakness in provision for points 3 and 4 , that is, local support/advisory centres and the training of teachers to use hardware/software is already being vigorously promoted by competitive hardware suppliers. Whilst their attention is welcomed they represent a vested interest and do not necessarily support the important pedagogical needs and interests. Their interests are for today whereas the interests of education must also encompass society's future needs.

The thrust of modern computer development is focused on the personal computer and it is the personal computer which is most effective in education. Fortunately, good progress has been made by the emancipation of school computers from their former more elementary 'number-crunching' niche within the mathematics department and its more appropriate placement in a 'service' role promoting Information Technology as well as Computer Studies.

With the change of subject emphasis (assuming the majority of pupils receive time-tabled group sessions under 'Computer Studies' or 'Information Technology') have come many new pedagogical and management considerations. They include:

a. Group disparity:

   Most classes are mixed ability and pupils probably bring a greater range of subject knowledge, experience, attitude and interest than to any other subject of the curriculum.

b. Teaching staff:

The profession is faced with the sizeable problem of training our teachers to understand and exploit the capabilities of the computer in education. Few teachers are close to good in-service programs and, where they are, only a fraction can be involved, given the resources. Most schools have or are developing networks of computers for the main purpose of sharing valuable resources such as printers, disk-drives and software. Consequently, it is no longer sufficient for teaching staff to be familiar with the use of a computer as a stand-alone; networking has introduced a far more sophisticated and elaborate teaching and learning system.

c. Non-teaching time:

Learning to use the computer is a practical activity requiring careful development and preparation of software and other teaching materials. At present there is a deplorable lack of standards of quality and most teachers and administrators are unable to distinguish good computer based learning material from poor. Efficient and appropriate criteria referenced monitoring systems need to be developed for each target group . In addition, machines need maintenance, networks require management and personal/commercial software must be developed, tested and documented before use. Finally, the time and ability to offer advice, technical help, support and training to other members of teaching staff is assumed.

d. Management attitudes and constraints can affect :

i. the year groups offered the subject as part of their general education.

ii. the number of pupils allocated to each class or group for general education and the time/frequency of the sessions.

iii. whether or not the subject is offered at external examinations in competition with the traditional subjects on the timetable.

iv. the calibre of pupils taking computer studies as an examination option.

v. the hardware and software, as well as texts available.

vi. subject isolation - the degree to which the computer is integrated across the curriculum.

It is not unusual, as the subject gains popularity, to find that all pre-option pupils, i.e. under 15 years of age, have computer studies on their time-table either on a weekly or rota basis. A first year, secondary school pupil may expect to receive four or more years hands-on computer experience and, as the majority are unlikely to take the subject further, it is here that the greatest responsibility lies, the responsibility of developing computer literacy.

After teaching Computer Appreciation or Computer Studies to a large number of students, school pupils, ranging from 8-18 years, as well as parents and teachers, patterns of behaviour have been observed which indicate quite clearly that the attainment of 'Computer Literacy ' could and should be more efficient, more effective and less stressful.

Whilst examining the aims and objectives behind such introductions to the computer I doubted the validity of the interpretation as well as use of the current buzz-word 'computer literacy' . Before any developmental process can be planned it is clearly necessary to define accurately the objective itself.

Computer Literacy the Goal of Teaching

Chenge and Stevens (1985) carried out a study designed to establish a working definition for school students in Nebraska. Their searches of available literature also showed that no common opinion existed. According to Gress[1] the objective of computer literacy is to teach students about computers rather than teaching programming skills or providing computer based education. Conversely, Hunter, Seidel and Murphy[2] identified programming skills as an indispensible element of computer literacy in their respective definitions. Their concluding definition of 'Computer Literacy was "whatever a person needs to know about computers and, be able to do with computers in order to function completely in society".

In an effort to solve the problem Cheng and Stevens(1985) used 10 computer literacy topics which had appeared most frequently in their literary search. Categorised under two headings a questionnaire was administered to 95 randomly selected Nebraska educators on an instructional computing course. They were:

AWARENESS ABOUT COMPUTERS

1. the history of computers
2. computer terminology
3. parts of computers
4. having positive attitudes towards computers
5. understanding the impact of computers on society

6. being able to run a computer which includes running computer software
7. having basic Mathematical concepts
8. being able to write algorithms or draw flowcharts when using computers to solve problems.
9. knowing programming language commands
10. being able to write computer programs.

Data was analysed and represented as a linear transformation of the ten topics obtained by plotting the adjusted z-scores on a continuum individually. This scale not only reveals the rank order of the importance of computer literacy topics but also provides meaningful interval distances among topics(see chart(1) Linear Transformation). To run a computer efficiently is regarded as important. It implies understanding and knowledge which returns the enquiry to its starting point - the need to identify novice-related learning difficulties which appear to frustrate full achievement for many pupils.

Christopher Reed(1985) opened his article on the work carried out by Steven Pulas[3] with the words "You can lead a child to a micro but you cannot make him understand it!". Nor, as Simpson(1978) says, do children all have the same attitudes toward computers, especially when they discover that computers do more than just provide games. In fact, the study led Pulas to argue that traditional classroom arrangements for computer teaching may be alienating children at an early age. His team examined 140 average children between 8-13 years, selected from state

Chart(1) Cheng & Stevens(1985)

Linear Transformation of Topics

1.93	Run a computer
1.63	Terminology
1.49	Language commands
1.36	Positive Attitudes
1.16	Inpact of computers
1.08	Write programs
0.88	Parts of computers
0.75	Write algorithms
0.62	Math concepts

schools, with intensive computer curricula, fourteen of each sex from each grade. The pupils' experience included weekly visits to the computer laboratory, computer assisted instruction, computer literacy training and some experience with word processing and programming. The pupils also had easy access to school computers during breaks, lunch-time and after school hours. He found that, despite nearly 4 years experience, almost half the older children had false ideas about how the computer worked and, together with younger children, they believed that computers were more intelligent than people, machines simply stored information and that discs were no different from programs.

Simpson R.D.(1978) carried out an exploratory investigation to examine the general attitudes of middle school students toward computers. He used the GATC scale[4]: ten items becoming scores on a Likert response scale ranging from 'strongly agree' (5) to 'strongly disagree' (1). The subjects were an entire middle-school population, resulting in 512 usable surveys which examined the general attitude differences between 'boys and girls', 'computer at home and no computer at home' groups. Three research questions were posed:

1. Were there any differences between the general attitudes of girls and boys towards computers?
2. Were there any differences between pupils of the 7th and 8th grades?
3. Were there any differences between those who had home computers and those that did not?

In order to answer these questions the data were subjected to the traditional t-test. No significant difference was found regarding gender or grade level. However, a significantly more positive general attitude was found to be present with students who had home computers. It was admitted that these pupils could have been influenced by their environment which was described as 'close to high technology', Table(1).

Table(1) SIMPSON: TABLE OF RESULTS

GROUP	N	X(bar)	T-VALUE	DF	p
girls	259	38.99	0 .11	510	<0.911
boys	253	39.06			
grade 7	284	39.39	1.37	510	<0.172
grade 8	228	38.57			
home comp.	94	41.02	31.18	510	<0.001
no comp.	418	38.58			

Similarly, Moore(1985) studied pupils' attitudes to computers and robots. He used a 64 item Likert type instrument to assess seven scales of attitudes: pupil sex, Piagetian stage, chosen course of study, home use of micro-computer and experience of CAL[5]. In this case the results showed more favourable attitudes were held by boys (consistent with the Newbould survey 1982 and Gardner 1975) and by pupils choosing Computer Studies. The disparity between the two sex-related studies serves to highlight the point made earlier that, due to the enormous changes that have occurred in a very short timescale, conditions which produced many such study results become out of date and no longer reflect the present situation.

Amongst pupils not choosing computer studies, Moore found that those reporting experience of CAL or high use of home micro-computers showed smaller declines of attitudes over a 1 year period (14+ to 15+). The results showing absence of a strong relationship between Piagetian stage and attitudes agree with Lavine's findings (1965) of only small correlations between general ability and a wide range of affective criteria.

Gardner, McEwen and Curry(1985) reported growing disquiet among educationalists that differences between boys and girls in choices of mathematics and science subjects were being echoed in the use of computers.They concluded that girls perceive computer studies as having a masculine image both in its use and presentation. It appears also that as a consequence of working more closely with boys in co-educational schools girls perceive computing in a more narrowly mathematical context and also see themselves as less capable than boys with respect to computer studies.

A different view on computer-related attitudes has been picked up by Diem(1955) in his study of children's attitudes and reactions to the New Technology. He says that society now demands that schools provide instruction in the uses of technology. Social and citizenship skills, especially those that emphasize the use and abuse of information and ethical standards must be part of any successful integration of technology. He discusses the operation of double standards where participants understand that information has an ownership quality such that when one knows who has produced the information on a personal basis, that quality is to be upheld and protected. However, once the ownership became dispersed and commercialized, these qualities lost their significance and, in fact, held little meaning to those using the information.

Also, as any computer network manager will confirm, students learn very quickly how to abuse information systems. The indentification of data access codes used by fellow pupils is common and they are used to write derogatory programs that appear when the owner logs in on returning to his machine. Likewise, pupils do not generally question the right to look at another's information. It is as though all information were open and accessible to anyone who wished to view it. These points have topical and serious implications beyond school life.

It is probable that the learning environment in which pupils' micro-computer activity takes place also plays a decisive role in attitude formation. Direct evidence for the effect of the learning environment on computer-related attitudes comes from a study by Cavin et al(1981) of college students using CAL materials in a chemistry course. It was concluded that the measured shifts to more favourable attitudes shown by some students were due not to the direct experience of using a computer, but to the interactive, non-threatening and easily used materials provided on the experimental course. The importance of the learning environment is also supported by findings of Fraser(1977). He showed pupils' attitudes to the social implications of science were enhanced by an environment characterised by a less hurried approach, more goal direction and more pupil satisfaction. These environmental conditions are present in self-chosen home learning and recreational activities and to a lesser extent they are also present in many instances of pupil controlled CAL.

Loyd and Gressard(1984) introduce a further attitudinal dimension, also frequently observed in the classroom by the writer, it is that for some students computer experience is extremely unpleasant. These students exhibit varying degrees of anxiety when required to learn about or use computers, and therefore have more than usual difficulty in mastering computer skills. Jay(1981) describes these negative attitudes as computer anxiety or "computer- phobia". Computer anxiety may take the form of resistance to thinking about computer technology, fear of computers, and even hostile or aggressive thoughts towards computers. Attitudes such as these inevitably inhibit successful mastery of computer skills in much the same way that anxiety about mathematics (Fennema & Sherman 1978) inhibits achievement in mathematics.

Information gleaned by an adult observer can be particularly helpful when studying affective attitudes and experience shows that young children are generally less inhibited about answering questions, for example, than their senior school counterparts. Whilst their questions can often appear trivial they offer a valuable source of clues as to how the child feels about the task he or she is doing. The study carried out by Shaw,Swigger&Herndon(1985), in a Texan elementary school, reversed the normal situation where adults ask children questions and instead studied the questions asked by the children. The results show that concerns fell largely within the following categories:

Operating System
"What am I supposed to do?" (Load and Run commands required to be entered)
Keyboard
"Where is the comma?"
"Where are the quotation marks?"
"Is this an L?"(child points to I)
Wait-time
"Well, what do I do now?" "Do I press RETURN ?"(computer was loading a program)
Data Entry
"What do I do now?" (child needs to press RETURN)
"Do I need to press RETURN?"
Program Content
"What does it do when you get the wrong answer?"
"What do I have to do with the program?"
"What does that mean?" (referring to correct answer feed-back)
Program Instructions
"What is that word?"
"What am I supposed to do?" (message on screen is 'Type in your first name')
"Do I type yes?" (message on screen is ' Are you ready')
Screen characters
"Is that an 8?"(child pointing to zero)
"What is that letter?" (child pointing to N,M orH)
Miscellaneous
"Can we play games?"
"How much did this computer cost?"
"How much does this TV weigh?"
"When can I do this again?"
"Who discovered computers?"

The questions categorised by Shaw et al reflect curiosity, lack of experience, and the strong need for re-assurance, all understandable and, seemingly, universal.

The introduction and implementation of a wider ranging use of computers in schools, encouraged by the Information Technology guidelines for the National Curriculum, introduce serious implications for all teachers.

As important as learner attitudes are those carried by the teacher. Yuen(1985) carried out a study which sought to determine the attitudes that Trade and Industrial teachers hold toward the use of micro-computers. It also examined factors that might be associated with these attitudes. The majority of trade and industrial teachers favoured the use of micro-computers in industrial education. However, there appeared to be significant differences in teachers' attitudes toward using micro-computers and (a) the teachers' micro-computer experience and training (b) utilisation of micro-computers and (c) the availability of micro-computers.

One of the main advantages of the computer as a teaching/learning medium is that it is interactive. One consequence of interaction is that the level of interest of the learner can be determined. As a result materials that are weak in interest can be modified and a different approach followed.

Although the computer allows a highly interactive approach with many benefits, which might include detailed views of pupil progress, not all computer based learning material is interactive. We need to develop standards for judging the extent and quality of interaction. Often beginners, both students and their teachers, are satisfied with very weak forms, because they are seen as an improvement over their previous experience of non-interactive learning media .

Rushinek A, Rushinek S and Stutz J(1984) observe "Frequently, end-users do not participate or have sufficient input in decisions concerning the acquisition, implementation, maintenance, and modification of interactive software. As a result, vast economic resources are wasted developing software which becomes under-utilized due to user dissatisfaction with systems which fail to meet their true needs. In summary they conclude that their study .... " shows that, as expected, users' performance is closely related to their attitudes toward the software".

Whilst all the foregoing factors are important this research shows that they are only secondary when seen against the background of the widespread misconception as to what the computer is, what it does and , most importantly, its limitations. This misconception has short and long term consequences.

Possible short and long term effects are revealed in:

(a) An apparent lack of interest in, or evasion of, computer use; most readily observed when two or more pupils share the use of a computer. This behaviour is also recognised as a 'defence' ploy when a novice learner is faced with more dominant or experienced partners . The long term effect could be continued avoidance and /or a fundamental mistrust of computers.

(b) An inability to transfer learning to new situations together with a teacher dependency revealed in a need for constant reassurance and/ or repetition of basic instructions. Long and short term implications may involve the ability to develop logical problem-solving methods of working.

(c) An inability or unwillingness to deviate from learned procedures in order to experiment either with the computer hardware or software leading to self-limiting and ritualistic behaviour in the long and short term.

(d) Long and short term problems related to the fundamental concepts involving computer data structures, manipulation and storage. The weaknesses are manifested in the way in which application software is used by the novice and, by the type of problems generated by beginners in computer programming. For example, incorrect logical order of procedure sequences, the mis-use of storage variables and of loop structures.

Towards a Principled Understanding

There is a well defined and urgent need for a deeper, principled understanding of the relationship between learner and computer, of how the novice understands the system he or she is using.

The computer presents a particular problem to novice users because of the inability to 'see' what happens between cause (pressing a keyboard character) and effect (e.g.a menu option is selected). To the young novice this is a magical transformation as they find themselves in the popular adventure "Granny's Garden" or in the "Turtle Graphics" environment. Equally, if the software is unreliable, the pressing of the wrong key can cause an abrupt and unsatisfactory end to the session, thus compounding any existing feelings of inadequacy or insecurity.

Before a state can be improved or a problem solved analysis of the existing situation is necessary. Good analysis involving people pre-supposes the ability to communicate effectively. A way has to be found to recognise and record the structure of the existing understanding and the way it develops both over time and under the impact of experience. Given this 'window' into novice understanding of how computer systems operate and a means of 'communication' we shall be in a far better position to promote the effective acquisition and development of computer related concepts.

The Mental Model

The problem has already been recognised and is being tackled beyond the school gates. Borgman(1982), for example , describes the features of the Xerox STAR system as both powerful and easy to use, for one simple reason, it has been designed around the way people think about tasks and the computer. Most of the work at the time of Borgman's writing was in text-processing and pocket calculators, her application was to information retrieval systems. Xerox researchers have applied cognitive theories about how people comprehend and reason about complex mechanisms to the design of the system. The theory emerging from the research is that people build mental models of the computer system. The mental model is a representation of the relationships within the system, a person can RUN the model to produce a mental simulation of the system's actions. From the internal simulation ways of dealing with the system are developed.

Similarly, in the first of his five papers on "Human-machine Interaction" Norman(1982) stated that "Peoples' views of the world, of themselves, of their own capabilities and of the task that they are asked to perform, or topic they are asked to learn depend heavily on the conceptualisations that they bring to the task. That is, as a result of different forms of experience, the user develops mental models which provide predictive and explanatory power for understanding, for example, a computer system. In short, however, the mental models are unlikely to be technically accurate or elegant". They will contain the embedded beliefs and uncertainties of the user. If continued interaction with the system enables modification or development of the models according to experience then it should be possible to influence the nature of the change.

In conclusion, no existing form of assessment allows us to investigate, evaluate and modify the child's intuitive understanding of computers. A principled understanding of the problem must involve the total system, input and output as well as the 'hidden processes' occuring between the two and, it is my belief, that the key to a principled understanding lies in the study of mental models.

NOTES:

1 Gress in Cheng & Stevens(1985)

2 Hunter Seidel & Murphy in Cheng & Stevens(1985)

3 Pulos Stephen, Lecturer Berkeley Graduate School of Education California.

4 GATC: The General Attitude Towards Computers scale in (Reece & Gable 1982)

5 CAL: Computer Assisted Learning.

CHAPTER 2
Intellectual Development and Mental Models

Introduction

In this investigation, learning with understanding is an objective shared with Borgman(1982) and others but, unlike the subjects of their research, the novices considered in this thesis are children with rapidly changing learning circumstances and needs. Whilst it would be inappropriate here to tackle in depth the many facets and complexities of cognitive development, with its great diversity of experimental and theoretical approaches (Eysenck 1984), the introduction of a label such as 'mental model' demands some basis from which to evaluate its perceived relevance. With this point in mind, I draw upon two strands of development considered appropriate and necessary to the content and subjects of this thesis.

One section of the thesis examines mental models of the computer system and the way it works, as explained and studied through the drawings of subjects between the years 9 and 14. The first step in the production of a drawing, requires the child to imagine the internal appearance of a computer, an environment of which it is neither a part nor with which it is familiar. The second step requires the translation of the image to a paper 'reality' . The result, a representation of the drawer's mental model of the system, a composite of inherent beliefs, understanding and knowledge about the computer.

Mental models will be examined in greater depth in the following chapter together with some examples of their use. Here, however, I intend to combine two relevant elements of development. Firstly, to trace the origins and development of the mental model concept which is embedded within human development. Examining such points as what imagery is, whether it is a facility of which we can make use, and the characteristics of a mental model. Secondly, to establish the developmental constraints which may be carried by subjects as a result of their age. Together these points help to provide the basis for anticipating the novice subjects' conception of the computer system and what it does, including nature and constraints.

This thesis is concerned with the child's developing conception of a computer system through mental models therefore it is important to consider the key processes of intellectual development. Starting at birth, with no model of the world available, we begin to learn about ourselves and our environment. As a result the process causes constant adjustment and re-adjustment of behaviour and these modifications are based on two main sources of information, what is happening to us now and relevant past knowledge. This process of learning from cause and effect, is cognitive development and covers many modes of knowledge, perception, memory, imagination, conception, judgement and reasoning(Drever1975).

Cognition, is the activity of knowing: the acquisition, organisation and use of knowledge(Neisser1976). By studying cognition we can examine that which is similar in all human beings, cognition being the structures and processes which are used to take in information selectively from the environment. When we 'sense' external stimuli there is co-ordinated activity in several sensory systems but the way in which the information is interpreted constitutes' perception.

Perception, is a constructive process, and depends on the experience of the perceiver; on what he brings to the event. Gibson(1969) points out that the difference between a skilled and unskilled perceiver is not that the former adds anything to the stimulus but that he is able to gain more information from it. The point illustrates the complexity of perceptions that may be held by, not only a single age-band of novices but of multi age-band groups when looking at a computer or working with the same device and on the same problem. We are led to ask why this should be so.

For all of us our external world is an infinitely rich source of information. To make sense of and unscramble this constant and infinite variety of stimulus information implies withdrawal from some things in order to deal effectively with others, a skill described by James (1890) as 'selective processing'.

The ability to learn from our stimulus experiences depends upon pre-existing structures, called schemata[1]. The schema is that portion of the entire perceptual cycle, (Piaget's circular reaction[2]), which is internal to the perceiver, modifiable by experience and specific to what is being perceived,(Piaget's accommodation[3]).

It is Neisser's view that the schema is anticipatory: it prepares the perceiver to accept certain kinds of information rather than others thereby controlling action and assuring the continuity of perception over time. Piaget(1952) in particular has contributed much to this area of study and believes that because of the built-in properties of the schema, repeated assimilation leads not only to generalisation but also differentiation through refinement after repeated experience. The important message for the educationalist in Piaget's work has been that change is always gradual, the step between the new and the old must never be too large. He illustrates the point by describing the inability of the infant to accommodate to the ring as a member of the abstract class of circles, or as a 'wheel-like' object, or even as an item to wear as a bracelet. In a more complex form, during the early schooldays we cannot expect our young novices, on first introduction to the computer, to accommodate it as an electronic diary with which they could produce a graphical and textual report about their class activities. However, by the the end of their first year, for the same pupils to be producing short records of class activities is a recognised and achievable activity. On the other hand, the same pupils would not yet be able to accommodate the computer as a source of stored data from which selected information about themselves could be rapidly retrieved , and so on.

Eysenck(1984) notes that it is natural to regard perception as a processing activity of which the individual is consciously aware and this view has been shared by most psychologists interested in perception. He adds that most perception research relies heavily on the subject's report of what he or she is perceiving; this makes little sense unless there is conscious awareness of perceptual activity. This view does, however, ignore subliminal perception which has been effectively used in attitude and belief 'conditioning', as supported by Dixon (1981) who points out that the brain responds to external stimuli which, for one reason or another, are not consciously perceived.

Neisser believes that consciousness, an aspect of activity, undergoes changes throughout the course of life because we learn to pick up new types of information in new ways. These processes of change are called cognitive development in some contexts and perceptual learning in others.

As a result, it must be concluded that the way children in school, who are being introduced to computers at an ever earlier age, perceive a computer or the way it works will be largely influenced by his or her stage of development. Piaget, for example, sees development as a continuum rather than stages which are fixed by age;

i. Sensori-motor period from birth to about 2 years of age,
ii. Pre-operational thought roughly spanning the years between 2 and 7,
iii. Concrete operations typically occurring between the ages of 7 and 11 finally, but not in all cases,
iv. Formal operations.

If, for example, the Piagetian stages were related to the subjects of these studies we would expect them to, characteristically, be of stages three and four. That is, pre-dominantly Concrete operations but with some blurring of the line into Formal operations; we are reminded that not all adults reach formal operations. Piaget believes there are qualitative changes in thinking during the years of childhood and that the modes of thinking characteristic of adolescence are radically different from those of early childhood.

From the above descriptions we can anticipate, in general terms, the qualities and limitations of conceptions carried by the subject about the computer system. Until the age of 11 or so the logical operations used by the pupil will not be independent of the immediate situation. That is to say, the pupil's thinking is effective only to the extent that it is concerned with a particular concrete situation. The novice will not be able to free his or her thinking from the immediate situation and, as a result, the indications are that the child is not yet equipped to work efficiently with 'what if' problem situations related to the computer environment. He or she is unable to seek alternative ways of looking at a problem if the computer responds differently to expected, and most particularly, is not equipped to envisage complex, hidden processes and causal relationships such as found in a computer system .

In contrast, Formal operations, has two main aspects, firstly "that which underlies adolescent thought in all its myriad expressions" 'logico mathematical' properties, and substructures called 'formal operational schemas' which are more limited and specialised cognitive tools. Formal operation characteristics are orientated towards the possible and hypothetical. These lead to the tendency to explore all possibilities by subjecting the problem variables to a combinatorial analysis. This analysis gives him a cognitive picture of what possibilities exist which are then refined by observation and experiment toward the establishing of logical deductions about the causal structure of the system. However, it is noted that the Piagetian theory presents just one possible viewpoint and that other competing explanations may be more valid.

The idea of cognitive structures as vast, everchanging data-banks of information which are stored for the sole purpose of recall, either in the present or the future, is an interesting one. However, as the picture of intellectual development begins to take shape issues arise which are of particular concern to the practitioner. Chiefly, what sort of things are are changing and how can we get at them to learn more about them.

I believe that a key may be found in our long historical interest in, and awareness of, images. In daily life, children and adults have a tendancy to talk and think in pictures, particularly when trying to understand or explain a difficult concept. Therefore, it was decided to undertake practical experiments using pictures drawn by the novice as a basis for analysis. I believe that this method will produce an insight into the sort of understanding which exists about the computer system and the type of changes taking place in that understanding at differing stages of development.

Imagery is also included in the child's continuum of development and is thought, initially, to be very static and limited to the internal reproduction of perceptible states. However, Piaget, Flavell(1963) describes a later, higher level of imagery developing in tandem with the concrete operation child[4], therefore applicable to the subjects of my studies, and which produces a more mobile and flexible imaging activity capable of anticipating the successive movements of a yet-to-be transformation from one state to another. A factor which could have implications for component communications or data interaction within the system.

For my own research purposes, the concept of using these 'pictures in the mind' is an important one because I hope that the child's image of what he or she believes the computer system to be, whether static or predictive, will be translated onto paper.

The concept of memory-held images which model objects in the real world are central to this study, and leads us to question what memory images are. For example, what do we know about them and at what stage might they become a model capable of helping us to solve a complex problem such as why a predicted result is not being achieved by the computer.

Imagery and Cognition

The relation between perception and imagery is introspectively most obvious in the case of dreams which we identify as visual experiences sometimes quite difficult to distinguish from real life because of their vividness. In dreams and in waking images, the same kind of constructive process is assumed to occur with the difference that the representational processes are sometimes activated in the absence of 'adequate' stimulation. The process may thus be described as having 'autogenic' properties (Bruner1957), or as a learned skill in which 'central sending produces the conscious image in the absence of a different support' (Tomkins1962).

There have been three sorts of operational procedure which constitute the basis of most of the experimental research into the psychological importance of mental imagery (Paivio 1972,1975). Firstly, by instructing experimental subjects to employ mental imagery in carrying out psychological tasks. Although subjects are not instructed to generate images in order to draw their pictures in my own research , some autogenic imaging is assumed to place. I also believe the method described has a great deal to offer procedures designed to modify existing models. Secondly, the measurement of the extent to which the stimulus material is reported to produce mental imagery by the experimental subjects and the extent to which these reports predict variations in performance in psychological tasks with different samples of material. And, thirdly, the comparison of different experimental subjects in terms of their reports of the subjective vividness of experienced mental imagery and in terms of their performance tests of spatial manipulation ability. The second and third points are seen as suitable methods for developing powerful new teaching methods and materials.

Aristotle claimed that "thought is impossible without an image" and "memory, even the memory of concepts, does not take place without an image". Therefore, as the novice user begins to speculate about the device he is using(Payne1991), it is not unreasonable to assume that some imaging is taking place. Most philosophers assume that mental representations of objects must be in the form of images(visual, auditory, tactile etc), verbal meanings or both. Tichtener(1909), argued that even abstract concepts can be pictured "if you do but have the imaginal mind". I think the intuitive assumption of most people would be that their images are by nature 'visual' and it is not known at this stage how important it is to know whether that assumption is correct.

What is important, however, is that most of us recognise situations when mental images are being used to solve a problem. In his book 'Ghosts in the

Mind's Machine' Kosslyn(1983), for example, describes the visualisation of a room or house, followed by a scanning of the image to determine the number of windows, as does Paivio(1975) below:

> "Occasionally, when I have been required to list the names of my colleagues from memory, I have found myself visualising the hallways in which their offices are located, systematically moving past these offices then picturing and naming the occupants". Similarly, "If I am asked about the number of windows in my house, I find that I must first "picture' the house as viewed from different sides or from within different rooms and then count the windows presented in these various mental images ".

The important factor emerging is that images are accepted as a legitimate means of representing perceived real-world models which can be profitably employed in real-world problem solving. For the purposes of the computer novice it would be helpful also to know whether different forms of knowledge are more difficult to store and image than others; whether the mode of presentation is a contributing factor. It appears that the picture imaged in memory has to be interpreted by the brain and the actual form of this representation and the way it is later decoded is unclear. It is also unclear whether images of things in the world involve different kinds of representation to, for example, the verbal description of that external object, a point relevant to existing, highly verbal, methods of teaching the use of computers to the very young.

An influential theoretical approach that distinguished between verbal and imaginal processes was put forward by Paivio(1971) and is often referred to as the dual-code theory. Of the theory Paivio(1979) stated that -

> The theory assumes that cognitive behaviour is mediated by two independent but richly interconnected symbolic systems, which are specialised for encoding, organising, transforming, storing

and retrieving information. One, the image system, is specialized for dealing with perceptual information concerning non-verbal objects and events. The other, verbal system, is specialised for dealing with linguistic information.

Paivio also assumes that the imaginal system represents information in a spatially parallel manner, so that different components of a complex scene are available at the same time, whereas the verbal system is sequentially organised in the same way as linguistic utterances.

Paivios(1971) preferred explanation in terms of dual-code theory is shown in Table(2) below. He assumed that pictures are most likely to be stored in both codes(i.e. verbal and imaginal) whereas abstract words are typically stored only in the verbal code. Memory for an item will be more likely when two codes have been formed, because it is possible to retrieve that item on the basis of one code even if the other code has been 'lost' during the retential interval.

Table(2): Paivios Dual-coding Theory. The greater the number of plus signs, the greater is the availability of the appropriate code

Coding System Used Type of stimulus	Imaginal	Verbal
Picture	+++	++
Concrete word	+	+++
Abstract word	-	+++

The distinction between verbal and imaginal codes makes sense in the light of research into hemisphere differences that has suggested that verbal processing occurs primarily in the left hemisphere, whereas the right hemisphere is more involved with spatial and imaginal processing(Cohen 1977).

In addition, the claim that concrete words are more likely than abstract words to be processed imaginally receives some indirect support from the high correlation of +.83 that Paivio,Yuille and Madigan(1968) reported between word concreteness and imageability.

They note, however, that while concrete or imageable words are better remembered than abstract or non-imageable words, this does not demonstrate that retention is causally related to imageability. The reason is that the relationship between memory and imageability is merely correlational in nature.

Bower (1972) suggests that interactive imagery instructions and verbal mediation instructions are both effective simply because they increase the organisation and cohesion of the information to be remembered. Bower presented pairs of concrete words, and gave his subjects either interactive imagery(e.g. form images depicting objects interacting in some way) or separation imagery instructions (i.e.construct a mental image of two objects separated in space). On a subsequent cued recall test, the interactive imagery subjects performed much better than the separation imagery subjects, who in turn performed no better than subjects instructed to use rote repetition. In other words, interactive imagery instructions are effective because they enhance relation organisation rather than because they involve the use of imagery.

For the purposes of teaching concrete stage novices about the computer system and, for example, about the interdependent relationships of essential components, it can be assumed, in general terms, that processing which produces deep, elaborate and distinctive encodings will usually be effective in leading to successful long-term retention, and that the use of imagery especially interactive imagery that serves to integrate information from different sources, can only lead to enhanced learning and understanding.

Critics of imagery theory (e.g.Pylyshyn1973) have attacked the notion that the visual images stored in memory resemble mental photographs of pictures. Firstly, this picture metaphor implies that images are perceived in very much the same kind of way as pictures. Whereas Paivio (1971) argues that pictures require basic perceptual processing such as figure-ground segregation, but images are already highly organised into objects.

Secondly, if we forget part of an image, it is not a random fragment that is lost, as if a corner had been torn off the picture. Rather it should be compared with one of a box of randomly collected photographs which have been stored with no thought of retrieval. However, Richardson(1980) observed "... part of an image may be forgotten or may be incorrect when compared with the original. "The image theory of memory is shaken by finding that revealed images are not faithful reproductions of reality even for 'photographic' memories "(Fernald 1912).

At a more general level, Paivio believes there must be perceptual or visual codes to account for our perceptual abilities, and verbal codes to enable us to transmit and receive verbally encoded maessages. According to Pylyshyn(1973) the fact that we can readily translate or exchange information between these two codes(as when we describe a picture) indicates the necessity for postulating some kind of common format. The great structural differences between visual and verbal representation preclude a direct translation.

Pylyshyn(1973,1979) and others have argued that the form of representation is propositional. Propositions being abstract, language-like representations that assert facts about the world. They may be more adequate for representing information about the world than are images, because mental pictures do not assert anything, and are thus neither true nor false, (Eysenk 1984).[6]

Kosslyn(1980,1981) puts forward valid counter arguments. It is suggested that what most imagery theorists believe is that image representations resemble those that underlie the experience of seeing something, but in the case of imagery these representations are retrieved from long-term memory and do not stem from sensory stimulation. He says, in other words, images are similar to percepts, and thus images share the highly organised structure of most percepts.

When my subjects draw pictures representing their perceived view of a computer system based on experience, that available experience includes the overt behaviour of the machine, knowledge about some tasks carried out by the computer, plus language based 'facts' gleaned from the media, teachers and other children. Therefore the 'information' perceived will vary in format and, unless these 'items' are stored differently according to whether they are, say, pictorial,verbal or aural etc., they will require some modification by the perceiver if the information is to carry subsequent usefulness. In terms of the value of what is actually represented in the drawings of the novice, it would be extremely useful to know whether some uniform type of change takes place on the original 'incoming information'. For example, the conversion of analogue signals to digital on input and the reverse when information is subsequently externalised whether it be verbally or pictorially. If this were the case the perceiver would, of course, require some extra equipment, some sort of A to D converter, thereby highlighting the need for some specialised treatment for certain types of input. The alternative might be some sort of separate but highly structured organisation, such as relational databases, where the storage format may differ according to the input type. However, this could lead to a distorted concept as some sort of 'compatibility' coding could need to be applied when inter-related items of different format are combined to create a single image or model.

Eysenk(1984) when faced with the argument that knowledge can be represented in the form of propositions but not in images said, ".. it cannot be argued that the mere possession of propositions in long-term memory constitutes knowledge, otherwise it would be necessary to claim that books are knowledgeable. Whether we are speaking of propositions or images, some processes must operate on them for us to have knowledge. If knowledge is regarded as involving active processes performed on stored representations, then it is not obvious that postulating propositional rather than imaginal representations is of much use. Knowledge can be derived from perceptual experiences, and it can also be derived in similar fashion from mental images".

One of the key criticisms of imagery theory is the notion that the translation between perceptual and verbal codes necessitates an intervening code which incorporates propositional representations. Anderson(1978) pointed out the difficulty with this argument, " if all translation between codes requires an intervening code, then we are left with an infinite regress. The translation between the visual or perceptual code and the propositional code requires yet another code(say code X) and the translation between the perceptual code and code X necessitates a further code, and so on ad infinitum".

Eysenk observes that it is indisputably the case that virtually any information can be represented in propositional form, and also that most(or even all) imagery phenomena can be accounted for by means of a formulation postulating only propositional representations.

He also suggests that the two major protagonists in the controversy concerning the theoretical status of imagery (i.e.Kosslyn and Pyslyshyn) both seem to have fallen into the trap of regarding images and propositions as the only possible sorts of representation. This may well be erroneous, he continues, but the lack of systematic attempts to identify other forms of represention means that our discussion necessarily focused on images and propositions.

In sum, Eysenk concludes, attempts to discredit imagery as a psychological construct have proved unsuccessful, and it must therefore be concluded that imagery is very much a viable construct. Propositional theories provide the main alternative to imagery theories but they have by no means demonstrated their superiority. Indeed, explanations of many phenomena in terms of imagery are clearly more adequate than explanations that do not invoke imagery.

Kosslyn(1980,1981) has proposed a computational theory of imagery that attempts to spell out in some detail exactly how the process of imagery occurs. It is sufficiently precise for it to be implemented by means of a computer-simulation model. Images are alleged to have two main components: One is the surface representation (a quasi-pictorial representation that occurs in a visual buffer) and the other is the deep representation, which is the information stored in long-term memory that is utilised in the generation of the surface representation. Image transformation typically seems to occur incrementally, and involves the scan process. The zoom and pan processes are sometimes used to transform the surface image, and an additional process that serves to alter the orientation of the image is the rotate process.

Kosslyn's computational theory has also been used to account for the finding that imagery is sometimes(but not always) used when people are asked questions about the visible properties of objects or scenes(Kosslyn & Jolicoeur1980). The theory assumes that image encodings are accessed in parallel with propositional ones. If the appropriate propositional encoding is located before the process of imaging is complete, then imagery will not influence the decision process. It follows that imagery will most frequently be used when relevant propositional information is either missing from long-term memory or inaccessible.

Those who believe that imaging and remembering in words(descriptionalists) both involve the same kind of abstract descriptive representations do not deny that mental images occur but that there are still many unanswered questions. It is my own belief that the novice user of the computer is capable of creating an internal 'topological' image of the internal machine which is based on his own speculations and practical experience. I believe the image thus created is closely related his mental model of how the interactive processes work and that together they enable further development and refinement of understanding.

These early topological images are likely to vary in intensity, detail and manipulability according to the robustness of the child's understanding at that time and, when asked to draw a picture of what is 'under the lid' of the computer, it is believed that the evoked image will be used as a model for the drawing. From these beliefs it migh also be anticipated that a similar width in detail and continuity will be found in the drawings.

In the absence of conclusive evidence about the coding, strorage and retrieval processes of our 'imaged' knowledge my own impressions best match those of Kosslyn & Jollicoeur(1980). I suspect that a system exists which draws on at least two strands, perhaps propositional paralleled by an imaging ability. The image being produced only when the stored 'item' has become part of the perceiver's understanding and thence undergoes modification with experience. For example, the image evoked by the novice on hearing the word 'computer' may be that of a BBC micro-computer in some form because the novice is familiar with that particular machine. If the instructions are changed and the novice asked to consider the internal system of that machine then the original image of the BBC microcomputer should reflect the modifications that have occurred as a result of subsequent knowledge and speculation by the post concrete transitional child about how it works.

In this way a drawing about the system will reflect the 'point in time' understanding of that system. At the earliest stage the modification may not, of course, have extended beyond the known input and output peripherals for the subject still very much rooted in the concrete operations stage. But for others, at a later, transitional stage, the 'model' produced is likely to be be clothed with partial understandings which reflect some speculative thought characteristic of formal operations. Some of these objects may have been perceived through language. For example, items drawn from everyday life may reflect the subject's understanding of 'technical' or 'electrical' - such as switches, fuses or light bulbs.

Likewise, the model may be 'language' clothed, for example, with rectangles simply identified as RAM, ROM or BRAIN, possibly portrayed thus because there is insufficient knowledge to create a visual image of the item. In the first example, where inapproppropriate components are substituted, it could be assumed that, in the absence of known features, schema relating to non-related concepts but which satisfy the criteria for the novice, are detached and drawn from memory. The value or strength of such images may also depend on the 'quality' of the stored memory.

If we ever think about memories we probably think of them as existing in a physical space. We talk of ideas and thoughts being in the 'dark recesses of the mind', of ideas being 'lost' or 'difficult to grasp' and of 'searching for lost memories'. When Plato[5] attempted to define memory he attempted to link learning, remembering and forgetting. He progressed from statements about knowing and not knowing (related to learning and forgetting) on the first attempt, to describing memory as a block of wax without order.

The wax he saw as being stamped in such a way that some impressions were clearer than others, according to its consistency. Plato was attemping to link learning, remembering and forgetting but I think the example applies equally well in describing beliefs based on intuition as might be carried by the novice.

On his fifth and final attempt Plato introduced the aviary metaphor. Pieces of knowledge were romantically described as being 'like birds in a cage, flying in flocks, groups or singly', hence a concrete image explaining the difference between possessing knowledge and having knowledge; between latent or potential knowledge on the one hand and actual knowledge on the other. Once again a powerful metaphor which describes the condition of the novice who may be acquiring knowledge and experience through exposure to the computer, these are added to the existing miscellany of intuition, assumptions and beliefs but they not yet gelled as a whole. The understanding is not yet robust.

Cognitive Mapping

Cognitive maps, or orienting schema(Neisser) are also discussed as if they were mental pictures of the environment that can be examined by the mind's eye. Neisser believes that the cognitive map is an especially useful form of imagery and it is this particular phenominum which permits the individual to anticipate the locations of places and objects which have not yet come into view. An important feature of cognitive maps is that, as noted above, such a schema can be detached from its 'context' and used as an independent source of information.

When, as adults, we recognise our use of a mental model to represent an absent place or item which we 'know', we are clearly making use of our stored memories which are 'projected' in a form which is perceived as a picture. Therefore, in whatever form they are stored and, based on the assumption that they are not one and the same, memories and images must enjoy a very close relationship. It is this relationship which is important if it is exploited when, for example, the novice is asked to draw a picture of a hidden process which it neither knows or understands. Any stored information which he or she believes to be relevant in the absence of the 'real' knowledge is likely to be called upon and it is the choice of attribute/s which are used to clothe the model that allows us to gain a better understanding of the developing model.

Summary

In this section the area of cognitive development which relates particularly to this study has been examined. It was found that all human beings follow a similar pattern of cognitive development, which begins at birth. By sensing external stimuli through the several sensory systems, information is interpreted against the background of experience brought to the event by the perceiver, thereby creating a particular, individual perceptual event. As a result of successive events involving what has happened and what is happening, behavioural responses undergo constant modification as schemata, complex cognitive structures, which are developed, refined and interconnected.

Piaget's theory has been identified as one explanation of cognitive development. He believes there are qualitative changes in thinking during childhood and describes a continuum of change which takes place between birth and adolescence. Whilst all normal children are expected to pass sequentially through the first three stages, Sensori-motor, Pre-operational and Concrete stages in turn, it cannot be assumed that all adults have reached the final, formal operations stage. Arrival at formal operations liberates the subject from his dependency on the present, immediate situation and instead allows exploration of the possible and hypothetical; progressing from intra-propositional to inter-propositional modes of thought.

The development discussed above includes the ability to image memories by recalling objects and events from the past. And, although there is still much speculation as to how images are stored, images such as cognitive maps, or orienting schema,(Neisser) are powerful tools which enable problem solving in the absence of real life models.

This ability to imagine scenes and objects based on experience is inherent in us all, and the usefulness of those images will vary according to the experience of the perceiver. However, if the images were only mental 'pictures', the mind's eye would be limited to the closed door of a room, cupboard or indeed the outer casing of a machine, but with the existence of 'anticipatory schemas' our dimensions of knowledge are immediately multiplied.

By recalling the image of a familiar object or person we are able to manipulate and inspect that image to gain 'visual' information that may have been forgotten. If the imager was an 'expert' aspects of the image which were 'wrong' or did not 'fit' our original perception would be readily recognized, rather like seeing a modern car in a nineteenth century landscape. However, the image of the novice is likely to include inappropriate elements. The image does not assert, it is seen as a statement of perceived fact.

The mental model, on the other hand is a structure of explanation about relationships. As a result of first or second-hand knowledge, experience and/or observation, these models provide predictive and explanatory power for understanding the interaction between people and their environment and, for the purposes of this thesis, objects and conduits interacting within and making up a computer system.

In this thesis, and in the short term, mental models held by the computer novice are seen as an invaluable aid to communication; by translating a 'sense' of them to a drawing the novice is effectively opening a window onto his developing perceptions of the computer and what it does. In the longer term, and appropriately developed, images and interactive imagery, could be used as part of a diagnostic programmme in the shaping and development of these natural but possibly faulty 'models of the mind'.

The following chapter leads on to introduce two related perspectives. Firstly, a descriptive overview of mental models in the form of a summary followed by a report which seeks to establish mental models as an appropriate route toward the solution of the fundamental problem sought to be addressed by this thesis.

NOTES

1 Schema: A mental framework or outline, which refuses to be sharply defined consciously, is of the order of a set or attitude, but less definite, and functions as a kind of vague standard, arising out of past experience, and placing any fresh experience in its appropriate context and relations.Drever(1964)

2 Piaget's Circular Reaction: Reaction, reflex, or response, the end result of which stimulates its repetition.

3 Piaget's Accommodation: occurs when the cognitive organisation is modified by the need to deal accurately with the requirements of environmental events.

4 Piaget's Concrete operational child. A child, typically around the age of 7 years, in the third stage of development, having already passed through the sensori-motor period and preoperational thought.

5 Plato: The Dialogues of Plato second and third period. From Paul Friedlander Bolingen Series University Press, Princeton.

6. The controversy surrounding the value and effectiveness of visual representation as opposed to propositional representation is recognised. It is the major reason why, in this thesis, alternative problem and language based techniques have also been employed.

The concept of using pictures for data collection is new and the related image/model relationship could be usefully expanded. Examination of the relationship between the novice imager and the image leads one to the conclusion that the perception is probably drawn from a narrow, probably implicitly held, real life experience. Individual images make few assumptions about present state knowledge; witnessed by the lack of explanation portrayed by a 'muddle of wires'. On the other hand there are occasions when imaging is apparently insufficient and recourse is made to verbal expression. For example, the portayal of computer memory by a labelled rectangle.

The imaged system is made up,initially, of component parts where each part is seen as an individual non-inferential step in the building of the whole. It is only when these non-inferential building blocks are seen to exist in relation to one another that the mental model becomes a possibility. Just as geometric figures based on linear relationships are used to draw inferences so can the relationship demonstrated by the imaged system create a mental model which offers the imager a system process solution. It is at this stage that propositional thought may play a part and, indeed, may also provide a solution as to why the images of many subjects are 'immature' and will, therefore, remain no more than picture images whilst others are 'ready' to make degrees of progress into mental model development.

CHAPTER 3

MENTAL MODELS, THEORY AND PRACTICE

Riley(1986) suggests that knowing more about the kinds of models people develop with experience would tell us something about the limits or complexity of the models with which humans reason or at least the kinds of models with which they prefer to reason.

I believe that it is the nature of the understanding held by many novices about how a computer system operates which limits their effective application of it. Further, that when such a user is observed using the computer, the extent of their misunderstanding about the system can be identified from their interaction with the system.

Because our pupils are introduced to the computer through Applications software it cannot be supposed that the problem has arisen out of failed attempts to teach the 'hidden system' between keyboard and monitor. Rather it must be the result of the child's own, naturally occurring, reasoning processes about the system.

These reasoning processes enable the results of experience and observation of most everyday situations to produce mental structures which reflect their perceived real-world counterparts. Such structures enable a level of understanding about the object, its components and their inter-relationships appropriate to the level of development achieved by the modeller.

However, these models are not isolated but form part of the inter-related continuum which begins for us all at birth. Consider, for example, a supermarket shopping basket consisting of a wire mesh base and a pair of swivelling handles. Few supermarket users would have difficulty in generating a mental model of it with sufficient clarity to enable the complete structure to be drawn; basket, handles and the relationship between the two. The relationship between the parts of the basket is straight forward, is easily recognisable from past experience and, most importantly, carries no parts or processes which are concealed from the user.

In the shopping basket example, no gaps exist which need to be filled nor compensations necessary for missing knowledge about its structure during the reasoning and model generating process. The same cannot be said for the computer system and the mental models generated of it by the young novice. For them the model is the result of some kind of understanding about how the device works but the knowlege is based mainly on known input, output and the external behaviour of the device. Most of what he or she is trying to rationalise has been hidden from sight, therefore, the unknown is simply ignored, as perhaps in the early concrete operational child, or an explanation 'which is 'satisfactory to the user' may be applied to 'fill the space' in a subject at a later stage of his or her development.

It is likely to be this 'user satisfactory' substitution within the model which provides resistance to modification of the model rather than the straight forward omission, Therefore, whilst the mental model is expected to undergo modification as a result of continued experience there is no reason to suppose that the modifications take a positive direction if no intervention is made to address the fundamental problem. On the other hand, as long as the unknown area or process remains a blank there is a chance that it will be correctly filled if the user develops further and is assisted by 'helpful' experiences.

There is no doubt, from the research projects outlined below, that the mental models held by novices are complex. They reflect embedded knowledge, beliefs and doubts, and, if the model is badly flawed then learning is likely to be impeded and user methods become ritualistic. Under the large umbrella of embedded knowledge, beliefs and doubts may be found the strongest and most influential factor of all, language. The research shows that language presents a triangular framework of related and, for the practitioner, recognisable problems which I believe effect and are effected by the mental model of the device.

Firstly, as educators, we use the least comprehensible method for communicating with the novice about the computer. We use the spoken word and give scant attention to prior meaning and specialised terminology. Likewise, we expect the novice to give a language based explanation of his or her understanding of what is happening with the system; a task which demands appropriate vocabulary and a robust mental model generally both beyond the tools supplied and beyond the understanding acquired by the learner.

Secondly, we expect the novice to be guided by prior meaning in their interpretation of basic system management, computer Applications and commands. Finally, we encourage problem solving and cross curricular activities using, for example, the high level languages BASIC and LOGO expecting the novice to understand the language of control structure without considering the difficulties of modelling such structures. However, in my opinion as stated above, language should not be treated as a separate issue, the device and the language used, in all its contexts, are inextricably entwined.

On a more positive note, the research clearly indicates that when intervention offers the learner methods of 'gap-reduction' or process 'explanation' which lead to better understanding, a measurably higher performance is achieved. In these cases the results strongly suggest that the inadequate mental model has been positively modified thereby becoming more 'helpful' to the user.

The idea that mental models can be examined, influenced or reshaped when the natural process appears to have fallen short is stimulating and opens up a new perspective on learning . A selection of research projects which are all concerned with analysis of mental models is outlined below and from them a general, descriptive pattern emerges.

For example, there appears to be agreement that, for a user to successfully interact with a system such as a computer, a mental model is a pre-requisite. This model will contain some element or elements which relate to the real world system and the relations between the elements of the mental model will correspond to those perceived in the real world system.

The mental model of how a device such as a computer works , then, is a representation of the relationships within the system and, further, the user should be able to 'RUN' the model to produce a simulation of that system. This qualitative simulation, or envisionment, allows the user to trace the chain of 'cause and effect'. The system's chain of cause and effect is significant for the user because it implies that each new state or condition can be predicted. It is also clear that when the mental model is flawed or incomplete in terms of its chief elements then the user is denied one of the greatest advantages offered by the mental model as a result. The lack of means through which the user can successfully infer and predict information about the system goes some way in explaining why many pupils are totally lost if the system responds in an unexpected way. Unable to solve the problem presented, the novice repeatedly resorts to switching the machine off and re-booting it, hence compounding the original problem and, in addition, perpetuating the novice's sense of 'not being in control'.

The research projects on mental models or the conceptual model, reported below, vary in approach and perspective but are all concerned with improvement of understanding. By understanding the processes of mental model generation better we are looking for ways of generating and using conceptual and device models to promote increased motivation, faster learning, better retention and inference of 'how it works' information about system components. And, for the computer system user in particular, an improved human-computer interface.

A solution through mental models is identified

Mental model research is seen by Gentner & Stevens(1983) as a confluence of two independently developed lines of research; cognitive psychology and artifical intelligence. Mental models research is defined as fundamentally concerned with understanding human knowledge about the world. They add that a typical piece of mental model reseach is characterised by careful examination of the way people understand some domain of knowledge. The domain is selected where normative knowledge exists and is relatively easy to detail explicitly. The methodologies used are eclectic and can include protocol analysis, traditional experimental psychology, developmental studies and expert-novice studies.

From their analysis of mechanistic mental models de Kleer&Brown(1983) suggest three distinguishable forms of learning which one might engage while acquiring an understanding about a new machine - which could, perhaps, be an Automatic Teller Machine(ATM), Payne(1991) or a standard computer.

The first kind of learning establishes a connection between the structure and function of the device, the user implicitly projects functional evidence onto the observed structure thereby producing 'hidden assumptions' in the computer models. The second kind of learning allows the implicit assumptions to be made explicit thus making the structure function connection more robust. Each time the experience is repeated the learner is able to modifiy the model held and the implicit assumptions made explicit thereby increasing the robustness of the mental model. The third type of learning is the storing of the results of the projection problem-solving on the intrinsic mechanism.

It appears that as one gets more exposure to devices, and hence develops component models with fewer hidden assumptions, understanding the next new device becomes harder, rather than easier. That is, the more experience one has of understanding devices, the harder it is to construct another device which is built out of the same components. However, one advantage, is that as the learner's familiarity with the domain and machine increases, he will develop a more powerful technical vocabulary for describing functional speculation. In turn, this will make his projection problem-solving simpler. When programming, it is suggested by Kurland&Pea(1983), that such 'experts' are guided by efficient mental models of how the programming code controls the computer operation; it is also suggested that a user may have what appears to be not one but multiple models for the same constituent component even though it is the result of a single envisioning model.

The model is used to discover methods, which are then learned, and which become more automatic with experience. Norman(1983) also suggests that as skills improve the novice relies more heavily on the methods derived. The mental model becoming used more as, for example, a device for locating errors in methods and their correction. This is one of the original problems upon which this thesis is based.

According to Johnson-Laird(1983) mental models play a central and unifying role in representing objects, states of affairs, sequences of events, the way the world is, and the social and psychological actions of daily life. They therefore enable individuals to make inferences and predictions to understand phenomena, to decide what action to take, to control its executions, and above all, to experience events by proxy.

The novice needs no encouragement to initialise the development of a mental model; it is clearly a natural process. Payne(1991), for example, found that when questioning his subjects they had already constructed mental models of automatic teller machines(ATMs). They had speculated about the inner workings of the system in advance of being prompted to do so. From his studies with calculators Mayer(1981) concludes that students have intuitions about how such devices operate and his work shows that the intuitions of any individual user are fairly consistent. I believe that by developing elementary mental modelling techniques related to computers and computing during early childhood, we are giving positive and valuable developmental experience to a generation of young people about whom computer literacy will be assumed.

The nature of mental models

In principle, a novice user begins with a general model of the computer system which helps him to become more familiar with the system.(Norman 1982).

The synthesised connection between the structure and the function of the device are the 'inferred' models or 'primitives' and include how the machine is intended to function. A component model is a description of the input-output behaviour of the particular component. The model's rules make no reference to how the component achieves the input/output behaviour, this behaviour is explained by the 'embedded' models. Williams,Hollan&Stevens(1983) describe a mental model as a collection of 'connected' autonomous objects. By mental model is meant some kind of understanding of how the device works in terms of its internal structure and process,(Kieras&Bovair1984). Moran(1981) states that it is widely accepted that peoples' ability to use an interactive device depends in part on their having access to some sort of mental model.

As noted earlier, complex devices, such as machines, are built from combinations of simple components and the behaviour of the components can be described qualitatively. Further, de Kleer&Brown(1983) have distinguished four related notions which form the basic distinctions for a theory of qualitative reasoning, and gives the opportunity to define terminology, they are:

1. The device topology is a representation of the structure or physical organisation of the device.
2. Envisioning, an inference process which, given the structure of the device, allows the function to be determined.
3. The causal model, describes the functioning of the device in terms of how the components causally interact.
4. Running of the causal model, produces a specific behaviour for the device by giving a chain of causally related events.

They add that the distinction between the intrinsic mechanism and the correct causal model suggests a possible solution to the long-standing issue of why people find some faults so much harder to troubleshoot than others.

De Kleer &Brown argue that the analysis of complex devices should be split into two relatively independent parts. The first deals with what the attribute values are when the machine is 'at rest' (i.e. equilibrium) and the second with how disturbances from that equilibrium propagate through the device. Because the analysis is concerned with disturbances from equilibrium generated by a local cause de Kleer and Brown say that it can be fairly reliably assumed that those disturbances which have been discovered causally dominate those which have not. The strategy is, until proven otherwise, to assume temporarily that unknown attributes have negligible value.

In their detailed study of mechanistic devices, and in particular the 'buzzer', de Kleer&Brown(1981), were aiming to achieve a better understanding of complex system modes: to find out what elements form a mental model,what characteristics heighten their usefulness to the expert for problem solving. Also, if an ideal efficient model is held, why are these essential properties important and how can these characteristics be used to improve the acquisition of skills for constructing mental models.

They describe the 'running' or qualitative simulation of a model in the mind's eye as 'envisionment' and credit envisionment with the ability to reveal the system's chain of cause and effect thus enabling each new state to be predicted with minimum amount of reasoning.

The constructing of a mental model involves:

Device structure = P1 =>P2=>causal attributes

where P1 represents examination of the device and P2 represents examining the result of the envisionment. A mental model being a combination of an envisionment and its causal attributes.

By using their 'no-function-in-structure' principle they see each component part of the system represented and understood in a 'context free' form i.e. independent of the overall system. If the principle should not be in operation and a defective or modified component in use, it renders the prediction of how the system might function using the simulation 'highly questionable'. In an attempt to explain the not-easily-understood aspects of how a device works de Kleer&Brown say the learner will often consciously disregard the no function-in-structure principle.

Working with mathematical learning models Greeno & Steiner(1964) introduced the notion of 'identifiability' and point out that a 'useful' model will carry a clear correspondence between parameters and states of one's mental model and the thing one is trying to describe. As noted above, they allow the subject to infer and predict.

Problems presented by mental models

Rupietta(1990) reminds us that there is no way to directly access the user's mental models other than observing his behaviour and utterances. Therefore, we assume that if a user has some task to solve he 'consults' his mental model of the system to decide how this can be accomplished and predicts how the system will behave. A mental model is only sufficient if it allows the user to solve problems with the system and, an important factor, to perceive its limits.

Payne(1991) concludes from his descriptive study of mental models based on High Street ATM, that a unitary description may not suffice largely because mental models are constructed from many sources. They may be influenced by culture and linguistic metaphor as well as the subject's natural preferences for direct empirical explanation.

Williams, Hollan&Stevens(1983) note that the ability to use inaccurate and often incomplete models is an important characteristic of human reasoning. And, as Norman (1983) also observes, these naturally evolving models are not precise, or elegant. He notes that generally:

1. Mental models are incomplete
2. The ability to 'RUN' models is severely limited
   (Where RUNNING a mental model corresponds to modifying the parameters of the model by propagating information using the internal rules and specified topology.(Willliams, Hollan & Steven 1983)).
3. Mental models are unstable: people forget the details of a system they are

using, especially if those details(or the whole system) have not been used for some period.

4. Mental models do not have firm boundaries; similar devices and operations get confused with one another.
5. Mental models are "unscientific". People maintain "superstitious" behaviour patterns even when they know they are unneeded because they cost little physical effort and save mental effort.
6. Mental models are parsimonious. Often people do extra physical operations rather than the mental planning that would allow them to avoid those actions; they are willing to trade off extra physical action for reduced mental complexity. This is especially true where the extra actions allow one simplified rule to apply to a variety of devices, thus minimizing the chances of confusion.

Kurland&Pea(1983) note in their LOGO study, see below, that most children between 8 -12 years are tied to using the simplest iterative methods because they have not accommodated the required control structures of the language. This supports the point made by Greeno&Steiner(1964) that certain kinds of mental models will be ruled out if the identifications cannot be made easily.

In addition, beliefs and doubts are embedded in these mental models and, even when the knowledge is correct, people often feel uncertain. For example, through selective analysis of conversational materials Payne(1991) found that when subjects speculated about the inner workings of the system they were seen to freely mix explanation types - design rationale in one breath and direct explanation in the next. They juxtapose pre-existing conceptions with novel constructions or summon special purpose analogues to deal with small components of the total system. Further, by preferring to quote first hand experience to justify claims about the behaviour of the device the users reveal the implicit nature of their understanding about the system.

Any assumptions held implicitly cannot be articulated, explanation requires a robust model. However, each time the experience is repeated the learner is able to modify the model held and the implicit assumptions made explicit which, in turn, will help to increase the robustness of the mental model.

The problems associated with language and function is illustrated by a study carried out by Mayer(1981) with 30 undergraduates using the high level programming language BASIC. The results of the study showed that although as beginner programmers they were able to perform adequately on mastery tests in program generation, they possessed a wide range of misconceptions about statements such as

INPUT A, READ A, and PRINT C.

The study comprised nine statements such as LET A=B+1 which were broken down into a list of transactions by two scorers. Three types of transacton were observed for each statement, plus an empty category. For example:

LET A=B+1

1. Correct transaction: store the value of B+1 in memory space 'A'.
2. Incomplete transaction: store the value of B+1 in memory.
3. Incorrect transaction: store the equation A=B+1 in memory.
4. Empty: if the subject produced no correct or an incomplete version of the key transactions.

The following is a summary of misconception and lack of understanding found in the study:

1. Input Statement:
   difficulty in conceiving where the 'to be' input data comes from (i.e.keyboard) and how it is to be stored in memory(indicated memory space). Failure to understand executive control - computer will wait for input from the keyboard as cued by '?' on the screen. Major misconception is INPUT A- letter A is input and stored in memory.
2. Read-Data Statements:
   Where the to be READ data comes from, the INPUT cue or DATA statement and how it is stored in memory,(i.e. specified memory spaces).
3. Conditional and Simple GOTO Statements:
   What happens if condition is false? IF A<B GOTO 99 without test.
4. LET:
   Confusion on 'solving' or 'storing' an equation equal signs as equality.
5. PRINT statements: PRINT C and PRINT "C"

STATEMENT	CORRECT%	INCOMPLETE%	INCORRECT%	EMPTY%
INPUT A	3	30	30	37
READ A	10	27	17	47
IF A<B GOTO 99	27	27	40	7
LET A=B+1	27	10	60	3
DATA 80,90,99	27	17	13	43
GOTO 30	27	56	10	7
PRINT C	33	0	47	20
LET D=0	43	3	53	0
PRINT "C"	80	0	13	7

A detailed transactions list appears in Appendix(1)

Although the above work was carried out with undergraduates, the study serves to reinforce the writer's own experience with children between the ages of 11 and 18 years and indicates that the age of the novice programmer in this particular is not a significant factor.

Mayer concluded that his study provided evidence that "hands-on" experience is not sufficient for the productive learning of computer programming by novices. Users tend to develop conceptions of the statements that either fail to include the main idea or that include outright misconceptions. Explicit training is needed which includes the use of concrete models.

As noted elsewhere, to be able to describe and predict the behaviour of a system the novice requires the basic tools for the job; the appropriate language, terminology and understanding. Riley(1986), for example, argues that internal coherence is aided by several forms of knowledge about the action and syntactic structure of the commands used, how the command works and about the system objects involved. In addition, if as educators, we believe there are positive advantages to be gained by using high level computer languages such as LOGO, PASCAL and BASIC with novice computer users then a better understanding about the effect of language and its key relationship with the mental model is essential.

Kurland & Pea(1983) investigating the use of mental models in recursive thought and recursive programming procedures noted that 'expert' programmers work with, and are guided by, efficient mental models of how the programming code controls the computer operation. On the other hand, the novice's faulty mental models cause conflict between the present-state model and program behaviour and, as a result, the existing schemata are modified. The models are adapted in response to direct instruction and feedback from their own programming and debugging experience.

Largely due to Papert(1980), many educators believe that children can learn recursion through self guided explorations of programming concepts in LOGO. However, observations of 8-12 year olds indicated that most avoid all but the simplest iterative programs which do not require the deep understanding of control structure; the prerequisite for an understanding of recursion. They say that a subject must acquire the insight necessary, the anticipatory schema, to generate a recursive description and implement it in specific programming formalism.

In addition to understanding recursion the child must understand the logic and terminology governing the language of control structure. Adult novices have trouble with both. When learning to program, they have great difficulty in thinking through flow control concepts such as PASCAL'S 'while' loop construction(Solway, Bonar and Ehrlich 1983) and tail recursion in Solo, a Logo like language (Kahney and Ersenstadt 1982), even after extensive instruction. Furthermore Bonar(1982) finds that prior natural language understanding of programming terms misleads the novice programmers in their attempt to explain how a program works. Prior meaning is brought to the task of constructing meaning from lines of programming code.

In effect we expect children to be guided by their natural language in their interpretation of programming language constructs, and by faulty mental models of flow of control structure. For example in the execution of Logo a program proceeds line by line. However, when a procedure calls another procedure, or itself, this inserts all lines of the named procedure into the executing program at the point where the call occurred. Control then proceeds through each of these new lines before carrying on with the remaining lines of the program. This control is **passed forward** to the called procedure and **passed back** to the calling procedure. The last command of all procedures is End. The End signals the completion of the execution of one logical program unit and directs flow control back to the calling procedure so the program carries on. There are exceptions to the line by line execution rule. An important one for recursion is the Stop command. Stop causes execution of the current process to be halted and the control to be passed back to the calling procedure. Functionally, Stop means to branch immediately to the nearest End statement.

For the Kirland and Pea study short Logo programs were constructed of procedures to reflect 4 levels of complexity

1. Procedures using only direct commands to move Turtle
2. Procedures using iterative repeat command
3. Tail Recursion procedures for example:
```
TO SHAPE B:SIDE
IF: SIDE=20 STOP
 REPEAT 4(FORWARD:SIDE RIGHT 90)
RIGHT 90 FORWARD :SIDE LEFT 90
SHAPE B:SIDE/2
 END
```
4. Embedded recursion program for example:
```
TO SHAPE C: SIDE
IF: SIDE = 10 STOP
SHAPE C: SIDE/2
REPEAT 4(FORWARD :SIDE RIGHT 90)
RIGHT 90 FORWARD:SIDE LEFT 9
```

The Kurland&Pea Study involved seven children 11-12 years of age who had averaged 50 hours classroom programming time in Logo and had demonstrated that they could use iteration and recursion in some context were given a written program. They were asked to :

a. draw by hand what the program was expected to do
b. run the program and
c. explain the discrepancies plus any additions presented by the program.

The results of the study showed that whilst all the children made accurate predictions for programs at the first two levels no child made accurate predictions for either level of complexity involving tail recursive procedures or embedded recursion procedures. The childrens' problem in explaining embedded recursion are traced to two related sources; general bugs in their mental models for how lines of program code dictate the computers operations when the program is executed, and the particular control structure of the embedded recursive procedures. Kurland and Pea suggest the need to teach program control structure, such as recursion, rather than expecting children to discover them on their own.

The Kurland and Pea study, whilst carried out with a group too small to support a convincing argument, does highlight a commonly observed type of self-imposed rote learning of instructions and ritualistic behaviour. Mayer(1981) established that the computer is not the only machine observed to cause the learner difficulty. Although most secondary school children use electronic calculators and appear to know how to use them correctly, many carry mis-conceptions with regard to how they work. As a result their use is often modified and ritualistic.

<u>Manipulating mental models,passive development versus active intervention</u>

Mayer(1981) concluded that his study provided evidence that "hands-on" experience is not sufficient for the productive learning of computer programming by novices. Users tend to develop conceptions of the statements that either fail to include the main idea or that include outright misconceptions. Explicit training is needed including the introduction of a concrete model showing the key locations in the computer (e.g. memory spaces, input stack etc.). Verbal and visual descriptions of the key transactions for each statement and encouragement of the user to role play 'what the computer is doing' for statements and programs.

Mayer's suggestion of a concrete model showing the key locations in the computer is one of the two approaches to learning computer programming distinguished by DuBoulay&O'Shea(1976) and DuBoulay, O'Shea & Monk(1980) 'Black-box v Glass-box'. In the Black-box approach no idea on internal working is considered; memorising procedures, key strokes as commanded, sequences which work etc. On the other hand in the Glass-box approach the user is able to understand the changes that occur inside the computer for each statement but not at machine code level.

Du Boulay, O'Shea & Monk(1980) have indicated that novices should be exposed to a 'notational machine' which is an idealized model of the computer implied by the constraints of the programming languages. For example DuBoulay & O'Shea have developed a 'Logo' machine to represent the internal actions that occur for Logo states. They also suggest two important properties:

1. Simplicity - there should be a small number of parts that intersect in ways that can be easily understood.
2. Visibility - novices should be able to see 'selected parts' and 'processes' of this notational machine in action.

Support for learning by understanding through such methods as Glass rather than Black-box is also shown in past research by Gestalt psychologists[1] who suggested that the method should lead to superior retention and superior transfer to novel problems. It is assumed then that the method enables preparation with the correct anticipatory schema, otherwise the learner would continue to see only what he knows how to look for which may be insufficient for the task: it is these schemata together with the information available that determines what will be perceived.

Such a method was used by Bayman(1982) in his study on the effect of instructional procedures on beginning programmers' mental models. He studied the conceptions of what happens inside the computer when statements are executed and questioned whether mental models were important for succes in solving problems. Ninety five undergraduates were observed, all were unfamiliar with computer programming in BASIC. They were split into two groups; the first was given a standard manual and the second the manual with procedural and/or pictorial representation on the internal working of the computer (conceptual group).

Method: Group 1 studied the manuals at their own pace during two 1 hour separate periods over one to two weeks whilst group 2 attended the laboratory for two 1 hour separate sessions to take tests.

The tests comprised:

1. Problem solving	- generally interpret or debug program or single program statements (18 programs)
2. Fact retention tests	- asked to fill in blanks on 20 sentence directly taken from the manual.
3. Procedure specification tests	- where asked to write down in plain English the action taken by the computer to execute 10 different program statements.
4. Content specific tests	- given a hypothetical model of the computer in diagram form instructed to fill in the necessary parts of the diagram for the same 10 sentences as in the procedure.

An extract from the pictorial representation used appears in Appendix 2.

Bayman summarised by giving answers to the orginal questions as, follows:

1. Did different instructional processes influence beginners conception of statements. Did they benefit from information given.

The study suggests that the instructional procedure for the conceptual groups influenced only the lower aptitude subjects conceptions of the BASIC program statements. Higher aptitude subjects seem to be able to abstract the correct meaning of the statement irrespective of the instructional procedures used in the study.

2. Did instructional procedures influence learners ability in solving programming problems.

The results suggest that the instructional procedures provided for the conceptual groups enhanced the lower aptitude subject to solve programming problems but hindered the ability of the high ability subject. One possible explanation for this detrimental effect could be that the high aptitude subjcts could not use their processing skills effectively in the presence of the instructional procedure used in this project.

3. Are individuals conceptions of the BASIC statement instrumental for the programming performance.

The results suggest that their conception of the programming statement does seem to have influenced their programming performance. Bayman assessed the top quartile of results - at least 8/10 correct as having a 'good mental model' and assessed the lower quartile of results - 4/10 and fewer as poor mental model.

Bayman concluded that the study provides evidence that subjects with lower mathematical aptitude benefit from studying description of the internal working of the computer as they attempt to learn their first computer language. The instructional procedures seem to have helped the beginning programmer:

a. in the generation of accurate conceptions of the programming statement.
b. in the acquisition of elementary programming skills.

A further perspective is added by DiSessa(1986) who points out that because structural models are removed from application, they are attractive as universal 'mechanistic' ways to understand a system but that such a perspective slights functional understanding. Her paper suggests that, in recognition that function is as important as structure in a language, functional and distributed models might be usefully promoted before structural models. She suggests aiming for a simple structural model, but to count on functional and distributed models at earlier stages of learning.

Distributed, Functional and Structural Models

In distributed models, learning is accumulated through a spectrum of partial understandings rather than as a single functional frame. The idea of a distributed model is derived from ideas developed about understanding complex systems in other domains.

These models represent a patchwork collection of pre-existing ideas in the learner, "corrupted" to new ends. DiSessa stresses that even if 'model' is too strong a word to describe them, "understanding " is not. For the kind of learning involved explains some important data on how people come to understand complex systems. For example, learnability of the procedure definiton process in Logo is due to its naturalness as a solution to a particular problem when interpreted in a number of frames, each of which partially explains the solution. The list of frames for procedure definition syntax includes:

- Δ a clear functional frame - the problem is making a new procedure; the solution is TO: TO SQUARE sends programmer to EDIT mode where the procedure named SQUARE can be written.
- Δ language is in English
- Δ Visual pattern matching
- Δ Rationalisations

Distributed models such as this one involve learning by prototype in the sense that users learn a construct primarily by example. An instructor often simply shows the student how to do something, much like a parent introducing a child to a new object by saying its name and pointing to it. Rationalisations and other partial understandings provide a backbone of reasonableness which allows the user to remember the form of the construct and understand some of the variation possible.

DiSessa believes that function, offers powerful leverage in coming to understand a computer language and helps to promote understanding about the way some prewritten code works.

Functional models have some defects, and unlike structural models one cannot expect the general behaviour of the system to be evident in a specific context. They provide only a view of part of the system and that only with respect to a non-universal frame of analysis. Again one needs a repertoire of them; the notion of a single model is tenable structurally but not functionally. On the other hand from the point of view of incremental learnability, teleology and other important aspects of understanding, functional models can be superior precisely because of their contextural specificity.

An example of a functional model is given in Young(1981) with the algebraic calculator, which allows entry in near algebraic format using parentheses. By following the format used with 'paper and pencil' method, pressing '=' does precisely what is wanted, it gives "the answer". As diSessa points out, 'doing what one wants' is quite a sufficient model of the system for many purposes. The general schematic of functional models is that one has a descriptive frame that includes recognizable objects. Given such an example functional models might be described simply as rules, e.g. "to get the answer, press = " but account has to be given as to why it is a memorable 'rule' i.e. it fits into the previously understood scheme of goals.

In general terms one must consider long-term effects of particular functional and distributed models. Some will remain and be integrated as 'special case' models. Some will fade away naturally and be replaced where appropriate by structural models. For the example quoted above, no learner believes Logo is English for very long. But we must be aware that globally destructive misconceptions may be fostered as well as 'profitable misconceptions'

The fundamental distinction between structure and function focuses on either (a) Structure - characteristics of an object or actions which are defining and independent of specific use or (b) Function - characteristics which have to do with a specific use, consequences or intent. The point is to separate descriptions according to whether the implied descriptive frame universally applies(structure) or not (function).

For example, the structural aspects of a variable in a computer language are given primarily by the rules for setting their values and for getting access to their values. These rules apply in all contexts. In contrast, a variable's functions might vary. Sometimes they might be described as 'a flag' or more generally, as 'a communications device' At other times a variable might fuction as ' a counter ' , 'data' or 'input'

DiSessa defines a structural model as one that is intended to capture the computational mechanism in such a way as to offer explanation and correct predictions in uniform terms. It establishes the bottom line for the user for saying what will happen, given the present state, and for saying what state must have existed, given a particular behaviour. Structural models are what one usually sees when a model is explicitly taught. A tutor will typically evoke one when a novice's expectations have gone awry.

Two of the most elaborate structural model examples that are attempts to encompass a large part of the operation of a computer language are the actor model Smalltalk (Tessler 1981) and the version of Papert's little man model of Logo done by Edinburgh Logo Project (duBoulay, O'Shea and Monk 1981)

DiSessa suggests that having a structual mental model would seem the ideal way to understand a system although this notion is, in isolation, part of the conspiracy to oversimplify understanding. He outlines the problems with structual models:

1). Learnability: structural models aim at being uniform views of rather complex systems, they themselves tend to be complex. When pushed to cover all behaviours of a system they can lose their air of coherence with add hoc elements for dealing with loose ends.

Because of their compact, tightly interconnecting natural structural models require a good deal of learning before they can be applied to even simple, every-day events.

2). Structural models, though very good for debugging are typically rather time consuming to 'run'. Routine, relatively fluid interaction with the system cannot be expected to occur as a direct result of acquiring a structural model.

DiSessa questions the very kind of knowledge available with a structural model, which makes such models unsuited to certain tasks. Hence a fundamental gap exists between the functional level of description needed for planning and the structural level. For example, in order to leave some horizontal space between two output lines in some programming languages, the programmer has to "print" an empty line (e.g. in Pascal, do "WRITELN" the write-a-line-of-text command, but without an argument), and, not surprisingly, novices must usually be taught this "trick"; it is a potential function not easily seen in the meaning of the command.

Development is seen through a pattern of function-structure-function type sequence. From function(or through even less systematic means) one gradually develops a sense for the invariances, the structure underlying function, through experiences in a broadening set of contexts. Eventually, a new wave of functionality becomes accessible as this context invariance can play itself out in initially unimaginable contexts and functions

The first, functional stage reflects the opportunistic connection to and exploitation of previous knowledge that one must expect if we want learning to progress spontaneously and naturally. The structural stage indicates a system with sufficient internal coherence so as to be understandable in its own terms. Such coherence is also important, in its own way, to learnability and stable remembrance. The final functional stage demonstrates that the power and expressiveness of the system indeed transcends the pre-existing knowledge systems, from which it might inherit a great deal.

The mental model - device model partnership

Much work has been carried out by Mayer(1982) in an effort to establish ways of increasing the understanding held by novices about computers and computer programming. As a result he produced the following outline:

1. provide the learner with a concrete model of the computer
2. encourage the learner to actively restate the new technical information in his own words.
3. assess the learner's existing intuitions about computer operations and try to build on them, or modify them as needed.
4. provide learners with methods for chunking statements in a long single, meaningful unit
5. provide learner with methods for analysing statement into smaller, meaningful parts.

Although all five parts are considered important and form a 'whole', Mayer's first and second points are, I believe, the most immediate to the educational process. The second point was considered by Kenneth Craik in 1943 in "The nature of Explanation" (the book responsible for the notion of 'mental model' in the first place) when he said that introspections are at best glimpses of a process rather than detailed traces of its operation " ....if you cannot understand something you cannot explain it". Three possible steps in the process were recognised:

1. First the need for the 'translation' of some external process into an internal representation in terms of words, numbers, or other symbols.
2. The derivation of other symbols from them by some sort of inferential process.
3. A re-translation of these symbols into actions, or at least a recognition of the correspondence between these symbols and external events as in realizing that a prediction is fulfilled.

When dealing with the mechanistic mental model, where the mental model of a computer, for example, refers to the "hidden" information processing that occurs between input and output, then "Cookbook" style instructions according to Clayton Lewis(1986), are not robust. Learners have difficulty in getting back on track after making errors and there is a poor transfer to other tasks..........."I did it by I don"t know what I did" ....etc. Therefore, we must look beyond such traditional methods.

A novice explanation of a process carries explicit evidence of the need for system 'expectations' to be fulfilled; as is noted in comments made by David Owen(1986) who writes:

> "When people claim to 'understand' something they often mean that they can follow a chain of cause and effect down to some particular level. At that level, which depends on the individual, an article of faith precludes the need for them to pursue the chain further. For example, I understand that when I turn the power/volume knob on my radio, it completes a circuit allowing electricity to flow from the battery to some electronics. I neither know nor care what the electronics does. But just on this level of understanding, it would not make sense to me if one day the radio started to emit sounds without my having touched it, or without there being a battery in it."

These novice expectations are also reflected in their explanation of what the device is doing whilst carrying out a task. Through selective analysis of conversational materials Payne found that subjects had already constructed mental models of automatic teller machines. They had speculated about the inner workings of the system in advance of being prompted to do so. Subjects were seen to freely mix explanation types - design rationale in one breath and direct explanation in the next. They juxtapose pre-existing conceptions with novel constructions or summon special purpose analogues to deal with small components of the total system. Further, by preferring to quote first hand experience to justify claims about the behaviour of the device the users reveal the implicit nature of their understanding about the system.

One response to the particular problem of filling conceptual gaps about system processes is to encourage active exploration of a system, such as that suggested by minimalist instruction(Carroll 1985). The learner is encouraged to take responsibility for setting goals and finding system facilities to accomplish them. However, even when the system is designed to permit this mode of learning, so that mistaken actions can be undone, for example, such learning can be inefficient and frustrating. Learners may waste a great deal of time looking in the wrong parts of a system, or looking for things that really won't help.

In the light of such comments the possible success of a device model, which can cut through such diverse influences by relating 'task' to components and supplying possible power flow paths is beyond estimation. The mental models promoted by it must be able to explain and predict the system's behaviour as well as help the user to explain why certain steps are necessary to carry out some procedure.

In studying the use of the calculator, seen as one of the most relevant topics studied in relation to the computer, Mayer&Baymen(1981) used the 'Transaction' approach where it was necessary to specify a triplet of location, object and operation.

The possible **locations** within the calculator are:

1. Display: the external display normally consists of at least eight places, where a place can hold one digit. The display fills from the right.
2. Register: an external register is inside the calculator and consists of a series of sub-registers that hold individual numbers and operators. Expressions are held in the order of input, with the first number on the left, followed by the first number on the left, followed by the first operator etc.
3. Keyboard: The external set of keys include number, operation and equal keys

And the possible **objects** include:

1. Numbers: A number is any simple or multiple digit sequence such as 2,14 or 156
2. Operation: An operation is a mathematical symbol for arithmetic computation such as addition(+) or multiplication(x)
3. Expression:An expression is a sequence consisting of numbers and operators such as 2+3 or 2+ or 2+3x7

Mayer believes that students have intuitions about how calculators work and that his work shows that the intuitions of any individual user are fairly consistent. He also believes that there should be as close a match as possible between the user's conception of how the calculator should operate and the actual operating system of the calculator.

For example, it is thought that the calculator should evaluate when an operation key is pressed after a number rather than in the existing infix form. Polish notation was developed by Jan Lukasiewiez because it makes the design of compilers easier if the arithmetic expressions are presented to it in what is called reverse polish notation. For example, X=A*B + C/D infix form becomes XAB*CD/+ = reverse polish notation and can be achieved through postorder traversal of a binary tree data structure. Mayer concludes that novices tend to have much more diverse conceptions and are significantly different from experts as can be seen in the following error:

Example error - '*incrementing display*'

7+=	interpreted as	7+7= 14
7x=		7x7= 49
2+3+=	either	10 or 8

Because Mayer believes that students have intuitions about how calculators work he suggests that teachers can build on these intuitions. He says they could be used to transfer to programmable calculators, to BASIC and other languages.

It appears that, ideally, a sequence of explanations is required starting with one built around component models which contain the implicitly embedded problem areas. By using the component model expediency the "explanation" is allowed to pre-suppose part of what it is trying to explain. In the initial stages this method can serve to develop a cognitive framework for later refinement using models which carry fewer implicit assumptions. In short, violation of the no-function-in-structure principle(deKleer&Brown1983) on the first introduction of a new system is best as long as the framework on which to 'grow' a correct mental model is firmly established. This could be particularly relevent, for example, in the school environment where development changes can be taking place over a 6 to 10 year period of time.

DeKleer&Brown's initial description of how a simple Buzzer worked served to illustrate that much of the description pre-supposed how it worked. The situation had to be reconsidered in order to connect the functioning of the device, i.e. the buzzing, with the structure of the device i.e. the buzzer, without pre-supposing how the device funtioned. They assert that a meaningful analysis based on either quantitive or qualitative component models must not allow any of the solution, that is function, into the description, i.e. structure, of its pieces.

The development of a causal model required the expansion of the syntax to include 'condition testing' thereby making it possible to identify the reason for component 'change of state'. For each state the definitional parts are asserted and each test examined to determine the consequences of those assertions if true. From these assertions the next state is determined. As pointed out by deKleer and Brown their models carry on-off states rather than, as in reality, transitional, or changing conditions.

The model for the buzzer SWITCH

SWITCH (1): OPEN(2)
Battery is disconnected (3),
if coil is not pulling(4), switch will become CLOSED(5)
:CLOSED(2)
Battery is connected(3)
if coil is pulling(4), switch will become OPEN(5)

(1)COMPONENT (2) STATE (3) DEFINITION (4) IF TEST (5) CONSEQUENCE

Syntactic restrictions can be placed on the model to ensure that they refer only to local quantities, particularly in rule consequences, in order to avoid violation of the no-function- in-structure principle.

A further group of principles which facilitates the learning process is introduced with the most important member, the 'weak causality principle'.

In this case every event must have a direct cause and not use any indirect arguments in determining the next state. "A learning system following this principle can discover where to assign credit or blame in the current model". If, however, the same principle is satisfied and viewed from the 'performance ' perspective they conclude that it enables the next possible states to be determined without requiring elaborate reasoning.

In addition to the model of the components deKleer&Brown considered the connection by which the models communicate. Particularly relevant when training users of the computer, for example, because of the fundamental concept of data flow within the system. Whereas the internal state of the component is inaccessible to other components and connections, models for components and connections communicate by sharing information. They describe the simplest model for a wire as one that consists of the knowledge of the current through it. It shares this information with the components on either end of the wire. The only information that is shared by the connections are attributes that are related to the actual physics by which components interact e.g. voltage, current, pressure, force, flow etc. The no-function-in-structure also applies to connections.

Another important factor pointed out by deKleer&Brown is that models should not predict that a machine still functions when a vital part is removed. Their deletion principle, in application, checks whether the functionality of some part is implicitly embedded in the structure of other parts.

<u>The benefits of working with device models</u>

Training users with mental models, Borgman(1982) believes, can improve performance. As defined above, a mental model is a representation of the relationships within the system; a person can 'RUN ' the model to produce a mental simulation of the systems actions. From the simulation, ways of dealing with the system are developed. Research indicates(Borgman) that if the learner is not given a model he may not make the effort to infer one or he will infer one from the system behaviour and the model is likely to be incorrect. However, it is also believed that even if the learner is given a conceptual model, he may not translate it into an appropriate mental model. Similarly, Mawby et al(1984) warn that "If the child novice's model of the computer functioning is badly flawed, the model will impede rather than support their learning about and with computers; that is children may acquire low-level skills, but the deeper conceptual understanding that allows skills to develop and generalise may elude them.

Kieras & Bovair(1984) describe two strongly opposing 'intuitions' about the role of a device model in learning how to operate a device or in being able to operate it once it is learned. The opposing intuitions are 'of great value' or 'unnecessary', and are held by 'most psychologists' and 'technological industries' respectively. They report that experimental attempts to demonstrate positive effects of teaching subjects a device model for a computer system have been inconclusive. As in the case of Mayer, some studies,using an ordinary calculator, are reported to have discovered strong positive effects on having a device model in some tasks, but not others.

Kieras&Bovair report three studies, the first demonstrating that a device model can result in faster learning and better retention of operating procedures. The second demonstrating that the device model can be used to infer the operating procedures(both from Kieras & Bovair 1983) and the third, (Kieras 1984), examines what is the critical type of information in the device model materials.

In the first two studies, groups of subjects who learned a set of procedures for operating a simple control panel device consisting of switches, pushbuttons, and indicator lights were compared. The goal of the procedures was to get a certain indicator light to flash. One group learned a set of operating procedures for the device by rote, and the other learned the device models, which took on average 1141s, before they received the identical procedure training. The model group learned the procedures faster, retained them more accurately, executed them faster, and simplified inefficient procedures far more often, than the rote group. The results are reproduced in Table(3).

The first experiment shows that having a device model does improve performance on learning and retaining the operating procedures for a device. Kieras & Bovair questioned how knowledge of the device model produces these effects and they conclude that knowledge of how a system works helps by enabling the user to *infer* how to operate the device. They suggest that the device model simply allows the model group subjects to reconstruct by means of inference the operating procedures even if specific details of the direct memory of them have been forgotton.

Table(3)Summary of Results on Learning Procedures with and without a Device Model.

	Rote	Group Model	Improvement
Mean Device Model Training Time(s)	..	1141	..
Mean Procedure Training Time(s)	270	194	28%
Mean Correct Procedure Retention(all)	67%	80%	19%
Mean Correct Retention after 1week	71%	78%	11%
Proportion of 'short-cuts' used	8%	40%	400%
Mean Execution time of retained instructed procedures(s)	20.1	16.8	17%

Based on the results of experiment 1, and using 'think-aloud protocols experiment 2, tested the implication that if subjects are asked to infer procedures, rather than learn them, subjects of the model group should have a decided advantage.

The basic measure of how easy it was to infer a procedure was judged by the number of changes made to the control settings, before the appropriate goal state was achieved. See Table(4).

Table(4) Mean No. of Actions Tried While Inferring Procedures

		Attempt	
Group	1	2	3
Rote	21.8	9.8	8.1
Model	8.6	7.9	..

Kieras and Bovair report powerful differences in the performance of subjects and in the protocol data if a device model is available when inferring procedures for operating the device. They interpret the findings of experiment 2 as strongly suggesting that the facilitation observed in experiment 1 was due to the model subjects being able to infer the procedures.

Finally, and seen as crucial to an area as complex as computer processing, they wished to determine whether other factors, which were implicit in the use of the device model, had affected the results. Apart from the critical 'how it works' information, the model contained greater motivating factors than the rote method and, in addition, carried details about the components and general principles about how the system works. Experiment 3 addressed these points and collected detailed inter-response times during procedure inference. The experiment was a 2x2 factorial design using Fantasy/No Fantasy(fantasy element using Star-Trek style descriptions) and Specific Information/ No Specific Information as Table(5) which reports the results.

Table(5)Mean Number of Actions Tried While Inferring Procedures

Specific Information Condition	Fantasy Condition: No Fantasy	Fantasy	Mean
No Specific Information	24.3	17.7	21.0
Specific Information	6.3	6.3	6.3
Mean	12.3	10.1	11.2

Kieras and Bovair report that the effectiveness of the device model instructions in the first two experiments was not due to either the motivational interest of the fantasy, nor to the how-it-works information about the system components, nor to the general principles underlying the system. Rather the critical how-it-works information in the specific items of system topology that relate the controls to the components and to the possible paths of power flow. They conclude that the three experiments not only show that device model information can have a definite and strong facilitative effect, but also show how it does, and what kind of information is critical for an effective device model for this type of device.

The supporting Borgman(1982) findings confirm the belief that when a user is given conceptual instruction and a mental model is adopted, he/she performs as well as or better than procedurally trained users on routine tasks, better on non-routine tasks i.e. inventive tasks, and are better able to debug problems and escape from errors. If a mental model is defective, whether learned in training or inferred, corresponding performance errors will occur. The conclusion drawn is that systems which are designed around a consistent model should be easier for the user to model, and hence easier to use.

<u>The future of system design</u>

We constantly return to the theme of cause and effect which can be seen in the design principles of more complex machinery of every-day life and is seen to consider human psychology as well as physiology. For example, in the car heater, the control raises temperature when a dial is turned to the right or a lever slides to the right i.e. both indicate 'more' psychology. Similarly, compatibility between concept and symbol, for example water taps are labelled for temperature; cold-blue and hot-red. People appear to make some association with given shapes, colours and patterns of movement more readily than others. It would, clearly, be of advantage to identify these tendencies and take advantage of them in designing the human-computer interface.

For the long term view for computers users, Norman(1982) has already suggested that in an ideal world the system would be constructed and its design based around a conceptual (system image) model. The image of that system(for the user) should be consistent, co-hesive and intelligible, and thus be able to develop a useful model. The instruction manual - all operating and teaching of the system should be consistent with the system image.

The Measurement of Mental Models

As an educator I see my role as encouraging positive change through growth and understanding at all levels. A necessary part of this continuum is the research into ways of making the 'vehicles' of change more effective and the improvement my own understanding of the learning-teaching partnership.

Much of the 'change' promoted in school can be objectively measured in terms of knowledge or skills acquired and the transfer of these attributes to similar but novel situations. However, for research purposes the whole approach to procedure, integrity of data and its measurement is necessarily more critical.

The only real evidence we have about the 'existence' of mental models and their effects is based on information gained directly from subjects and by changes which can be observed and measured from experimental tasks. Patterns of commonality and subject development in these 'elements of mental models' can be sought through analysis and their reliability confirmed or rejected statistically. Lastly, and based on earlier findings, changes in behaviour, as a result of intervention, may also be analysed and measured for statistical reliability.

There are theoretical and empirical problems in establishing measurement of this kind and, in the following study for example, collecting data presented problems and led to methods such as protocol collection being rejected because I believe that data integrity may not have been maintained.

The study and evaluation of drawings in this study is an original method therefore there have been no other appropriate examples on which to base methodology. My work is founded on the belief that subject drawings will be a statement of understanding expressed through the relationship of mental models and the evoked image.

Mental models have been defined above as an understanding about how a mechanistic device works in terms of its internal structure and process. A mental model of the computer is a representation of the how-it-works relationships within the system. With it the user should be able to 'envision' or RUN the model to produce a simulation; it is at this moment of the envisionment that the topological image is required.

The mental model/image relationship is, I believe, interdependent, the evoked image being the necessary 'visual' peg on which to hang the mental model. It is analogous to textual data and a word-processing program. In order to load previously typed text into a computer and present it to the visual display unit requires both the data and the program. The data alone can make no sense because it has no structure. The evoked image gives substance to the implicit understanding of an interactive process and enables the mental simulation.

The ability to produce an accurate mental image of an object is dependent on knowledge held about that object. It is thought to be difficult to produce an image of something not first seen, or if the knowledge or experience of the object is weak then the image may well be poor and inaccurate. The same will apply to the imaged mental model; it has been shown elsewhere that the ability to use inaccurate and often incomplete models is an important characteristic of human reasoning.

The imaged mental models may be poor, in terms of usefulness, firstly because the 'process' is not understood and secondly because the topological image can only be clothed with 'items' that are known to the imager. For example, some of the subjects of the research will rely on their experience or knowledge which includes only the visual input and output process of the computer. Their imaged mental model is incomplete. However, upon this baseline we must add other effects because the mental model is constructed from many sources.

The subjects are likely to be influenced by culture and linguistic metaphor as well as their natural preferences for direct empirical explanation. The most limiting factor for the studied age group is their place on the cognitive development contiuum. It is likely that many of the concrete operational children will be unable to hypothesise about the unknown, hidden processes.

The drawings created by the subjects, however simple, will play an important role in informing on the image-model relationship, the child's understanding and the level of his/her cognitive development.

Summary

This, the final chapter of the first section, has looked at the theoretical and practical faces of the mental model. It appears, in 'theory', that the many benefits offered by the mental model are desirable. In 'practice' it is found that, although mental model generation is a natural process, factors such as language and hidden processes contribute to the production of faulty models which may be unhelpful and manifest themselves in limited, ritualistic user behaviour.

In particular the inferred models or primitives generated by early concrete operational children are locked into an unhelpful mode by virtue of the child's stage of development. As a result they are unlikely to speculate productively about the hidden processes which may be simply ignored. With appropriate experiences, I believe that these 'gaps' can be, at least partially clothed during the period of maturation. For example, it has been shown that novices perform measurably better when they are shown the critical how-it-works information in the specific items of system topology that relate the controls to the components and to the possible paths of power flow.

Although there may be problems with the early, intuitive mental models their value is without question. When mature they can enable the user to develop a more powerful technical vocabulary for describing functional speculation and can contribute to the development of understanding about how programming code controls computer operation; allowing individuals to make inferences, predictions, correct errors, experience by proxy and generally improving performance.

However, the problem should not be viewed in too simplistic terms, because consideration should be given to the possibility of multiple models which are generated from inherent culture and metaphor rather than just a single inferred model of the system. There is little doubt that misconceptions arise from computer-related terms as a result of prior natural language understanding. In general terms, the conclusion is that experience alone is insufficient and that for complex systems with hidden processes, explicit training about the system is required which should include an element of language development.

There are already signs of change and a growing awareness of the importance of the human-computer interface but, until real changes have percolated to the 'chalk-face' in terms of design and appearance of data processing systems we need to modify our aproach to the teaching of information technology to our new, very young population of computer users.

Most of the research involving mental models investigations mentioned above and elsewhere, has been undertaken with adults. The following section of this Thesis pursues the mental models enquiry further during empirical studies carried out with school children.

SECTION TWO

THE PRACTICAL STUDIES

CONTENTS PAGE

SECTION TWO

The Practical Studies

FOREWORD

I have reached the conclusion that if school children of today are to meet the educational criteria for Information Technology they must be taught to be computer literate. I believe that a principled understanding of the relation - ship between learner and computer should be pursued through the concept of mental models.

The previous chapters have indicated a starting point by highlighting expected conditions. The conditions are based on the widely accepted view that peoples' ability to use an interactive device depends in part on their having access to some sort of mental model(Moran 1981) and that generating a mental model is a natural process(Norman 1982).

The next important factor is that the user begins with a general model of how the computer system works in terms of its internal structure and processes(Kieras &Bovair 1984). These inferred models of how the machine functions, are likely to be incorrect and to include implicit, hidden assumptions based on functional evidence. Even if given a conceptual model the user may not translate it to a 'useful' mental model. (Borgman 1982)

Mental models carried by the novice are characteristically inaccurate, incomplete and misleading. They often cause confusion, misconceptions and ritualistic use (Williams,Hollan & Stevens 1983 and Norman 1983). The educational implications lie in the conclusion that, if badly flawed, mental models held by the novice, may impede rather than support learning. Consequently, children may acquire low-level skills but the deeper conceptual under- standing that allows skills to develop and generalise may elude them.

Finally, it has been shown that mental models of the computer system are affected by language, not least because of differences between everyday and esoteric meanings. Distributed and functional models may be promoted through programming and general user terms; some of these will remain and be integrated as 'special case' models, some will fade away naturally and be replaced where appropriate with structural models. However, some globally destructive misconceptions may be fostered during the process as well as 'profitable' misconceptions.

Mental models can improve performance to better that of procedurely trained users - they promote inventiveness and the ability to overcome problems(Borgman 1982). The generation of mental models is a natural process and one may take advantage of it. Also, because a mental model is a combination of an envisionment and its causal attributes, the 'running' or qualitative simulation of a model produces a specific behaviour for the device by giving a chain of causally related events, (deKleer & Brown 1981). The distinction between the intrinsic mechanisms and the correct causal model suggests a possible solution to the long-standing issue of why people find some faults so much harder to troubleshoot than others.

A device model giving system topology and the possible paths of power flow that give the novice knowledge of how a system works helps by enabling the user to infer how to operate the device and have sufficient understanding to enable short-cuts in method to be seen and taken (Kieras & Bovair 1984).

Language is seen to play a part in mental model development and expert programmers are observed to work with and be guided by efficient mental models of how the programming code controls the computer operation. In novices the models are adapted in response to direct instruction and feedback from their own programming and debugging experience.

DiSessa (1986) suggests that functional understanding is not well supported by a structural model. He observes that characteristics of distributed models are akin to learning by prototype and that functional models offer powerful leverage in coming to understand a language; they also help to promote understanding about the way some prewritten code works. He sees a solution through a pattern of function-structure-function type sequences. Where one gradually develops a sense for the invariance, the structure underlying function, through experiences in a broadening set of contexts.

INTRODUCTION

Perhaps the most important responsibility of school based teaching is the development of understanding which, when related to the use of computers, has its fundamental roots in such questions as 'how much and what kinds of understanding are required'. For example, how much understanding does a user need to become a skilled user and what kinds of understanding areneeded to allow the user to transfer skills between systems.

The aim, in the promotion of understanding, should be to avoid 'swamping the novice with information' or, of equal importance, 'confusing by concealment' Riley(1986). I believe that the preferred alternative is the development of methods which enable the identification and analysis of existing beliefs held about the system and its processes.

The very nature of our introduction to the world helps to create a generally supportive learning environment Owen(1986). We are, from our earliest experiences, acquiring and storing the primitives which enable all subsequent learning and development, therefore, the process of understanding the complexities of our environment is necessarily an ego-centric and self-pacing affair. However, when the learner is introduced to a highly developed system such as a computer in which the "compexity underlying the device is glossed in an article of faith which never has to be unpacked" (Owen), the process of gradual acquisition of primitives breaks down.

The denial of the conditions which promote natural development of primitives combined with procedurally taught computer techniques is likely to frustrate the desired development of understanding. In its place, beyond a superficial level, the more likely outcomes are ritualistic behaviour and inflexibility resulting in an inability to cope when the system acts unpredictably. Furthermore, Riley suggests that experience does not diminish the need for understanding; it will continue to be important regardless of acquired experience and the same criteria of understanding should be required to evaluate a user's problem representation.

Normal acquisition of primitives is encouraged by first hand experience and we can learn much by monitoring this learning process with novice users of the computer. The writer has observed, for example, during school-based computer activities that 'other users' in the group play an important and continuously supportive role. A clearly defined inter- active behaviour is observed to develop within the group and includes exchange of knowledge and feedback systems about actions taken during task execution. A similar phenomenon has been observed by O'Malley(1986) and by Bannon(1986) who points out that by focusing exclusively on the human-computer dyad we are missing out on the importance of social support networks which any group of learners naturally creates.

Computer systems , as with many other modern devices, can be described at different levels(Brown,Burton & deKleer(1983)), together with objects appropriate to that level, and their related commands and procedures.Thus, depending on the particular task a user wishes to accomplish, the objects in the user's conceptual model will differ. Understanding something often means that one can follow a chain of cause and effect down to a particular level and at that level, depending on the individual, an article of faith precludes the need to pursue the chain further (David Owen). In an opaque system such as the computer where most of the processes are hidden Brown(1986) argues that conceptual models could be used to achieve the fundamental aim of overcoming "opacity and complexity". Similarly, Norman(1982) suggests that a conceptual model of a system can be thought of as providing a scaffolding upon which to build the bridges across the gulfs.

A Method for Examining the Existing Mental Models

To seek information from our very young learners about their mental models of a computer system presents a formidable task. The concept is new and the existing methods available to do so are few. Larkin and Raynard(1984), for example, say that information processing psychology offers a good way to find answers to questions like "How do students currently understand the task?" and " How does that interfere with their being able to think about it more effectively?". They add that when we need data on mental processes, the steps in a persons thinking while solving a problem, a simple and powerful way to get such data is to simply ask the subject to do the task and to talk aloud about all thoughts that occur to them. These comments are tape recorded and transcribed to form a record called a 'protocol'.

The information to be collected on the user's mental models about a computer system would need to be carried out with a substantial number of novices, preferably without allowing prior discussion among themselves about the exercise. It is, of course, the very nature of young children that as soon as protocol collection was begun, information about the subject and nature of the interviews would be widely spread and discussed. There are at least two other reasons why protocols are considered unsuitable for studies such as this. Firstly, the extreme youth of the subjects and secondly, the nature of the mental models about which information is sought.

The extreme youth of the subjects introduces such factors as:

a. Subject's sensitivity about the purpose of the interview even when careful explanation and reassurance has been given. The exercise may be viewed as a form of assessment or test; in which case it would present a threatening situation resulting in with-held or modified responses.
b. Subject's inexperience and ignorance about interview content could produce a reluctance to say anything that might appear 'silly' or 'wrong'. It could produce a defensive stance leading to 'don't know' or 'don't understand' responses.
c. Subject's limited powers of expression together with inadequate knowledge of appropriate terminology could inhibit dialogue or produce inhibited responses.
d. Subject's verbal description of objects, particularly the very young novice, is unlikely to prove sufficiently full and

areas requiring clarification could easily be interpreted, by the novice, as a 'cue' to expand information in a way which would not otherwise have taken the place.

Factors to be considered because of the nature of mental models about which information is being sought:

a. A protocol is not a complete record of the solvers thoughts nor does it say why the solver does what he/she does. It is likely that the 'hidden processes' would remain untackled unless a great deal of prompting and guiding are employed; an approach considered inappropriate for these studies if integrity is to be maintained.

b. For the novice to explain a computer state or process implies existing knowledge or understanding which cannot be expected or assumed in these studies. The information sought initially may be nebulous; based on beliefs and attitudes held by the novice. It is less a case of what they 'know' but having used the computer for varying periods of time 'how they interpret' the experience in terms of objects and processes required to do the task.

c. Unlike sequential tasks such as solving an arithmetic problem the interview situation does not lend itself well to the promotion of thinking through a non-linear problem.

Similar problems exist for collecting information by written description of an environment or process. A large number of novices are able to carry out the exercise at the same time and the method allows opportunity for working through the problem quietly and unprompted. The novice could, for example, be asked to describe in prose form how he/she imagines the inside of a computer to look or how a particular process might be carried out, but many of the limitations presented by protocol collection also apply here. For example, the novice could be limited by his/her

a. ability to express partly formed or abstract ideas in writing

b. existing vocabulary and terminology

c. understanding of the subject

d. reluctance to reveal lack of knowledge therefore is more likely to say don't know or don't understand, any one of which is likely to inhibit the writing of a natural and imaginative description truly reflecting what the novice thinks or believes.

An alternative method exists which uses one of the earliest and most natural activities carried out by young novices and one which is 'safe' as well as 'enjoyable'; drawing or painting. This method of expression allows the novice to transcend the very real barriers of language, verbal and written, together with problems related to low confidence and self-consciousness about existing knowledge and understanding. By asking the novice to draw a system, or a system process the problem, whilst still an exceedingly daunting one, becomes far more manageable. The task presented becomes more familiar and less threatening so long as respect for the novice's level of interpretation is seen to be maintained.

The Piagetian description of the fourth stage of development in children's drawings achieved at around 8 or 9 years of age is 'visual realism' (Barrett 1983). At this stage the child is said to attempt to represent only what is visible from one particular viewpoint, endeavouring to take into account the principle of perspective and of distancing when producing his drawings. Consequently, the child now eliminates from his drawing those objects which are invisible from his own point of view. Barrett suggests that Piaget views drawing as an activity which is characterised by imitative accommodation, where the child always accommodates and adjusts his own graphic schemes in order to make them represent and imitate reality. Consequently, Piaget construes drawing to be much more closely related to the construction by the child of mental images(which also imitates reality) rather than to the symbolic play of the child (in which the child may quite freely assimilate objects to his own schemes without any concern for their objective characteristics).

The concept of asking children to draw pictures in order to examine attitudes and understanding is not new. Drawings have been used extensively by psychologists, particularly in connection with maladjusted children, in an effort to unravel deep-seated emotional problems. I believe, however, that the method as it is used in the context of this study, is new and it is hoped that some essence of conscious and unconscious thought about computers and how they work are captured in the drawings.

The Experimental Studies: an overview

The practical section is covered in three chapters. Chapter 4 reporting on the pictorial studies, Chapter 5 concerned with changing mental models and Chapter 6 the relationship between language and the mental model. The work is based on the conclusion that if school children of today are to meet the educational criteria for Information Technology they must be taught to be computer literate. To this end a principled understanding of the relationship between learner and computer is pursued through the concept of mental models.

The previous chapters have served to 'sign-post' direction by highlighting expected conditions. These conditions are based on the widely accepted view that peoples' ability to use an interactive device depends in part on their having access to some sort of mental model(Moran 1981) and that generating a mental model is a natural process (Norman 1982).

Chapter 4, the first three studies, is designed to establish the feasibility of using graphical means of collecting data about mental models and to examine the content and nature of those models. Because a mental model is defined as the combination of an envisionment and its causal attributes, the 'running' of it should reveal a chain of causally related events(deKleer & Brown 1981). Therefore, in detail, I wished to establish the type and structure of components included in the model, their universality, level of incidence and evidence of effects exerted by age and/or experience.

The next important factor revealed by the previous section is that the user begins with a general model of how the computer system works in terms of its internal structure and processes(Kieras &Bovair 1984). These inferred models of how the machine functions, are likely to be incorrect and to include implicit, hidden assumptions based on functional evidence. Even given a conceptual model the user may not translate it to a 'useful' mental model, (Borgman 1982).

Mental models carried by the novice are characteristically inaccurate, incomplete and misleading. They often cause confusion, misconceptions and ritualistic use (Williams, Hollan & Stevens 1983 and Norman 1983). The educational implications lie in the conclusion that badly flawed mental models held by the novice may impede rather than support learning. Consequently, children may acquire low-level skills but the deeper conceptual understanding that allows skills to develop and generalise may elude them.

Therefore, Chapter 5 presents two studies which investigate whether 'flawed' models held about the computer memory can be identified. As described elsewhere, misconceptions about storage and 'memory' cause anxiety and ritualistic behaviour in many of the young users as well as longer-term problems for Computer specialists at GCSE. The studies start by looking at a more formal method of collecting data and proceed to examine the concept of 'memory' , i.e. whether it is considered at all , how it is used and what is actually stored. Lastly, based on the assumption that novice mental models are generally incorrect, the possibility of exerting change in a mental model is examined by intervention between pre and post tests.

It has been shown elsewhere that mental models of the computer system are affected by language, not least because of differences between everyday and esoteric meanings. Distributed and functional models may be promoted through programming and general user terms; some of these will remain and be integrated as 'special case' models, some will fade away naturally and be replaced where appropriate with structural models. However, some globally destructive misconceptions may be fostered during the process as well as 'profitable' misconceptions.

It was considered that no investigation of the child's understanding and beliefs about a computer system could be complete without looking at possible ways of measuring some of these effects. Chapter 6 investigates two related areas of language. The first investigation is in three parts. Studies A and B are based on individual 'words' and program "sentences' respectively and concern the perceived relationship between a 'word' and its function. They look at the difference between expert and novice perceived functional categorisation of computer related 'words' and for evidence of age related changes from subject responses. Both parts also examine 'memory' representation with a view to establishing a link between the pictorial and the linguistic evidence.

Without the helpful support of some kind of understanding about the effect of user commands and programming code system behaviour cannot be predicted or problems easily diagnosed and rectified. Part C presents the subject with an incomplete and incorrect piece of programming code enabling examination of novice attempts to analyse and predict outcome.

For the final set of experimental studies the concept of a flawed mental model is revisited with the acknowledgement that information about misconceptions held by novices is significant to the educator only if change, perhaps through diagnostic teaching, is an objective.

A method by which 'understanding' can be shared between young novice and 'expert' has to be recognised and developed. Through communication of understanding, which necessarily involves language, the novice must make implicit understanding explicit and thereby robust; I believe imagery can assist in this process.

The first of the final experiments seeks to establish whether young children are able to recall images of known objects or places to solve a problem. Further, whether they are aware of this use of evoked images and will, without prompting, describe the method used when reporting on how the problem was solved.

The empirical studies are concluded with a final language-imagery study which links prior meaning and language 'imagability'. Prior meaning and degree of imagability of computer terms are believed to exert their own strong effects and constraints on the developing mental model. The investigation looks at the ability of the young novice to report the imagability of commonly used computer 'words' in terms of picture, word or nothing evoking and whether the reports of imagability are modified with age and experience.

CHAPTER 4

PICTORIAL REPRESENTATION OF THE MODEL

The experimental work involving subject drawings developed in three stages. The first, a Pilot study was carried out at Lomond School Helensburgh. It examined the feasibility of collecting data by this method and, in effect, determined whether the method was capable of extracting the type of information that was required from the subjects. By asking the subjects to draw what they thought the computer looked like 'under the lid' I hoped the results would reveal clues about the existence and nature of mental models held about the system.

The pilot study was encouraging in its success as a method of data collection and the data promised to be a good source of information about system component models. The second study was designed and carried out as a replication of the first, on a larger scale and with a wider age-range of pupils on this occasion from City of London Freemen's School. Like the first study the problem was 'open-ended' with the subjects given no direction or clues about what to include in the drawings.

The drawings produced in the second study revealed the existence of a system model containing some common, fundamental system features and illustrating age and experience related changes. The discovery of pictorial attempts to explain 'cause and effect' was unexpected and added a further dimension to the study which led to the question of whether the general system model would become modified or totally transformed if the task changed from an 'open-ended' to a 'constrained' problem situation.

By formalising the third study, also undertaken at City of London Freemen's School, to a specific system processing task, I was able to enforce greater structure and explicitly encourage the subjects to explain hidden data-processes and communication with the system.

A PILOT STUDY: LOMOND SCHOOL

The first 'mental models in picture form' exercise was designed to establish how subjects respond to the unusual task set. They were, in effect, asked to draw their own ideas about how a computer might look inside; requiring them to think about, select and draw the internal components. It was hoped that if the novices were able to tackle what was clearly a difficult task successfully their drawings would reflect something of existing under- standing and beliefs about the computer system.

No list of components was drawn up before completion of the drawings so that naturally emerging elements, which might otherwise be masked by a-priori checklists, could be observed. The objective was to identify fundamental and commonly occurring elements, concepts which would, in themselves, be a statement of novice understanding and beliefs.

For example, if the concepts are limited to consideration of what a computer 'is' then one might expect to find unrelated objects with a low incidence of communication. Alternatively, if the problem solution has a greater global sense, that is, seen as much about what the computer 'does' as what it is, one might expect to find representation of object interaction. The analysis should be able to provide evidence about the nature of the mental model and whether it exists in a pedagogically useable form.

Method

Arrangements were made with the school Art Department for the drawing exercise to be carried out during a normal art session, a deliberate decision enabling experimental tasks to be carried out with the same status as a normal lesson work.

The scene was set for the study by adapting the idea of a current and much loved television program featuring the popular botanist David Bellamy. Bellamy had been appearing in a series during which, with much technological help, he shrank to the size of a small insect. The camera and audience were able to follow his explorations in the undergrowth and ocean-sized puddles as he simulated life as a small insect.

The subjects were asked to recall the series and the advantages and disadvantages of being such a small creature in a large world discussed with them. They were next led to consider a similar idea of shrinking to a size which would allow them to crawl inside the BBC(B) micro-computer through one of the user ports; which were identified for them at the rear of the machine. The question was then posed - "What would you expect to see if you crawled through the port, stood up and looked around inside the computer?" They were asked not to discuss, but to draw what they thought they might see.

The subjects, 38 first year pupils from Lomond School, Helensburgh, were a mixture of boys and girls with an average age of 12 years. They had mixture of previous experience with computers ranging from no 'hands-on' to the home 'enthusiast'. However, most pupils had used a computer during a weekly 80 minute 'activity' session throughout theprevious year and during a computer club. The materials used for the exercise comprised standard A4 drawing paper, coloured pencils and felt tipped pens

Results and Discussion

On completion the drawings were identified by subject and group. From the first stage analysis it was clear that there were a number of common components which could be classified to a number of categories. By far the most significant were Communication-links, Transport, Memory and Input/Output functions which are defined below whilst others exist as clearly labelled items and are self-explanatory.

Communication-links were seen in three distinct modes. First, as part of a 'fantasy' environment illustrating road or rail links with pedestrians and/or vehicles, Drawing Ref(1). Second, the communication-links were entirely non-fantasy 'wire or tube' representations, Drawing Ref(2) and third, a combination of the first two modes,i.e. showing road or rail representations and 'wire-linked' components, Drawing Ref(3).

Transport, seen as a confirmation of the communication-link concept, fell into two main categories, land or air-based vehicles. The land-based included cars, trains, Ram/Rom transport and Monobus as Drawing Ref(4) whilst the air-based included aeroplanes, helicoptors, rockets and 'air-bugs' as seen in Drawing Ref(5).

'Memory', as a classification, proved to be more difficult and necessarily relied more on interpretation than other categories. Included were any objects which were labelled memory, memory bank or bank, see Drawing Ref(6), byte, K-shaped, 32K or similar see Drawing Ref(1), RAM(random access memory), ROM(read only memory), see Drawing Ref(2) or Data store. Some drawings contain more than one of the items listed. Although 'chip' appears as a component in the drawings, lack of evidence indicating that it has 'memory' intent has excluded it from the Memory category.

Input/Output functions appear in four different forms: as a labelled component, see Drawing Ref(7), a graphics facility and Keyboard, Drawing Ref(4)or VDU, Drawing Ref(2)

The raw score incidence tables for the two subject groups may be seen in Appendix(3). Table(6) shows the eleven categories with percentage incidence together with a breakdown showing composition of the first 4 categories

<u>Table(6) Lomond School: Component list with percentage incidence</u>

1. Communication links	94.7%	}	Road/rail	55.26%
		}	Wires	10.5%
		}	Combination	28.94%
2. Transport	81.6%	}	Air	10.6%
		}	Land	71.0%
3. Memory	44.8%	}	Mem.Bank	34.2%
		}	32K	5.3%
		}	Ram/Rom	5.3%
4. In/output	57.9	}	Labelled i/o	28.9%
		}	Keyboard	15.8%
		}	VDU	5.3%
		}	Graphics	7.9%

Table(6) continued

5. Games/programs/wordprocessor	31.6%
6. Chip	34.2%
7. Dump	18.4%
8. Micro	23.7%
9. Data/information	10.5%
10. Centre/repair	5.3%
11. On/off	2.6%

A REPLICATION STUDY

CITY OF LONDON FREEMEN'S SCHOOL

The first drawing exercise allowed the novice free interpretation of the given situation. The only content constraints imposed being those implicit in the chosen environment, that is, the inside of a computer. No particular components were requested nor an a-priori check list prepared so as to avoid the production of a biassed analysis of the drawings.

The results showed that although the subjects were very young, they were able to carry out the task with surprisingly little difficulty. The initial analysis of the drawings produced a list of 11 components, of these it was shown that almost 95% of the drawings contained some form of communication link. It is this particular element however simply represented, combined with the range of other common objects found, that justified further investigations with a view to clarifying the nature of the mental models revealed.

The second drawing, whilst intended initially as a replication exercise on a larger scale, also gave the opportunity to compare novices from a wider age range and from a totally different part of the British Isles; mid

Scotland and the south of England. Interest in the second drawings was particularly directed to 'Communication links', evidence of communication in the form of 'Transport', 'Memory', and 'Input/output' functions, all as defined above. The main objectives were to establish whether system components were replicated in a similar fashion to the first drawings, whether choice or number of components selected were affected by year-band of the subject and, to examine the nature or form of the communication link models observed.

Method

It was arranged for the subjects to undertake the drawing exercise under the supervision of their normal art-teachers Frances, Janet and Jane. The study was explained to and discussed with the staff who showed great interest and gave their enthusiastic support; two critical factors for the success of the exercise.

To avoid cross or inter-group discussion about the exercise all groups started the exercise on the same day. The members of staff involved were given identical written introductions to the work, in order to standardise the information received by the subjects, Appendix(4) and guidelines for the drawing materials, identification and annotation of the objects drawn. These arrangements were all discussed at length during a group meeting before the exercise took place.

Consideration was given to the factor that many of the new population would not know of David Bellamy or of his ability to 'shrink' to the size of an insect, so a suitable extract was selected from Lewis Caroll's story "Alice's Adventures in Wonderland. An extension was added which led from 'Alice' to the requirements of the drawing exercise, see Appendix(4).

The study involved 132 pupils attending The City of London Freemen's School; they were of both sexes and in three age bands:

25 pupils	9-10 years
57 pupils	10-11 years
50 pupils	11-12 years

The first two groups had not received formal computer studies at school before the academic year in which the study fell, but had, between September and December of that current year, received one 40 minute lesson per week in the computer room. Several children also had access to computers at home. During school computer sessions, using BBC(B) computers and, in addition to the use of standard software pachages, pupils learned how to program simple coloured text and graphics; the faster pupils had progressed from straight line drawings using MOVE and DRAW statements to filled and random shapes. All work was carried out using work-sheets enabling pupils to work at their own pace.

The feed-back from the supervising staff on completion of the exercise was positive; the pupils on the whole found it a stimulating experience with the teachers describing pupils as 'very quiet' and 'totally absorbed'. The pupils were also happy, in the main, to label or annotate components where they could although this was not normal art lesson practice. There appeared to be very little copying and the work generally reflected originality.

Materials used for the exercise consisted of standard A4 drawing paper, pens, pencils and felt tipped pens.

Results and discussion

It was clear from the first analysis that, following the trend seen in the Lomond Study, a similar pattern existed in the CLFS drawings. The same components, Communication links, Memory, Input/output, Chip and Transport, were clearly the most frequently used, therefore, an assisting colleague (JE) and I examined the drawings retaining the definitions laid down in the Lomond Study. For each of the five components we drew up tally, or raw score, schedules which were completed for each of the five groups; the presence of 1 indicating the presence of the component. On completion the two sets of scores were compared and any element of disagreement set to a nil score, see Appendix(5)

It was predicted that older subjects would include more components in their drawings than the younger pupils, therefore, an analysis was carried out to establish the incidence in selection of one or more of the five top components. That is, how many subjects had included 1,2,3,4 or all of the five top components and can we expect the older pupils to reliably include more components.

The data was subjected to a One-way Analysis of Variance and yielded $F_{(2,129)}=4.706(p<0.05)$ indicating an increasing inclusion of feature by age. Year Means are seen in Table(7) and Summary to 'F' in Appendix(6).

Table(7): Year Means

M1(first year)=2.2

M2(second year)=2.842

M3(third year)=3

In order to establish where the differences were coming from a Scheffe 'S' test was computed which produced I=0.748(p<0.05), indicating that only the means of Year-1 and Year-3 subjects were reliable although the trend is in the right direction.

The next point considered was the likelihood of any one of the five components being represented in the subject drawings. That is, were there likely to be any differences in the incidence of a particular component or components across the year-bands. Table(8)shows the observed percentage incidence in each year-band for each component, followed by the ranked data, Table(9)

Table(8):Section Five Selected Objects. Percentage for each year. (Appendix 7)

Component	Year 1	Year 2	Year 3
1.Communication-link	100	100	96
2.Transport	20	16	26
3.Memory	36	56	60
4.Chip	24	54	64
5.I/Oput	40	58	54

Table(9):Ranked Data.

Component	Year 1	Year 2	Year 3
1. Communication link	1	1	1
2. Transport	5	5	5
3. Memory	3	3	3
4. Chip	4	4	2
5. I/Output	2	2	4

To establish whether all years produced similar rankings a Friedman 2-Way Analysis of Variance by Ranks was applied to the data in Table(9). This produced Chi Square $X^2_{(r)}=9.867 (p<0.05)$ when K=5, N=3 and H=479 i.e. X^2 showing that there is a reliable agreement across years. See Appendix(7)

The results show that, although a relationship exists between number of components chosen out of five and subject age-band, representing an age-related differential, we can still expect the components to appear in a particular, preferred order; Communication link, I/Output, Memory, Chip and Transport. The singular change is seen in year three, when a position reversal occurs between 'Chip' and 'In/Output'.

It was predicted that the communication-links might vary in form according to the year-band of the subject, for example taking on greater sophistication in the older pupils. Therefore, one of the main objectives of the exercise was to examine the nature of the communication link as defined below. The observed representations were split into three categories:

1. **Disorganised** : this group represents a 'communications' structure which appears to be little more than a 'muddle of wire. Where components are identified, no relationship exists between them and the 'wires', as illustrated by Drawing Ref(8). In this example the subject has included several of one type of component, 'chips', and, in an apparently random fashion, has drawn an overlay of lines which she has annotated 'wires'. There is no attempt to relate the wires to the components. This group is regarded as the simplest, or weakest, form of communication link.

2. **Combination:** this group is regarded as an inter-mediate level to the disorganised and organised groups. It represents a structure with some component-A to component-B connections but with additional lines belonging to the disorganised group. Example Drawing Ref(10) shows specific connections from Memory Bank, lighting supply, channels etc. but in addition, has included 'space filling' lines that appear to have no purpose.

3. **Organised** : this group represents a structure which clearly connects two or more points in a planned fashion. Regarded as the most significant form of communications link observed it could present a sound foundation on which to develop a model based on component-A to component-B interaction. The subject Drawing Ref(9) offers a clear example which includes four components both strongly drawn and strongly connected.

The results for the three categories over the three year bands were:

Table(10): Percentage Results Communication Categories.Appendix(8)

YEAR	disorganised%	Combination%	Organised%	Nil
1	16.0	60.0	24.0	0.0
2	1.8	51.7	43.0	3.5
3	0.0	34.7	65.3	0.0

It was predicted that although the subjects' choice of communication model within each yearband would lack agreement, the ranked pattern of model choice would show a progressive change across the yearbands.

To establish whether a significant difference exists in the type of conduit used in the drawings, a Chi Square X^2 test(1) was carried out. It produced Chi $X^2_{(r)}$ ($p<0.01$) and df =3 for the three year-bands:

1st year = 19.3,

2nd year = 45.59,

3rd year = 58.2, X^2 indicating a significant difference in the mode of 'Communication-link' observed in drawings within each of the three year-bands. Appendix(8)

Given indications that subjects from individual year-bands are not single-minded in their choice of communication-link model, I wanted to establish whether a subject's choice of conduit and his/her particular year-band were related. A second Chi Square X^2 (kxn)test(2) was applied which produced Chi $X^2_{(r)}$=20.83 ($p<0.01$) and df=6. X^2 indicating that the subject's year-band is a significant influence on the choice of Communication-link model. Appendix(8). The means for the three catagories across all subjects were :

Disorgorganised	Combination	Organised
0.039	0.47	0.48

To narrow the field further, a third Chi Square X^2 test(3) was carried out to establish whether a subject choosing one of the two more sophisticated conduit types, Organised or Combination of Organised and Disorganised, is reliably influenced by his/her academic year. It produced Chi $X^2_{(r)}$=4.13 for the year-bands 2 & 3 and Chi $X^2_{(r)}$=7.96 for year-bands 1 & 3, ($p<0.05$) and df=1. X^2 indicating year-band influence.

The results confirm an increasing differential as the positive trend from a majority 'Combination' mode in the first year to a minority in the third reliably links the most sophisticated communications model, 'Organised', with the age and experience of the subject. Therefore, the older the subject the more likely he/she is to include an 'organised' or component-A to component-B mode communication system within their system model.

A STUDY WITH CONSTRAINTS

The search for clues about existing novice mental models has been much advanced by the first two drawing studies. There is clear developmental evidence of subject attempts to explain the perceived complexities of the system. The age-related increase of system detail, inclusion of more common components, greater consideration to the means of input/output and the developing use of 'Component-A to Component-B' communication models, are indeed very promising signs.

However, it was observed that attempts to explain the process of 'data communication ' using 'visible' transportation of characters along wires or tubes ended in the third year with no apparent attempt to develop an alternative method, see Drawing Ref(11).In order to explore this aspect further a third study was implemented. It had greater structure and was designed to encourage all subjects to attempt an explanation about data processing and communication within the computer system.

The two earlier studies had, naturally, raised expectations about the format of any future drawings designed to portray general system conditions. However, it was not known whether this general model would remain the same, be modified or totally transformed when the novices were subjected to task constraints. Whether or not changes occurred experience indicates that each piece of work would be an individual explanation with elements common to other members of the group, and as long as these explanations lead to predictions which match those of the design model they will be valid, Riley(1986).

The next drawing exercise would present a situation in which the subject had input his/her name as data into the computer via the keyboard and could see it output to the VDU(visual display unit). After explanations to the group they were asked to describe, in a drawing, what they thought had happened between input and output.

Given the constraints of the study, it was anticipated that the results would fall into three main categories: firstly, an intermediate step showing a function between keyboard and VDU, secondly, a keyboard and VDU representation with no intermediate step and, thirdly, any other solution or no solution to the problem i.e no keyboard or screen representation and no intermediate step.

The main objectives of the study were to place the subject drawings into one of the three categories outlined, to examine, where appropriate, the nature of the intermediate steps or process, and to examine methods used in explanation of text arrival at the screen.

Method

The exercise began with an opening discussion during which the pupils were asked to recall their work on earlier drawings. I asked each group to imagine themselves programming the computer to print a name on the screen, suggesting they use their own name. The discussion included the programming requirements for printing a name on the screen and a reminder about line numbers, inverted commas and the need to type RUN for the computer to execute the command. The program line, 10 PRINT "CHARLOTTE" was suggested as an example along with commands to produce various colours.

In addition, the subjects were asked to imagine they could watch everything happening from the moment their instruction was typed into the computer until the name appeared on the screen. They were to draw what they thought would be happening during that time and to annotate their drawings where possible. A particular point was made to assure them that no 'right 'or 'wrong' way existed for carrying out the task.

The subjects for this study were 122 pupils from CLFS, of both sexes and in three age bands:

24 pupils	9-10 years
50 pupils	10-11 years
48 pupils	11-12 years

The computer experience of the subjects and the method used to undertake the study were as described above.

Results and discussion

The results, representing beliefs about the system processes, were sorted into three categories: PROCESS MODEL, NO-PROCESS MODEL and NO-SOLUTION defined as below and tally recorded as before, firstly by the writer and then by JE. In this case there was no disagreement in scores and the resulting raw scores are seen in Appendix(9).

PROCESS MODEL: This category covers all model representations which included an intermediate step or process between keyboard-input and VDU-output; it includes those indicating rather than showing both keyboard and VDU. Example Drawing Ref(12) shows a case of indicated rather than illustrated in/output; here the annotated intermediate processes are memory, a problem box and a sorting box. The data flow is also seen to be a 2-way affair.

An alternative example is seen in Drawing Ref(13) where the subject has included keyboard(plus typist's hand) and VDU. The subject shows the front and side-section of the VDU wherein the 'intermediate processes' are taking place, that is, behind the screen rather than inside the computer. We see labelled components: Memory data bank, screen control, mistake check, screen transfer, command clarify, command analyser and signal variable. Well annotated, the subject writes; " I think that when type in the comand it is checked for mistakes, varified and passed. I then think it is transfered to the screen".

NO-PROCESS MODEL. This model representation includes no intermediate step or function between keyboard input and VDU output as, for example, Drawing Ref(14). The subject has clearly featured the keyboard which has been adapted to the sequence of characters to be pressed, annotated " 1. I press here,[keys PRINT] 2. all of these should be pressed next, followed by this [RETURN] and then this will be pressed [RUN]" . The keyboard is linked to the VDU with a coiled wire. The VDU carries the program, RUN and the subsequently printed name.

NO-SOLUTION: Any other solution or no solution to the problem i.e. no keyboard, screen representation or intermediate step. This group of subjects appear to have been unable to represent the problem as set. They tend, as a group, to create patterns which includes the name but bears no identifiable relation to the input - output theme. Example subject Drawing Ref(15) has presented such a drawing which includes a repetition of her name and the word 'print'.

The observed categories frequency for each of the three yearbands is seen in Table(11).

Table(11) System model categories: observed frequencies

Year group	Process	No-Process	No-solution	Totals
1	3	5	16	(24)
2	19	14	17	(50)
3	28	19	1	(48)
Totals	(50)	(38)	(34)	(122)

It was predicted that the process model held by the subject would become more sophisticated with age and experience. That is, the older the subject the more likely he/she will be to include an ' intermediate step' between input and output. Therefore, a Chi Square in kxn table was computed to establish whether a subject's model representation and his/her year-band are reliably related.

It produced Chi Square $X^2_{(r)}=35.57$ ($p<0.01$), see Appendix(10), with X^2 indicating that a significant relationship does exist reliably linking subject year-band to the effectiveness of the subject explanation. That is, the older the subjects the less likely they are to produce a no-solution type model; they are more likely to include an intermediate process to explain how a key pressed causes a character to appear on the VDU. Means across all subjects: Process 0.41 No Process 0.31, No Solution 0.28

The indications of year-related growth led to further analysis of the intermediate functions of the 'Process' Model category. Three distinct groups were identified: 'Memory' Drawing Ref(16), a 'Sorting or Ordering' Drawing Ref(17) process and 'Others'. 'Others' grouping together all those not replicated or not identifiable, for example, unlabled 'boxes' Drawing Ref(18). The breakdown of the process-model category by year is seen in Table(12)

Table(12)Category Process-model : observed frequencies

YEAR	MEMORY	SORT/ORDER	OTHERS	TOTALS
S1	1	0	2	(3)
S2	9	1	9	(19)
S3	18	7	3	(28)
Totals	(28)	(8)	(14)	(50)

To establish whether the intermediate process, Memory, Sort/order or Other, chosen by the subject is related to his/her year-band the data was subjected to a Chi Square in kxn table. It produced Chi $X^2_{(r)}$ =11.3 ($p<0.05$) and df=4. Means across all subjects: Memory 0.56, Sort/order 0.16 and Other 0.28, see Appendix(11). X^2 indicates a relationship between year-band of the subject and the choice of intermediate step. That is, the older the subject the more likely he/she is to explain the hidden system processes between input and output by using a 'Memory' component as the intermediate step.

Finally, the drawings were further analysed with particular reference to explanations of how the output characters 'arrived' at the screen and four distinct groups were revealed.

NO-EXPLANATION: the diagram included no explanation of how the input via the keyboard actually arrived at the screen. A first-year subject example Drawing Ref(19) illustrates the underside of keys with the characters F R E D progressing downwards via separate 'tubes'; no characters are seen beyond this point . The 'tubes' are combined to a single wire leading to the back(hidden) of the VDU on which appears the name FRED. That is, no explanation of arrival at the VDU therefore it remains a 'hidden process'.

MANUAL PLACEMENT: there is no doubt that the task set is a difficult one and this group has overcome the problem by using 'fantasy', figures to place characters onto the screen, using ladders, conveyor belts etc. In example Drawing Ref(20) a second year subject uses such a solution and the drawing shows a conveyor belt transporting a line of computer program, a lift transporting individual characters to a little man who has the task of climbing up and down a ladder. He carries up each individual character which he places onto the VDU.

WIRE- TUBE: wire or tubes used as conduit between computer and VDU but with no evidence of data transmission i.e. the 'stuff' flowing through it. A third year subject Drawing Ref(21) has produced an annotated and 'transparent' keyboard to VDU explanation. She shows separate cables connecting the keyboard extensions which then combine to a single cable "Turns up to the screen as one" and then "hits the screen". Near to the screen the single cable end branches to form five smaller ones, each with one character at its end, carrying the characters SARAH.

TRANSMISSION:evidence of transmission of data including program character, electrical current or impulses along tubes or wires.

For example, subject Drawing Ref(11) has produced a delightful cross-section of computer and VDU which are connected by transparent tube. The viewer is able to see the keyboard characters progressing from beneath the keys, along the tube and up into the VDU where they appear to be expelled from the enlarged mouth of the tube onto the screen.

In contrast subject Drawing Ref(22) uses a combination of wires, transparent tube and "electricity currant". She shows a side-view outline of three keys and the screen of the VDU between which a three stage journey takes place. The first stage, shows each selected character moving along a wire connecting the key pressed to the second stage which is a transparent tube where the characters progress as a group. At this stage they are propelled along the tube by electricity/ radiator heat. On route the characters undergo a sorting process before finally being propelled onto the screen from the end of the tube labelled 'rocket', again, heat fuelled. Each of the drawings were placed into one of the categories from which the following raw score pattern emerged Table(13):

Table(13) Input to Output Model System Category: observed frequencies

Year Group	No-Solution	Manual	Wire/tube	Transmission	Totals
1	13	1	8	2	**24**
2	16	4	18	12	**50**
3	5	9	17	18	**48**
Totals	**34**	**13**	**43**	**32**	**122**
Means	0.28	0.11	0.35	0.26 across all subjects	

It was predicted that the explanation model chosen by the subject would be influenced by his or her age and/or experience, for example, categories 'Wire/Tube' and 'Transmission' would not be the obvious choice for the youngest subjects. The data was subjected to a Chi Square in kxn table to establish whether a reliable relationship exists. It produced Chi Square $X^2_{(r)}$ =20.6 ($p<0.01$) and df=6, Appendix(12). X^2 indicating that the category of model used to explain output arrival on the VDU is reliably influenced by the year-band of the subject. We can, therefore, expect the older pupils not only to attempt an explanation of the hidden process of 'how the characters from the keyboard arrive at the VDU screen' but they are more likely to illustrate the explanation with visible transmission of data.

DISCUSSION

In order to comment on the development of the drawing studies effectively the Pilot and Replication studies are discussed first with the objective of giving a clear shape to the emerging mental model. Next, with the scene set and the assumptions made, the following discussion includes observations of a study affected by imposed constraints and comments on the differences those constraints appear to generate.

The Model

The first two studies were required to address the question of whether a drawing is a viable means of collecting data about mental models of a computer system: whether the subjects are capable of expressing the required information in a drawing and whether the collected data is in a useable form. Implicit in the discussion of 'method' is the requirement that it proves to be better than, not merely an alternative to, other available methods.

This particular method of collecting data requires good organisation and support. It may be difficult to arrange for a large number of subjects to undertake the exercise simultaneously unless the researcher has the whole-hearted support of the relevant departments and the novices. The drawing sessions in these studies were well received by the members of staff involved as well as the subjects who were totally absorbed in the exercise and enjoyed it. In terms of the nature and worth of the data collected, the answers lie firstly in the existence of this Thesis and thereafter, may be judged on its interpretation and evaluation therein.

The first two studies point to the existence of a developing mental model comprising a number of identifiable components; the study involving 132 subjects, for example, identified eleven different categories of component. From the results we are able to predict that, given the unconstrained problem of illustrating beliefs about the 'internal appearance of a computer', the subject drawings will include five particular components which are common to the three age-groups, that is, 1st to 3rd years. The five predicted components are Communication-links, Transport, Memory, Input/Output functions and 'Chip'.

It will be clear that not all subjects will elect to use all five of the listed components but the older subject will use more of the 'top five' components to furnish his model. It has also been shown that, as the number of items in the drawings (from the listed 5) increase, a particular and reliable prefered order is maintained across the three age-bands, giving the rank order as Communication-link, Input/output, Memory, Chip and Transport.

The discovery of the common items and, in particular the identification of Communication-links, Means of Input& Output, and a Memory component is fundamentally important to the whole concept of a system model. Their existence together with increased frequency and maintained rank position maintained with age is seen as fulfilling a prerequisite for future mental model management.

It was envisaged above, that two major alternative system approaches would emerge in the drawings. Firstly, an arrangement of system objects showing no obvious relational or dependancy links between components and, secondly, an arrangement of objects and communication-links indicating an elementary interactive system. The first model reflecting beliefs about what a computer 'is' and the second about what a computer 'does'.

At the time of definition it was not assumed that one or the other method would be present or be promoted, but the results fall clearly into the second category. The concept of communication is fundamental to a system and has been strongly represented here; as a result we would expect, under similar conditions, inclusion to occur in almost 100% of novice drawings. This suggests that the majority of subjects consider the computer procedurally in terms of 'what it does', and therefore as a 'total' system.

The Communication-link was illustrated in three distinct modes: a straight forward road or rail representation, wire or tubes, or a combination of the first two modes. The form chosen in most cases determined the 'sense' of the 'Communication' which again had three distinct forms. At the simplest level, termed 'Disorganised', the communication-link appears to have no pre-determined route. Like a muddle of wire used to 'fill a space'. At the other end of the range, the most sophisticated version termed 'Organised' represents a structure with a 'planned' appearance. It has a communicating 'sense' ; observed to connect and sometimes inter-connect component-A to component-B, and so on.

It is the Organised model which appears to offer the most in terms of future model development. The third group, termed, 'Combination' represents an intermediate stage between the 'Organised' and 'Disorganised' groups. It includes random wires which appear to have no purpose as well as some component-A to component-B links.

As might be expected there was a significant difference in the subjects' choice of 'Communication-link' within each of the three age-bands, that is, all three types were not represented in a uniform way. The across age-band results, however, show a clear age-related developmental pattern. As a result we can predict that the older subject is more likely to move on from the 'Combination' model to the 'Organised' model. The percentage incidence of the Organised model is seen to increase over the three years in contrast to the other two which decrease over the same period. This development is seen to be directed towards and reflect a growing awareness of the inter-dependence of system components.

The 'Transport ' component, most prolific in the Pilot study at 81.6% was reduced to an average of 21% in the three year-band study. Because a disparity does exist between the two studies which, at this stage, cannot be explained, it was felt inappropriate to develop it further in these studies. It is useful to observe, however, that the element does exist and that it helps to confirm the 'intent' of the communication link.

The 'means of transport' used take various forms, for example, cars, trains, aircraft, RAM/ROM cars and Monobugs. Whilst at first sight, cars and RAM/ROM trains might be dismissed as elements of pure fantasy, they are, on further consideration, totally understandable given the limited means available to the novice: apart from annotated drawings, it is the obvious way for them to illustrate 'stuff' in transportation around a complex system. The choice of terminology, i.e. RAM/ROM is also regarded as significant.

In addition, a few subjects, who, having included a keyboard in the drawing have followed the explanation through by showing keyboard characters transported visibly around the system. This they do for example, using rail-trucks, conveyor belts and transparent tubes which appear perfectly natural when viewed as part of the user's mental model of the processing sequence which also contains methods of storing, sorting and modifying the characters. Unwittingly or not, the novice has illustrated the concept of data-transmission by drawing on familiar conceptually-related methods. A further clear illustration of the advantage drawings have over any other method of data collection. An almost 100% appearance of communications together with 'means of transport' can be viewed as an analogy for the concept of data transfer within the system via data buses

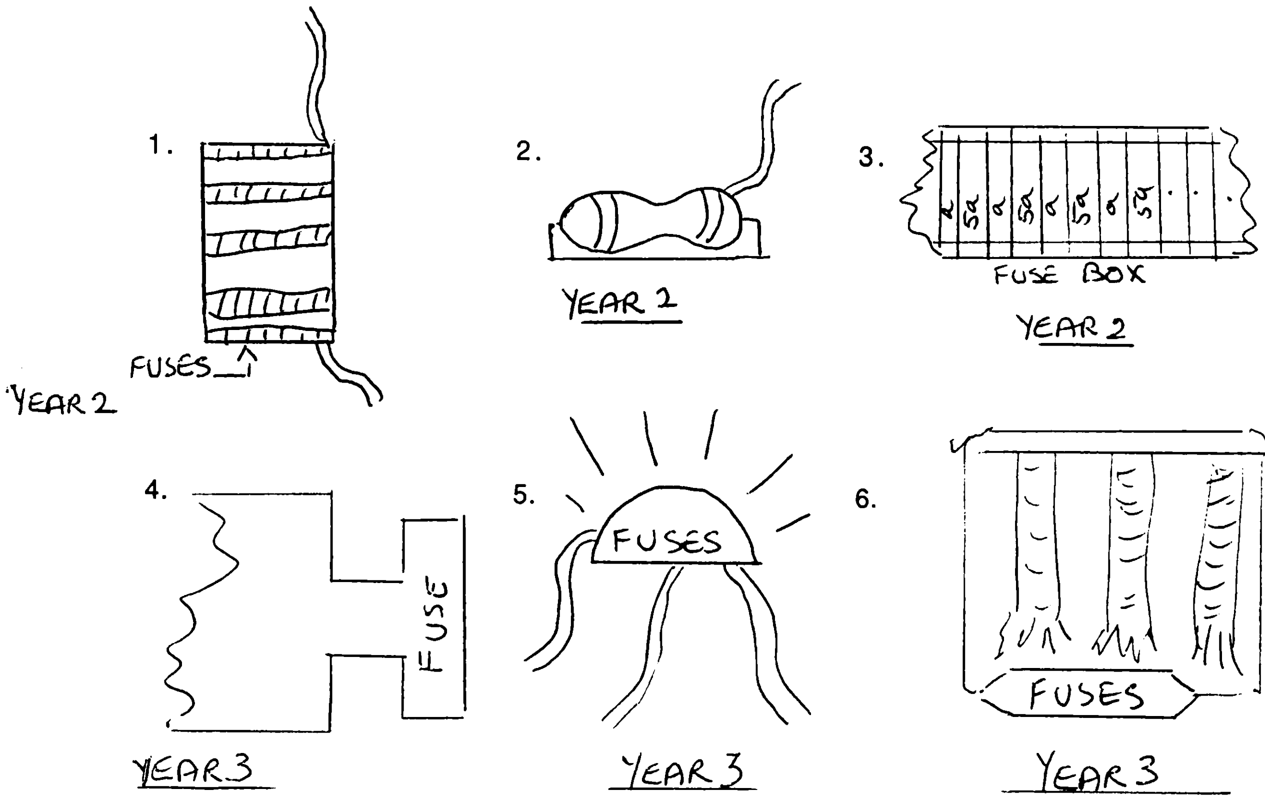
At first sight the 'Memory' element of the drawings appeared to play a smaller role than expected. However, in retrospect, taking into account that Memory is not normally included when young novice are taught to use the computer, it is considered a significant result. The 'Memory' component, as defined above, appeared in 45% of drawings in the Pilot study and was subsequently supported in the larger study at 50.7%. A sustained increase was seen in the use of memory in each of the three age-bands; 24%, 58.9% and 59.4% and it is believed that the results indicate a growing awareness of 'memory' as a property fundamental to the system.

A computer system is composed of three parts Input=>Process=>Output, (feedback assumed) therefore, the appearance of 'means of Input and/or Output' at 54% in the Pilot study, is an important observation. The items included in this group were components specifically labelled 'Input' or 'Output', Keyboards, VDU or 'Internal screens' and 'Graphics' facilities. It seems that the subjects concerned have explained cause and effect; all the more significant because it is not an explicit element of the task specification.

As observed, the exercise was based on the internal appearance of the computer, nevertheless, the keyboard was represented in the second study by 16%, 26.7%, and 28.5%, first to third year drawings, and suggest a growing awareness of 'computer-user' interaction through input. A progression towards an explanation of the 'total system' is shown, for example, by inclusion of a keyboard complete with key extensions within the computer and was observed in 12%, 16.1% and 16.3% of 1st to 3rd year subject drawings.

In general terms, further age-related development is shown in the increasing attention to, and inclusion of, object detail interpreted as an attempt to explain the perceived complexity of the system and its hidden processes. The explanations include 'chips', 'fuses' and 'power' illustrating the way novices draw on known or familiar objects from every-day life. Fuses, for example, were a new detail and made an unexpected appearance in 11.5% of the second drawings reaching a maximum of 14.3% in the second year. They took a variety of forms, including **fuse** dept./air **fus**e/ long wire or tube/ shaped as a resistor/**fuse** boxes from simple rectangle to multi-storage of **fuses**; as shown below in Table (14):

Table(14): Examples of 'Fuse' representation



Similarly, 'power' was observed in 10.5% of the first study drawings followed by 43% in the second study, reaching a maximum again in the second year at 50%.

The items discussed beyond the initial five top items have have been included to 'flesh' out the system bones as perceived by the young novice and will not be carried forward to later studies. However, the information gained about the system model and its components has given an insight into naive perception. It has demonstrated a mental model containing fundamental features of the system which undergo distinct age and experience related changes, some of which are clearly demonstrating cause and effect.

Task Constraints: Observations and Implications

The explanation, above, of data communication illustrated by characters visibly transported within the computer system was unexpected and exciting. However, the decline of its use after the first year without an obvious alternative solution was disappointing. The study which followed was specifically designed to pursue this point and was structured in such a way that each subject was forced to, at least, consider the problem.

The constraints required the subject to explain in graphical form the hidden processes occurring between data input via the keyboard and output via the VDU. To simplify the procedure, only three solution categories were predicted:

Novice drawings in the No-Solution category were so placed because they had avoided addressing the problem as set, ommitting the main observable objects; keyboard and screen representation. It reveals that the ability to explain a complex process such as 'what happens between keyboard and screen' is quite dramatically affected by the age and experience of the novice. The number of pupils not able to attempt a solution to the problem diminished sharply from 66.7% in the first year to 2.1%. in the third.

The No-Process models, 31% of the population, were so placed because they represented a part explanation. The input data had, in effect, become output without further processing thereby leaving the 'hidden processes' unexplained. In many cases the illustrated data varied in form from the name to be printed to the whole program clearly visible in transportation between the two points. This group showed that a steady age/experience related development occurred with first year subjects gaining the lowest result at 21% followed by 28% in the second and 39.6% in the third year.

Subjects in Process Model Category, 41% of the population, showed the ability to recognise a complex process such as what is 'happening' between keyboard and screen with an explanation of the 'hidden processes'. The ability to formulate intermediate steps or functions is, again, seen to increase with age and/or experience. For first year subjects it was, predictably, their lowest category at 12.5%, but, conversely, the highest for the second and third year subjects at 38% and 58.3% respectively.

The Process model, representing 36.3% of the total population and the largest of the three categories, clearly carries the most potential in terms of a working system model. From the results we see that as the subject gets older he can be reliably expected to progress to the Process-model and by doing so will be demonstrating the ability to explain, albeit in his own terms, the hidden processes of a computer system.

Having established a group which has included a process between keyboard and VDU, the actual choice of component was surprisingly narrow. It comprised 'Memory' and 'Sorting or Ordering' as the only clearly recognisable groups and again the choice was shown to be influenced by age-band; the older subjects more likely to use Memory as the intermediate process.

Given that 'Memory' represents the very essence of what a computer system is; upon which its existence depends and 'Symbol Manipulation' represents the very essence of what it does, it is significant to find that both fundamental concepts are featured as intermediate processes in the

observed system models. The former at 56% and the latter 16%. 'Memory' often simply labelled memory or memory box, is shown to increase throughout the three years from one to two thirds. In terms of the whole population the incidence is modest but reveals an upward trend from first to third years at 4.2%,18.0% and 37.5%

The intermediate step of sorting or ordering introduces a new factor not seen as a hidden process in the first year, but observed in 5% of the second and 28% of the third year. The data changes appear to take place after the characters have been keyed-in causing the characters to require re-arrangement by a sorting or ordering processes before they appear on the screen. This process sometimes including the separation or 're-routing' of non-screen characters; the PRINT command, inverted commas and 'Errors' for example.

In order to create pictorial explanations the learner is forced to select the components used and those selected items are likely to reflect the stage of his/her cognitive development, beliefs and understanding. Given a keyboard for input to the system and a VDU for output, the next and further step in the chain of investigation was to examine what understanding the learner has and how that understanding shapes the explanation of the actual transmission of a character onto the monitor. Understanding implies that the user should be able to explain the user action by saying how it contributes to the goal of the sequence of activities of which it is a part, (Clayton Lewis 1986).

The model of 'little men' transferring characters manually to the screen was low in incidence but, surprisingly, increased with age. It is thought probable that the increase in Manual placement of the characters onto the VDU by little people using ladders, conveyor belts etc., was due to a more active attempt to explain rather than ignore the problem. Conversely, the 'No-Explanation' model reduced with age from 52% in the first year to 9% in the third. The closest relationship across the three age-bands was seen in the explanation using 'wire or tube' as a communication-link. By this method the user has, as in the No Explanation category, avoided a description of how the characters actually arrive on the screen. In the Transmission model, transmission of data is shown by the use of characters, current or impulses, passing through wires or tubes to the screen. The incidence increasing from 12% in the first year to 38% in the third, giving the highest category percentage for this, undoubtedly, the most sophisticated model.

Four levels of explanation were identified for the arrival of text at the VDU after input and again indications are that the level chosen is related to the year-band of the subject. A total of 61% were identified as using the 2 categories ' a wire or tube' either with or without transmission of data to describe the arrival of text on the screen. The category wire or tube with evidence of transmission of data increased over the 3 years 10%,30% and

35% respectively. The results show that the youngest subjects are the group least able to explain the steps of transmitting a keyboard character to VDU, at 54% with their next largest category using a wire or tube leading directly from keyboard to VDU at 32%.

If we set aside the age-related 27.8% of subjects(67%-yr1,34%-yr2 and 2%-yr3) who were unable to address the problem, the remainder, almost without exception dealt specifically with the task in hand. The contrast is seen between the 'open-ended' task where the system model is seen in very general terms and the constrained task where 'objects' seen as superfluous to the task are stripped away.

For ease of identification during discussion the drawings from the second, unconstrained, study are encoded drawing(O) and the constrained as drawing(C), effectively identifying the study as 'Open-ended' or 'Constrained'. The results above show that the transition between the Drawing(O) and Drawing(C) involves a large step in terms of cognitive development, therefore, a cross year sample comparison illustrates some particular differences.

It will be recalled that Drawing(O) was a general internal view of the computer whilst Drawing(C) was an attempt to explain the transmission of a keyboard character to the VDU. One comparison is made for each of the three years.

E.D is a first year subject and her work is seen as example Drawing Ref(23&24). Her drawing(O[23]) contains very little fantasy and includes detail of communication between objects where the objects include keyboard character stores and an explanation of the BBC(B) colour mode system as well as screen and scale detail. Her drawing(C[24]) bears no resemblance to drawing(O). The system developed in the first drawing has broken down completely with the result that the system has become 'non-computerised'; an example of no-solution

L.M. is from the second year and her work is seen as example Drawing Ref(25&16). Drawing(O[25]) contains a large element of fantasy. It illustrates a cross-section of a keyboard giving three layers visible beneath the keys. Two keys are being pressed and there are 'letters going through the wires' which are 'connection wires' disappearing into the ground; both points annotated. There also appears to be an internal grid to show which keys are pressed and an internal computer screen. Fantasy elements include 'telebugs' and 'telebug dogs', stairs, ladders and two doors marked toilets. Drawing(C[16]), again, is in complete contrast; it shows a cross-section of keyboard and VDU, is 'technical' in appearance, with no obvious fantasy elements. All the intermediate processing is seen to be carried out inside the VDU including memory, checkers and a collection of boxes labelled 'minds' which appear to be responsible for particular keyboard characters. There also appears to be a direct route from the main memory to the screen.

M.G. represents the third year with a further contrasting pair of drawings, example Drawing Ref(26&11). Drawing(O[26]) is a firm 'no nonsense' diagram with main socket and power switches, communicating leads and 4 main components; Control processing unit, Computer memory system (looking like a tape cassette), data finding system and a set of expansion chips. In contrast his drawing(C[11]) whilst still a strong representation is more 'artistic'. It shows a cross-section of keyboard to VDU with a direct tube-like link between the two. There is no evidence of the internal 'computer' in the keyboard cross-section: the keyboard characters are seen tumbling along the tube from underneath the keyboard, out through the connecting wire and up into the VDU where 10 PRINT "M is seen to be propelled from the tube onto the screen.

As can be seen from the sample of compared drawings, drawings(C) have become 'task' orientated, all components superfluous to the needs of the task are largely lost as were many of the elements of 'fantasy' used in drawings(O). The factor of task specificity implies that the concept of a single 'mental model' of a system does not exist. The implications, therefore, for the design and modification of system models, if further research proves contextual models to be a constant factor, are far reaching.

One problem area for the over-simplified model which may have 'vague assumptions' applied to it, by the novice anyway, is lack of correct information about the system should it not work as predicted. It is thought that there is probably a connection between explanations such as those shown in the drawings and the sort of explanations generated as a result of the system reacting unpredictably.

The devised explanations will tend to make the effects of errors seem reasonable and acceptable to the novice; as a result they will sometimes continue working without any attempt to correct errors since they have satisfactorily explained them away. Such activities suggest that explanations may be an important mediating structure in learning to use a system (Clayton Lewis) and therefore must be taken into account in the system model development.

The 'explanation' indicates how their actions contribute to the goal of the exercise, and indicates what has caused the system to respond in the way it does. Lewis states that understanding how these explanations are built up is important for the following reasons:

(a). They appear to play a role in failure to detect errors. Learners are able to construct a satisfactory(to them) explanation of events that actually are reflections of serious problems(Norman1984,1986)

(b). If we could control or anticipate the workings of the explanation process we could reduce the incidence of error.

(c). Some sequences of events are easier to explain or require less knowledge to explain than others. If we understood this we could design systems that are easier to understand.

(d). Explanation building probably involves some top-down processing, in which user ideas of how goals might be achieved includes the explanations they build. Violated expectations could be avoided.

(e). Explanation requires selection of what is to be explained from the whole event. Knowing how selection is done could enable systems to be designed which make explanation building easier and more accurate.

The drawings themselves generally become 'task specific' when the subject is asked to consider a particular system process and explanation of how the computer carries out a particular task must be dependent on the mental model carried by the explainer. The mental model of how the system functions relative to both its constituent parts and a given task provides the most stable and robust basis for such an understanding.

All the points noted above serve to highlight the unmatchable advantages of using novice drawings or paintings to identify and explore existing mental models. I believe that, by using an alternative method of data collection, at worst both factors would have been lost or at best they would have been modified as a result of verbal or written discourse.

In the following chapter the investigation extends to a further level of novice understanding about the hidden computer processes with particular reference to memory. Because the concept of system memory and its role is weak, the aim of chapter five is to establish whether, given appropriate experience, useful modifications can be affected on the initial mental model as demonstrated by the novice. To meet the new requirements of pre- and post-test data collection it was, at this stage, necessary to modify the concept of 'free drawing' as a method of data collection.

CHAPTER 5

CHANGES IN MENTAL MODELS ABOUT THE NATURE OF MEMORY

This chapter pursues the mental model investigation from the perspective of the misconception or 'flawed' model. It was the discussions of DuBoulay & O'Shea(1976) and DuBoulay, O'Shea & Monk (1980) on Black Box versus Glass Box theories, that led me to consider a further method of investigating changes in mental models.

Having established in Chapter 4 that almost 100% of the observed user models include an element of communications and that an age or experience related use of memory as a system object exists, the questions I wished next to consider were:

Given a formally structured problem environment with a specific, minimum subset of components, how would

1. communication and memory, as system objects, be transferred to, and translated within, the system model and
2. How would the response to the first point be modified if practical work was undertaken which gave the subject suitable experience.

For example, specific experience in the use of

addressed memory locations for data storage within a computer system allowing the contents of these locations to be checked and altered.

The conditions required for such a study would need to include :

a). a set of addressed memory cells which would, at best, be a complete set rather than a subset.

b). a method of 'viewing' several memory cells at a time or means of stepping sequentially through a set of cells as well as 'jumping' to specified addresses.

c). a method by which data could be allocated to specific memory locations, altered and checked.

Desmond: Changes in Mental Models

At this point Dr. Simon Garrod introduced me to DESMOND, see Appendix(13) for specification, a simple, single-chip computer without a 'lid' designed for the Micros in Schools projects by The Open University.

Desmond's processor architecture included an accumulator, program counter, two flags and managed 100 locations of user memory. This simple computer fulfilled the above requirements with points b and c achieved by using Desmond's small subset of assembly language. Two computers were purchased and after an initial period of familiarisation with the language and functions it was introduced to a group children.

The small group of pupils involved in the study were 11 first and second year pupils, between 11 and 12 years of age, attending Lomond School, Helensburgh. Previous experience of the subjects included 1 term in the use of simple BASIC programming with the BBC(B) micro-computer whilst working at their own pace from prepared work sheets. All pupils had some elementary experience of using the computer to solve simple arithmetic problems but none had either seen or had previous experience of DESMOND before the exercise.

Method

The main objectives of the study were to examine and evaluate the way novices handle the concept of data storage before and after relevant practical experience. In effect, to establish whether the concept of data retrieval exists. Whether memory is initially envisaged as a 'space', like a piece of paper onto which characters can be written anywhere on its surface, or more sophisticated storage ideas exist. If the former is the case, it will be necessary to establish whether practical experience, which includes data retrieval, influences the way memory space is modelled by the novice.

As outlined below, two identical tasks were given to the subjects, forming a pre and post test, and were separated by a related practical exercise. The objective was to examine the effect, if any, of the practical work on the pretest performance.

For the pre-test the subjects were given a set of work sheets on which were drawn empty rectangular drawing response frames measuring 18x12cm. The size was fixed and designed to standardise the size of the drawings, see sample Appendix(14). An instruction sheet was also provided for reference which repeated the initial, verbal instructions, Appendix(15)

The subjects were interviewed in pairs and the exercise explained to them before they were offered the opportunity to withdraw from the exercise should they wish, but no subject felt unable to continue.

Each pair of subjects was asked to consider a computer carrying out the following operations:

a. Input and store the number 5

b. Input and store the number 8

c. Add the two stored numbers together

d. Store the result of the addition

Having described the task to the subjects, they were then asked to imagine they could see into the computer and watch the job being done. They were asked to draw diagrams of how they thought the computer might look and change during the task. They were asked to use one frame for each step, with no restrictions put on the number of frames they could use.

In addition to the standardising effect of the drawing 'frames', a further constraint was applied which restricted the minimum set of objects to three; a keyboard, memory, and the means of getting from one component to another if required. Any additional 'part or parts' could be added so long as it/they were clearly labelled.

The instructions were given twice, the first time verbally. A subject ready to start the exercise was seated separately in a quieter corner of the classroom, given instruction and work-sheets and were allowed as much of the eighty minute lesson as they required to complete the task.

On completion of the pre-test the pupils were introduced to DESMOND and the use of mnemonic code(assembly language). The practical work was designed to give specific experience of the tasks referred to in the pretest and included:

STORING A NUMBER at a memory location chosen by the subject
CHANGING A NUMBER STORED in a memory location
ADDING STORED NUMBERS using the accumulator
STORING THE PROGRAM which enables the computer to carry out the task
STORING THE RESULT OF THE ADDITION at a third address
USING MNEMONIC COMMANDS; LDA/ADD/STA etc.

All the practical experience was carried out using Mnemonic code, as in sample program below, that is, memory locations indicated by code only. They appeared in the form 'STA 23', for example, representing the instruction 'Store the contents of the accumulator in memory location 23'.

(1) SAMPLE INTERMEDIATE EXPERIENCE PROGRAM IN MNEMONIC CODE FOR USE WITH DESMOND

BYTE	000	
ADDRESS	000	
00	LDI	000
02	STA	103
04	LDI	001
06	STA	103
08	JSR	215
10	JMP	000

At no time before, after or during the practical experience was reference made to conceptual, visual or concrete memory location representation.

On completion of the practical exercises, the pre-test instruction and worksheets were repeated exactly to form the post-test.

Discussion

In general the drawings were task specific with very little addition to the given subset of objects, see Drawings(27 to 45). James included ROM, RAM and micro-chips and, along with Chan and Wayne, he also followed the idea of A to B type communications with each character having a direct link with memory; each of the numbers 'saved' having its own route to memory. This is thought to be further evidence of developing data consideration as it enters the hidden process section, i.e. the need to sort or re-arrange data order. The commonest form, however, was a single wire or tube as, for example, drawn by Lee.

Some graphical interest was lost between pre and post test. For example the characters visible in transit, see Wayne pre-test, were lost in the post-test because he has drawn the relevant objects from Desmond thus losing the conduits and much of the explanation used in the pre-test. Claire's representation in the pre-test reflected prior knowledge; she included an intermediate step which translated data into computer language and the numbers are stored in binary. In the post-test the objects had been reduced to a circuit of instructions, for example, enter what to do, finalised, selected addresses, finalised, address wanted, finalised etc. again losing the original essence of the explanation.

The visible communication of characters through the system was observed in three first year and one second year pre-test drawings reduced to one first year and one second year in the post-test.

The way memory was used changed in its most basic form, i.e. the use of a separate cell or box for each item of data stored, from one first year and one second year drawing in the pre-test to two first year and three second year post-test drawings. At the next level, addressed memory cells or boxes only 1 second year pre-test drawing was found to contain this form of storage and this figure was modified to one first year and one second year representations in the post-test drawings. These particular changes, visual/concrete space to a visual/concrete partitioned set of memory locations, are small but significant because they were achieved after a short coded memory locations experience.

Finally, changes in the way the sum of the two numbers, was stored. Storing numbers in the form, 5,8, and 8+5 or 8+5=13 was observed in three first year and one second year pre-test drawings which reduced to one first year in the post-test. Data storage in the form 8+5 and 8+5=13 is used commonly by pupils working with Basic in Computer Studies where variables, a common source of problems, are used e.g. A=5, B=8 and C=A+B would be interpreted by the pupil as location A contains 5, location B contains 8 and location C contains A+B or 5+8 rather than 13. Of the pre-test drawings one second year stored 5,8,13 increasing to two first year and 2 second year drawings in the post-test.

Examples of particular changes:

(i) JASON PRE-test already using cells A, B and C to store the numbers 5,8,13 gleaned from programming in BASIC i.e. LET A=5,B=8 etc.only designated cells shown,

JASON POST-test using numerically labelled storage cells in stead of A,B & C. More than designated cells shown. Drawings(27&28)

(ii)CHAN PRE-test using a circular shape labelled memory disc giving a 'space' for storage rather than allocated spaces - 5+8 is seen in store POST-test the disc space is now partitioned 5|2|8|15 where 15 is the stored sum of the other stored numbers. See Drawings(29&30)

(iii)LEE PRE-test stores 5+8=13 in a shape

POST-test rows sectioned memory space is seen with sequential numerical labels along the top. See drawing(31&32)

Although working with Desmond was a novelty enjoyed by most of the novices several of the subjects experienced difficulties, in particular;

a. the membrane keyboard was very different to the standard keyboard, if the pressure was not correct i.e.not firm enough in most cases, the intended key press was not executed.

b. if repeated presses were required such as when 'stepping' through the addressed memory locations, the result of the exercise was wrong because an incorrect number of presses had been executed as in (a) or simply through losing count after repeated presses.

c. as a result of (a) and (b) comparatively long assembly language programs could be keyed in and need repeating

d. Desmond as a computer proves to be rather too small in use particularly when the control switches are being programmed and tested and too complex in appearance.

Conclusions

The DESMOND exercise was halted due to lack of time and resources; I needed to find a method of investigation of the same area but on a larger scale. Important points had been gained from the exercise, however, which served to confirm the belief that young children are willing and able to undertake more complex tasks than is generally expected from them. These points were reflected in their enthusiastic response to:

a. DESMOND

b. A MNEMONIC LANGUAGE

c. PROBLEM SOLVING in terms of diagrams to represent situations/procedures in the computer.

The study confirmed that novices are unaware of the need for system or application programs storage and that most children imagine the 'memory space' of a computer system to be a formless space where 'numbers' are stored. It also confirms that the existing novice model for the latter is capable of modification through experience with computer instruction code which explicitly uses 'separate' memory location identification. It is also seen that in addition to 'how' memory was used the 'what' was stored is also capable of change, thereby beginning the move away from arithmetic formulae storage towards storage of the result of the arithmetic calculation.

In conclusion it should be noted that by using a Desmond type model, which possesses a simplified system but complex appearance, does nothing on its own to explain the hidden processes. The exercise serves to remind us that alternative, even less helpful system objects and conduits can be generated in the system model as easily, if not more easily, than the desired attributes.

Modest evidence of model modification was achieved after a comparatively short and simple task experience allowing the confident prediction that more significant changes could be achieved with the use of appropriately designed conceptual models. The investigation which follows takes up the same objectives as the present one whilst extending its depth and scale.

BBC Network Study

Introduction

It was seen in Chapter 4 that when novices were asked to explain the computer system process between a key press and the appearance of the pressed character on the VDU, 41% of the novices included an intermediate or processing step in their explanations. Of the intermediate steps 56% of the subjects introduced memory in some form. In order to develop a conceptual model which is able to promote greater understanding of the fundamental, underlying role of system memory, more information is needed about how memory is initially modelled by the novice.

The Desmond study produced some encouraging indicators for changes occurring in mental models of computer memory and data storage but, because the study was limited in terms of its scale, that is, too few machines available, the search continued for an alternative method to pursue the same objectives.

The network of BBC(B) microcomputers was next considered for the task. It offered scope for a larger scale study, twenty computers were available, with the added advantage that the subjects were familar with the equipment. The practical experience requirements as detailed for the DESMOND study remained the same for this investigation, i.e.,

a). data to be stored at specifically addressed memory locations,

b). a means of checking the contents of addressed location and

c). a means of altering the contents of addressed locations.

the difference being that the mnemonic code used would be BBC micro-computer Assembly language rather than the DESMOND subset.

On the basis of experience gained so far, two particular factors were expected to emerge. As seen elsewhere in this thesis, the novice has tended not to associate the computer memory with any process carried out on the system unless it involved a deliberate intention by instruction to do so. For example, data saved to or loaded from disk, accessing a disk catalogue or the use of such commands as LIST, NEW or OLD etc. Therefore, the first anticipated point to emerge from the study is that the computer program, which is stored and in use during task progress, will not feature in the mental model of memory before, during or after its execution.

Secondly, the small study carried out with addressed memory locations on DESMOND suggested that a growing number of novices will, after exposure to relevant practical work using addressed memory locations, see DESMOND method, transfer this knowledge to a subsequent related problem. The required information about what and how data is stored could be gained by asking the novices to write down what they thought was stored in the computer's memory at particular points during the system processing, thereby allowing the researcher to analyse its form. Thus the Desmond exercise idea was extended with standard system objects, or models, supplied in the data capture form as described below.

The series of 18 x12 cm empty frames used for novice drawings in the Desmond exercise allowed individual freedom in problem interpretation. However, for the data capture forms to assist in the analysis of exactly what and how data was being 'stored' in system memory greater structure was required to standardise the 'explanations'.

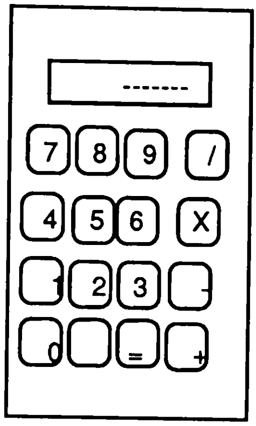
The form of the object models used in the study was developed directly from methods used by Mayer(1982), who similarly wished to examine particular conceptual models held by subjects, undergraduates, in studies based on the use of electronic calculators. Mayer investigated methods of teaching novices to use handheld calculators; both line and stack models. As a result he produced two forms of data capture sheet one for line and the other for stack models, see Illustrations(1 to 3) below.

The stack model was of greatest interest because it contained four registers, the display register, X and Y registers and the operator register, thereby matching fairly closely the needs of the following study. Of the four registers only the display register was visible from 'outside', matching the VDU of the computer system; the others being hidden processes. In his study, Mayer's subjects were

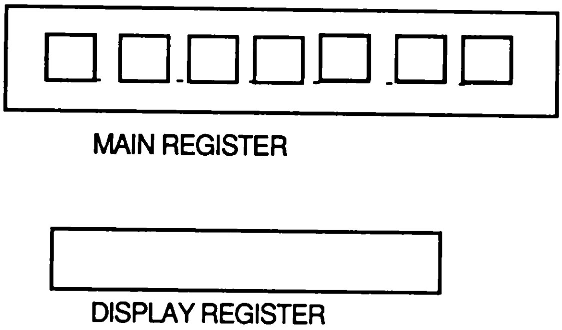
MAYER R.E.(1982)

ILLUSTRATIONS (1 TO 3)

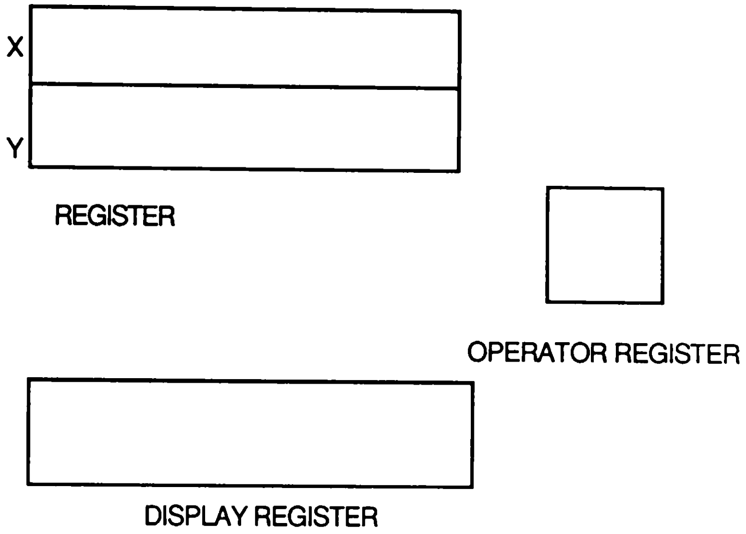
1. Four Function Calculator:



2. LINE MODEL



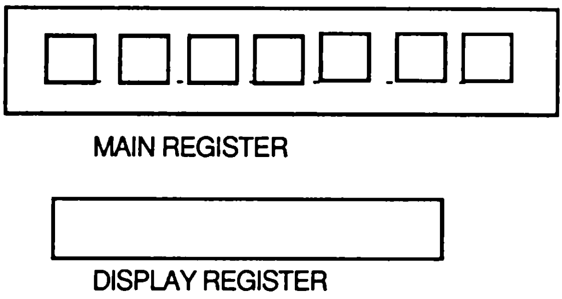
3. STACK MODEL



EXAMPLE OF PROBLEM FOR LINE, STACK AND NO MODEL GROUPS:

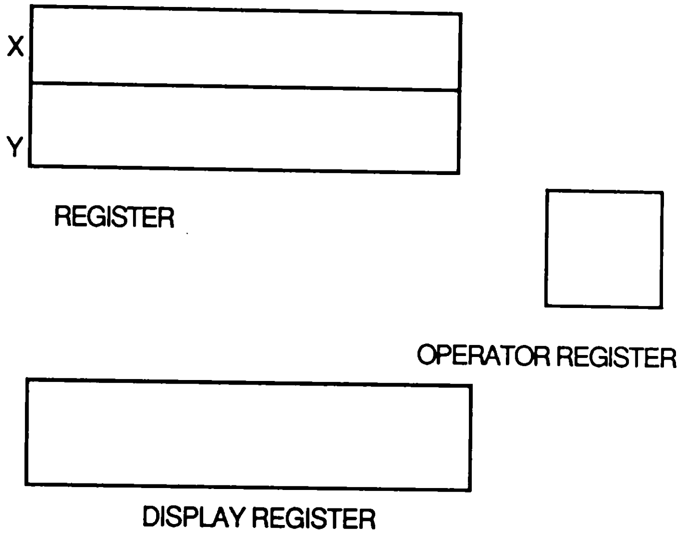
<u>Line Group:</u>

2 + + =



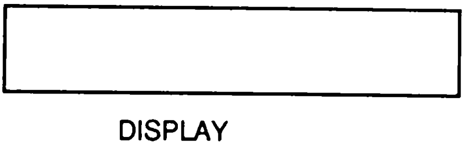
Stack Group:

2 + + =



No Model Group:

2 + + =



given the 'rules' of the registers and statements such as 2+2 and 2+2= which they were asked to relate to the four registers.

Whilst the present study conditions were to differ from Mayer's in its use of computer rather than calculator concepts and by having subjects from a much younger age range than the original study of undergraduates, it did suggest the idea for presenting four system objects on data capture sheets.

The model was designed as a four object 'system' using an Accumulator, Arithmetic Unit, Memory and VDU. Four objects were chosen deliberately to lessen the focus of attention which would, otherwise, be on the key areas, 'memory', the hidden process, and the VDU , the 'observable'. The resulting data capture sheet, see Appendix(16), consisted of five sets of four system object boxes, one set for each of the steps in the program, which would be required to show the changes occurring within each system object at each of the following five stages of the exercise:

i. After the program has been input but not executed

ii. After storing the first number in memory

iii. After storing the second number in memory

iv. After adding the two stored numbers

v. After storing the answer to the addition(iv) in memory

Method

The subjects, 81 pupils, girls and boys, attending Lomond School, Helensburgh were from three age bands; 27 first year, 11 second year and 43 third year and the exercise undertaken as part of their normal computer studies lessons.

The main objectives of the study were to observe the extent to which memory is used during several program processing stages and to evaluate any changes which take place between the pre-test and post-test stages as a result of relevant experience in the proposed 'key areas'.

As outlined below, two identical tasks were undertaken by the subjects, forming a pre and post test, which were separated by a related practical exercise. The objective was to examine the effect, if any, of the practical work on the pretest performance.

For the pre-test the subjects were given two sheets, the first a written description for each of the four system objects to use as reference during the exercise, Appendix(17) and the second, a task sheet, containing five sets of four object boxes, one set for each of the five program steps, as detailed above and Appendix(16). The five program steps were identical to those used in the DESMOND exercise.

The four system objects, Accumulator, Arithmetic Unit, Memory and VDU, and their functions were outlined verbally to the subjects and, as a result, a general question and answer situation arose. This session was considered necessary and important; satisfying general interest, curiosity and helping to dispel such anxieties as "is this a test?". It was stressed that there was no right or wrong way to complete the task-sheet because the object boxes were only helping to describe what we thought might be happening to the data at particular stages in the programming exercise. However, it was not thought that the discussion in any way enhanced or adversely affected the actual content of subject responses given during the exercise.

The subjects, all of whom had previous experience with simple arithmetic problems on the computer, were asked to record the changes they thought would occur in each of the system objects, i.e. the Accumulator, Arithmetic unit, Memory and VDU, during the processing of a program.

The program, written in Assembly language, instructed storage of the number 15 at address 'a', the number 8 at address 'b', and the sum of the two stored numbers at address 'c'. The subjects were required to think through the input and execution stages of the exercise before writing down their answers.

The following is the subset of assembly language used by the subjects for data manipulation with addressed memory locations during the pretest :

Code	Meaning
1. ?a =15	store the number15 at address a,
2. ?b= 8	store the number 8 at address b
3. ?c=?a+?b	sum contents of addresses a & b and store the result of summation at address c.

Following the pre-test the subjects were given the opportunity, during two separate 40 minute periods, to carry out practical work on their own and to experiment as they wished using the assembly code.

Two coding examples follow, (1)an explanation of the practical experience and (2) a sample exercise:

(1) Altering memory values-symbol explanation

[? used to examine a memory location is equivalent to a 'PEEK' statement on other micros]

[? used to alter a memory location is equivalent to a 'POKE' statement on other micros]

[#means what follows is an ordinary number rather than a memory location]

[To make the memory location **70** contain the decimal number **15** it is necessary to type ?70=15]

(2) A sample exercise

P. ALLOWS YOU TO EXAMINE THE CONTENTS OF A MEMORY LOCATION. TYPE IN THE FOLLOWING ASSEMBLY CODE AND WRITE DOWN THE NUMBERS WHICH ARE PRINTED ON THE SCREEN FOR EACH OF THE FIVE **P.** CODES

```
LDA  #8      |LOAD or put the number 8 into the accumulator
STA  &250    |store the contents of the accumulator in memory location 250
LDA  #5      |LOAD or put the number 5 into the accumulator
STA  &260    |store the contents of the accumulator in memory location 260
P.   ?250    |check the contents of memory location 250
P.   ?260    |check the contents of memory location 260
LDA  &260    |LOAD or put the contents of memory location 260 into the
              accumulator
ADD  &250    |add the contents of the memory location 250 to the contents of
              the accumulator
STA  &270    |store the contents of the accumulator in memory location 270
P. ?250      |check the contents of memory location 250
P. ?260      |check the contents of memory location 260
P. ?270      |check the contents of memory location 270
```

REPEAT THE EXERCISE AGAIN BUT THIS TIME YOU CHOOSE THE TWO NUMBERS TO BE STORED AND ADDED.

On completion of the practical exercises, the pretest instuctions and worksheets were repeated exactly to form the post-test.

In general, the exercise was seen to be enjoyed by the majority of pupils, serving to confirm earlier observations that young children enjoy the element of 'secret codes' implicit in the use of assembly language. However, programs should, for longer term use, be developed with the use of 'sensible' variables.

It was also concluded, as a result of the exercise, that children are capable of undertaking tasks which are far more complex than we, as educators, usually explore; the limiting factor often being the way they are explored.

Results and discussion

The first test was undertaken to illustrate and investigate the empty memory phenomenon which is noted particularly in the first of the five stages, that is, when the program had been input but not executed.

At each of the five program steps in the pre and post-tests a score of 1 was allocated if the memory box was empty and 0 when at least one item was recorded. See Appendix(18) for raw scores.

The aim of the first test was to determine whether a relationship existed between the year of the subject and the ranked scores. That is, can we expect the proportion of 'memory empty' responses at each of the five stages of the program to be similar regardless of the subject age band. The scores recorded at each of the five stages for each of the three agebands were ranked from 1 to 5 in ascending numerical order i.e. the fewest 'empty memory' predictions, in first position.

A Kendall's Coefficient of Concordance 'w' was then applied to the pre and post-tests separately to establish similarity of response across the three year bands. They produced, at pre-test, $W=0.816092$ with significance found through S at 71($p<0.05$) $k=3$ and $N=5$, and, at post-test, $W=0.793$ with significance of 'W' through S at 69 ($p<0.05$), see Appendix(19). The results indicating a reliable relationship of year-band on pattern of responses at each of the five memory stages before and after relevant practical experience.

Table(15), below, shows the means of the ranked data at each of the five stages at pre and post-test.

Table(15):Means table, ranked data, memory.

	stage 1	stage 2	stage 3	stage 4	stage 5
Pre-test	5	3.5	2.5	2.8	1.2
Post-test	5	3.8	1.8	2.5	1.8

It will be recalled from the foregoing that a 'zero' score indicates that data is present in system memory whilst 'one' indicates that memory has been left empty; a low score being seen as more sophisticated than a high one. The results show that the highest scores are directed to stage one, that is, when the program has been input but not executed, thereby confirming the belief that subjects are more likely to consider explicit memory use, for example, than to hold concepts about program storage. The 'memory is empty' score is seen to reduce immediately when the instruction to store the first item is explicitly given. Experience gained between pre- and post-test did not produce a significant change in the response pattern as predicted by the ranked scores.

If the Memory cell is not blank then apart from the user program, data items are stored. The data should not, of course, be just stored and forgotten but stored in a form which enables subsequent retrieval. It is predicted that the concept of data retrieval is rarely considered by the young novice and is probably one of the main causes of 'Memory' absence.

For stored data to be retrieved from their location they must be identified and, by convention a variable location label is used; in pictorial terms and to illustrate 'Memory', the labelled 'pigeon hole' works quite well. It is predicted, however, that the novice, will, if the Memory 'box' is not blank, show individual items written in as though on a sheet of paper. The concept envisaged is likely to be that of storage as an end in itself rather than as a temporary measure.

It was predicted that few subjects would use addressed memory locations before relevant practical experience at which time influences of age may occur. To establish whether or not the age-band of a subject has an influence on whether he/she uses an addressed or non-addressed memory locations for storing data, a Chi Square kxn was applied.

The results for the pre-test were such that the Null Hypothesis could not be rejected, but at post-test produced $X^2=14.46$ ($p<0.01$), means across all subjects were Addressed locations:0.35 and Non-addressed locations: 0.65, see Appendix(20). X^2 indicating that at post-test, relevant practical experience, (as described above), has exerted an influence on the relationship between year-band and the mode of memory storage which did not exist in a measurable form at pre-test.

It is indicated above that beliefs held by many subjects about the way data is stored in the system is modified through practical experience. The extent of the change appears to be affected by the year-band of the subject. At post-test, second year subjects, for example, are more likely to use addressed memory locations to store data items, followed by first and then third year subjects.

The desire to gain a fuller picture of the addressed data concept led me to consider, in addition to the difference shown between pre and post-test subjects in their use of addressed or non-addressed memory locations, the effect of interaction between age and practical experience.

The analysis looked at the way data was stored in the single Memory box at stage 5. It was expected, in terms of the immediate task and in addition to the user program, that three data items, i.e. 8,15,23 should be stored as individually addressed items. Appendix(21) gives the breakdown of stage 5 memory contents for all subjects.

A Two-Way Anova for repeated measures, with the two independent variables, age and intervention, was used to observe systematic effects. The variance caused by the main effects of the independent variables namely:

(a)Independent Variable (1) the effects of age on the variance

(b) Independent Variable (2) the effects of intervention or no intervention on the variance

(c)Interaction(age*intervention) the combined effects of factors 1 and 2 on the variance.

A score table, illustrated by Table(16), was devised giving a three point range of scores, 1 to 3, which could be allocated to pre- and post-test result for each subject:

Table(16):Three point score scale

Score	Content of memory box 5
1	Blank
2	Non-addressed numerical data
3	Addressed memory locations

The obtained value for the Independent variable(1) was $F_{(2,78)}= 5.85$ ($p<0.01$), indicating a reliable relationship between age, and the score achieved. That is, it is concluded that subjects from different year-bands do have different beliefs about the way data is stored in the system at the final stage of the program.

The obtained value for the Independent variable(2) was $F_{(1,78)}=19.26$ ($p<0.01$) indicating a reliable relationship between intervention, and the score achieved. It is, therefore concluded that the particular experience gained through intervention does affect subject beliefs about the way data is stored in the system at the final stage of the program.

The obtained value for the combined effect of Independent variables(1) and (2), age-group and interim experience, $F_{(2,78)}=3.76$ ($p<0.05$), indicates that a reliable relationship exists and it is concluded that age and intervention may influence test results. Tables Appendix(22)

Finally, an examination of the data stored in memory. The objective of the third test was to examine specifically the way data was stored in Memory stage or, box 5. In particular, the summation of the two stored numbers at stage 4, whether it appeared at stage 5 as an equation as a single figure or other. This particular analysis is of interest because it is related to the addressed memory location problem. For this example a=8, b=15 and c= a+b leads, even at GCSE level, to the final answer given as c=8+15 where an equation is found stored rather than 23.

The stage 5 responses for all subjects at pre and post-test are seen in Tables(17 & 18) and, in particular, it was predicted that the use of 8+15=23 in pre-test would be modified at post-test, that is as a result of practical experience. The data was subjected to Chi Square kxn at pre- and post-test. Producing 16 at pre-test and 11.5 at post-test ($p<0.01$) X^2 indicates a reliable relationship between age and the diminishing use of the equation 8+15=23 at both test levels. See Appendix(24).

<u>Table (17): PRE-test Memory Equation/No Equation OBSERVED</u>

YEAR	EQUATION	NO EQUATION	TOTALS
1	12	15	**27**
2	3	8	**11**
3	2	43	**45**
TOTALS	**17**	**66**	**81**
means	0.21	0.79 across all subjects	

<u>Table (18): POST-test Memory Equation/No Equation OBSERVED</u>

YEAR	EQUATION	NO EQUATION	TOTALS
1	8	15	**27**
2	0	11	**11**
3	2	41	**43**
TOTALS	**10**	**71**	**81**
means	0.12	0.87 across all subjects	

Moving to the second system object, the VDU(visual display unit), it was predicted that the observed process, because it was observed was likely to follow a different pattern of results to the hidden process, i.e the system memory. The first test, Chi Squared, was carried out to determine whether or not subject opinion about whether a screen is blank or not blank during processing differs significantly by age. I wanted to know whether a relationship existed between the subject age-band and the pattern of predicted 'VDU blank' at each of the five stages of the program.

At pre-test X^2 = 16.27 ($p<0.01$) indicating that subject ageband may well be exterting a significant influence on subject opinion but at post-test it was not possible to reject null hypothesis.See Appendix (25). The means at pre-test across all subjects: Blank 0.27 and Non-blank 0.73.

To look at the results from another perspective, I then applied a Two-Way Anova for repeated measures, with two independent variables, age and intervention as was applied on the memory element this was to observe

(a)Factor (1) the effects of age on variance

(b)Factor(2) the effects of intervention on the variance and

(c)Interaction (age*intervention) the combined effects of factors 1 and 2 on the variance.

A six point scale was devised to identify the number of blank screens observed during the five stages of the program and is seen in Table (19).

Table(19): Scores for Blank Screens 0-5

Score	Number of blank VDU
1	None
2	one
3	two
4	three
5	four
6	five

The obtained value, Appendix(23), for Independent variable(1), produced $F_{(2,78)}$=1.10. It was not possible, therefore, to reject the Null Hypothesis that the age of the subject and the number of 'Blank VDU screens" are unrelated, thus supporting the findings of Chi Squared above.

The obtained value for Independent variable(2), produced $F_{(1,78)}$=14.57 ($p<0.01$) indicating that the post-test performance was affected by intervention, i.e the preceding practical experience.

Interaction between Independent variable(1) and Independent variable (2), produced $F_{(2,78)}$ = 5.4 ($p<0.01$) indicating that post-test scores were affected by the combined effects of subject age and practical experience.

At pre-test it is seen that the ordered means of the scores, in ascending order, lowest score being most accurate, is year 1, year 2 and year 3 and at post-test changing to year 3, year 1 and year 2. See Table(20).

Table(20): Means related to Blank screen predictions.

	year1	year 2	year3
pre-test means :	1.9	2.0	2.7
post-test means :	1.7	2.3	1.5

Discussion: Memory and VDU

The study has clearly confirmed that novices do not, in general, consider the system or application program as occupying space in the system memory. Both pre- and post-tests show the highest allocation of 'blank memory' to stage 1, that is, when the program has been input but not executed. In ascending year order at post-test , 93%, 100% and 84%.

It can be seen that immediately the memory component has explicit use, e.g. the program starts to store data to memory, at stage 2, the incidence of 'the memory is blank' is reduced and continues to reduce throughout the 5 stages with one exception. The exception being stage 4 at years 1 and 3; i.e. two previously stored numbers are added together, therefore not a task explicitly related to 'storing data'. These findings confirm the prediction that many novices do not consider the use of memory unless that use is explicit.

On the whole it is the second year subjects who score lowest in terms of empty memory prediction followed by the first year with the third year scoring the highest. The results indicate a reliable relationship between age and pattern of responses at each of the five stages at pre- and post-test.

The pre- and post-test group means for the three age-groups follow the same pattern for both tests and have similar mean differences, as seen in Table(21)

<u>Table(21): Pre and Post Differences in Means</u>

test 1 difference of means	yr1: 1.3	yr2:3.2	yr3: 0.7
test 2 difference of means	yr1: 1.1	yr2:4.8	yr3: 0.8

In the DESMOND exercise it was shown that at post-test some subjects had begun to move away from the perception of 'memory' as a nebulous space and to think of the memory space as partitioned into separate, individual 'boxes' or cells. In the present study the subjects were asked to consider separate memory locations from the onset of the exercise so the significant point of interest was whether the locations were a) actually used in the storage description and b) whether the storage locations depicted were 'addressed'.

Memory, perceived as a 'nebulous space' without an attached concept of subsequent retrieval, has been seen to change as a result of the practical exercise. The addressed memory cells phenomenon made a significant appearance in stages 2 to 5 of the post-test at 26%, 31%, 28% and 27% levels.

The five stages of the exercise together with the percentage of subjects allocating data to memory and using non-addressed storage locations are displayed in Table(22) below. The table shows clearly the significant proportion of change from non-addressed to addressed locations beginning at stage two.

Table(22):Pre and Post-test Percentage allocation of data to Memory

STAGE	PRE-TEST	POST-TEST	
1.	2.47% non	2.48% non	nil
2.	50.61% non	25.93% non	25.93% Addressed
3.	60.48% non	33.33% non	30.88% Addressed
4.	48.14% non	25.93% non	28.38% Addressed
5.	55.55% non	29.63% non	27.67% Addressed

The above results suggest that all 'pre-test' subjects whose model uses system memory, perceive the memory as a formless space where data are stored. A comparatively short exercise reveals a significant change to the adoption of 'separated' storage spaces and, to a lesser degree, addressed separate storage spaces thus making possible the concept of data retrieval to be developed in model form.

Progressing to 'what' was stored in Memory location 5 in terms of the stored sum of two numbers. It will be recalled that two numbers 8 and 15 had been keyed in and stored at two separate addresses. The next task was to sum the contents of the two addresses and store the sum at a third location.

From previous experience of the problem, it was predicted that, certainly at pre-test, many subjects would store the equation 8+15=23 in memory rather than the solution 23. The results showed a significant relationship between the use of the 'equation' format and the age of the subject both at pre and post-test. The developmental pattern is seen in the percentage incidence Table(23) at both tests, with the post-test result clearly demonstrating improvement for first and second year subjects.

Table(23):Percentage Incidence:Pre/Post test Equation/No Equation

YEAR	1	2	3
PRE%	44	27	4.4
POST%	30	nil	4.4

Although it was not possible to show that the age of the subject on its own significantly influenced his or her pattern of Blank VDU results, experience and experience combined with age was shown as significant. Table(24) shows a comparison of pre and post-test percentage of Blank VDUs illustrating the reduction in 'empty screen' beliefs as a result of experience. See also Appendix(26) for raw scores.

Table(24):Pre/Post test %age

	Yr1	Yr2	Yr3
Stage	30/30	0/82	0/9
Stage2	15/11	27/9	44/7
Stage 3	7/11	9/9	42/12
Stage 4	26/7	36/9	47/12
Stage 5	11/nil	18/9	40/14

It can be seen, from the foregoing, that positive changes have been effected as a result of a short practical experience, in terms of :

a). an overall reduction in beliefs that memory and VDU are empty or blank,

b). increased use of 'how' memory is percieved; addressed memory locations and

c). increased use of 'correct format' data for what is stored in memory.

It has been considered frequently during the foregoing studies, that objects, qualities and instructions explained to the novice may not have always been interpreted in quite the way expected and clearly such a factor would directly affect the success of a conceptual model. Johnson-Laird(1983) for example, uses the example of a bee collecting nectar, where A is the actual state of affairs in terms of the location of nectar found. The internal representation or model of the spatial location of the nectar is represented by A' and the model referencing the essential features to fellow bees via dance signals represented by A".

Johnson-Laird draws attention to the fact that the bee's referencing is dissimilar to human communications which tend to suffer from literal recovering the speaker's referential intentions.

Referential intention is not in isolation under the umberella of language related phenomena. The language used to explain computer related objects or processes to the novice will contain a mixture of vocabulary; some carrying prior meaning and some new. Whilst ease of learning is dependent on the extent to which objects, relations and language in the new system can be integrated into the existing components of knowledge it is not necessarily beneficial. Lewis(1986) points out that learners unfamiliar with a domain often make connections that are not valid and students may mis-interpret or distort information to fit their naive views.

The emphasis in the preceding studies has been placed on the objects drawn by young computer users, and, although the introductory instructions to the novices were verbal, they were posed in such a way that picture images could be anticipated and subsequently expressed graphically. It was not expected that the system would be evoked as a whole but rather that the non-inferential component parts would be generated in a form which the subjects could draw. It is not known whether the graphical expression imposed any constraint upon the subject but it is interesting that on occasions they felt free to incorporate verbal expression into their drawings.

The major difference evident between the pictures was the extent to which the subjects had developed independent free standing components into an interactive system. Indeed age related differences, i.e. numbers of component, types of components and their interaction, could be a strong argument for supporting a relationship between picture imaged and propositional thought.

This introduction of the concept of propositional thought to the development of mental models requires the search for some underlying factors which could affect the nature of the mental model produced. Therefore, the following chapter is an attempt to measure, from several vantage points, some of the affects of language which appear to be particularly relevant to the novice computer user.

CHAPTER 6

THE ROLE OF LANGUAGE

This, the final chapter of the empirical section, looks at the relationship between language and the mental model. In the search for understanding about a system and its use, any factors which appear to significantly affect the construction of good model concepts should be examined. I believe language is one such factor.

Concepts emerge and are perceived through language. To quote the argument later developed by Johnson-Laird, (1988), language enables people to experience the world vicariously since they can imagine it on the basis of a description. That description in turn is produced from the speaker's model of the world. As educators we should examine this medium, particularly with regard to the terms used, embedded beliefs and the impact they have on the novice learner.

To be able to describe and predict the behaviour of a system the novice requires the basic tools for the job; the appropriate language or terminology and understanding. With these points in mind it is noted that

Greeno(1977,78) suggests that understanding depends on three important criteria ; internal coherence, validity and integration. Internal coherence for the system user, argues Riley(1986), is aided by several forms of knowledge, about the action and syntactic structure of the commands used, how the command works and about the objects involved, for example, files, buffers and program.

The Riley interpretation provoked the idea for a practical study of observed novice understanding of computer related terms and their implications. The investigation examined the perceived role of computer related terms and was developed in three parts. The first two parts, A and B presented the novice with a pre-determined selection of role categories to which they were asked to allocate a selection of computer related items according to their perceived function. In part A the terms and symbols were presented as separate items thereby removing relational stimuli whilst part B items were presented in sentence form and required decomposition before allocation to the appropriate categories.

Part C progressed to the analysis and debugging of a faulty program written in BASIC and, finally, an examination of explanations given about the syntactically required inverted commas used in conjunction with the 'PRINT' statement.

Language Study A

For Language Part A a list of 20 computer related words most commonly used by the subjects was drawn up as seen in Table(25). However, the underlined words were omitted from the first year pupil list because they were not contained in the syllabus for that year:

Table(25) Language Part A: Computer Related Words.

INPUT	NEXT	NEW	COLOUR
PRINT	RUN	CHAIN	OLD
PROGRAM	*CAT	MOVE	MODE
LINE NO.	LIST	DRAW	REM
FOR	SAVE	CLS	DISC

The four categories chosen were: COMMAND CONTROL INPUT/OUTPUT and OTHER, and whilst it is acknowledged that no category is exclusive, it was expected that the novice would choose the category which best matched their beliefs about the word.

Command: There are two ways of getting the computer to do something, firstly by giving it a command which is acted upon immediately, or secondly, by giving it a series of numbered instructions called statements which can be stored in memory until instructed to carry them out in sequence; this stored set of instructions is known as a 'program'. Many of the keywords in Basic can, of course, be used both as commands and statements but it is not thought likely that the novice would differentiate between the two terms. When allocating an item to this category the novice may be correct at a primary level, but for the purposes of this study it is considered insufficient and evidence that the subject has not recognised the deeper implications of the item.

Control: As a result of independent 'expert' classification of the words my original items for the control category - PROGRAM, LINE NUMBER, OLD and RUN were reduced to LINE NUMBER. Firstly, as has already been revealed, the model of system memory held by the novice fails to include storage of the operating system or user software and its related data, therefore LINE NUMBER featured appropriately as a key item. Secondly, all subjects beyond the first year use program control structures in the form of loops; conditional and non-conditional, therefore, this category was intended to reveal how the subjects allocated the loop control items, FOR and NEXT.

Input/Output: The requirement of the input/output category was primarily to show whether novices recognised the less obvious commands such as LIST, DRAW and COLOUR, which output to the screen or printer on execution, as output related functions. All the subjects were familiar with elementary programming techniques to produce colour and graphics on the BBC(B) micro-computer but it should be noted that although the colour and graphics are automatically output to VDU the BASIC codes are not explicit instructions to do so.

Other: Any items which could not be placed in one of the other three categories by the novice were to be placed in this column.

Memory: Evidence has been revealed elsewhere in this thesis that the novice does not consider the system memory as fundamental either to its operation or for storage and running of programs and data. In general, memory is recognised only in its explicit forms and the final phase of Part A offers a further opportunity to examine this trend. A new single category was introduced for 'Memory' at the end of the first task which replaced the former four. The same items were read again and this time the subjects were asked to write down, under the single heading, 'Memory' items which they believed would affect the computer memory in any way by their use.

Method

The subjects were two hundred and twenty nine pupils from City of London Freemen's School. They were a mixture of boys and girls from six year-bands:

28 first, 45 second, 60 third, 58 fourth, 24 Lower fifth and 14 Upper fifth

The subjects were asked to create the four category column headings: Control, Command, I/Output and Other on file paper, and the exercise was explained to them. The explanation included a discussion about, and practice with, a list of non-computer related words which were categorised during the discussion. By chance it became clear during the discussion that some words could be categorised in more than one way, thereby re-inforcing my assurances that the opinion of each subject was far more important than 'getting a right answer'.

At the start of the exercise the subjects were asked to listen carefully to each word as it was read aloud, decide which heading it belonged to and write it down in the appropriate column. In cases where they thought that more than one heading would be appropriate they should choose the one heading which he/she thought was the most suitable descriptor for that item. Any item which could not be placed appropriately under one of the first three headings was to be placed under 'Other'. Each word was read twice to the subjects as a separate item, thereby removing relational stimuli.

Results and Discussion

It was predicted that the subjects were unlikely to categorise the words in the same way as an expert therefore, as a first step, I categorised the set of words and then asked four independent 'experts' to carry out the same task in accordance with the definition above. Whilst high agreement was found between the four independent experts there was not total agreement with my own categorisation although based on the same definition. As a result of the disagreement statistical analysis was carried out using the catagorisation generated by the independent experts as shown in table Table(26). Percentage results appear in Appendix(27).+ (27a)

Table(26)Languague Part A: Expected category allocation

COMMAND	CONTROL	I/OUTPUT	OTHER
RUN	LINE NO	INPUT	REM
SAVE	FOR	COLOUR	PROGRAM
MOVE	NEXT	PRINT	
*CAT		DRAW	
NEW		DISK	
OLD		LIST	
CHAIN			
CLS			
MODE			

To establish whether there is a significant difference between the observed and expected categorisation of the words, a Chi Square X^2, One Sample Test was applied separately to each of the six age bands, see Appendix(28). This produced the individual year-band results, seen in Table(27), at $p<0.01$ level with Chi X^2 indicating that there is a reliable difference between the way young novices categorise the computer related items and the expected.

Table(27) Language Part A: Chi Square Results:

YEAR	CHI SQUARE X^2
1	197.01
2	392.91
3	446.15
L4	419.99
L5	141.28
U5	158.42

N=4 DF 3

To establish whether there was a significant age-related difference in the category choice order made by the subjects and whether that choice was directed, the data was subjected to a Two-way Analysis of Variance.

The results were subjected to a Friedman Two-way Analysis of Variance by Ranks(X^2_r) to establish whether subjects from different age-bands categorise computer-related words similarly. It produced $X^2_r = 17$ ($p<0.01$), X^2 indicating that the subjects had not acted independently and that a reliable similarity exists across the age-groups giving direction to the Command category. See Appendix(29)

Table(28)Language Part A:RANKED DATA

YEAR/CATEGORY	CONTROL	COMMAND	I/O PUT	OTHER
1	1	2	4	3
2	1	3	4	2
3	1	3	4	2
L4	1	3	4	2
L5	1	3	4	2
U5	1	3	4	2
SUM	R1=6	R2=17	R3=24	R4=13

It was further predicted that developmental differences exist in the way subjects choose to allocate lists of computer-related words to the single 'Memory' category. The results from a single categorisation were expected to reflect age-related changes, albeit at a simple level, in subject understanding, about the role of system memory. Therefore, the data from the Memory category was subjected to a One-way Analysis of Variance for Independent Samples to establish whether the average number of allocations to the memory category is affected by the year- band of the subject. This yielded $F_{(5,84)}=25$ $(p<0.01)$. Raw scores, percentage incidence and summation to F in Appendix Table(30), and the results clearly suggest that the awareness of system memory increases with age and/or experience.

The F test shows that at least two means differ significantly, therefore, the Scheffe S test was applied to find the limits of confidence interval (I), for each difference between means. It produced $I = 7.93$ $(p<0.05)$ and $I= 9.50$ $(p<0.01)$. It can be seen from Table(29) that the means for the 1st and 2nd year subjects differ significantly at the 0.05 level whilst all others differ significantly at 0.01.

Table(29) Language Part A :

Means differing significantly at $p<0.05$

1st and 2nd years

Means differing significantly at $p<0.01$

2nd and L5th	3rd and 1st	4th and 1st
2nd and U5th	3rd and L5th	4th and L5th
	3rd and U5th	4th and U5th

The individual computer related words in Table(30), reproduced below, were allocated to four categories Command, Control, In/Output, Others and results indicate a reliable similarity in categorisation by subjects from different year-bands; 'Command' taking the dominant role.

TABLE(30) Language Part A: Computer Related Words.

INPUT	NEXT	NEW	COLOUR
PRINT	RUN	CHAIN	OLD
PROGRAM	*CAT	MOVE	MODE
LINE NO.	LIST	DRAW	REM
FOR	SAVE	CLS	DISC

In Language part A, word allocations to the Command category of particular note are PRINT which was chosen by over 78% of all subjects along with LIST 64% and DRAW 53%. Normally considered as strongly and primarily output related functions their allocation to this category clearly contributes to the surprisingly low scores found in the INPUT/ OUTPUT category. The pattern of allocation across the six age bands appears to be consistent, whether at the low, middle or high percentage response level.The interpretation made at this stage is that the word 'command' has been used in its common everyday sense, for example to command a dog to SIT.

In the control category,see Table(31), FOR and NEXT, have been given zero scores by the first year subjects but progress to a solitary equality at 40% at Upper five. NEXT is a familiar and commonly used word; easily identified with repetition and queues. FOR, on the other hand, appears to have little relevance and, although together they form a control structure, NEXT is generally regarded as more an item of control than FOR indicating they are not modelled as a 'working pair'. In practice this is a common stumbling block even at GCSE level.

Table(31)Language Part A: Control Category examples

control %	1	2	3	L4	L5	U5
LINE NO.	53.6	22.2	6.7	32.8	16.7	33.3
FOR	0.0	31.1	8.3	19.0	25.0	40.0
NEXT	0.0	40.0	10.0	29.3	41.7	40.0

Similarly, the words MOVE and DRAW, which might be considered as having equal importance in the production of graphics either to screen or hard-copy, show identical scores for the first three year-bands, see Table(32), but then proceed to pull apart with DRAW retaining the higher score . It should be noted that the four 'experts' did not categorise MOVE as input/output related word as they did for DRAW.

Table(32) Language Part A:Input/Output Category examples

input/output%	1	2	3	L4	L5	U5
INPUT	100.0	100.0	100.0	100.0	100.0	73.3
PRINT	0.0	2.2	16.7	3.4	12.5	20.0
DRAW	7.1	6.7	1.7	5.2	16.7	20.0
MOVE	7.1	6.7	1.7	3.4	8.3	13.3

Results in the Memory category clearly confirm and reinforce the findings above, that is, whilst the computer is incapable of functioning without memory the novice appears to recognise only its explicit use. The common characteristic of the highest percentage words in this category is that they are explicit instructions. For example, to store a program onto disk, to recover a program if BREAK key is pressed in error (OLD), or to clear a program from RAM before starting another, 'NEW' program. A further example, the command *CAT, an abbreviation of *CATALOGUE, a request to see the catalogue of files on the disk - in this case a pupil would be requesting the list of files which appear on his/her own directory on the fileserver work disk.

The words which were allocated to the memory category by over 50% of the subjects in each year-band are listed in Table(33) in descending percentage order. The significance of the selection is that in five out of the six year bands the highest scoring word is "SAVE " related to the explicit use of memory, the command used to save data to disk.

Table(33) Language Part A: Memory Category Allocation over 50%

1st	2nd	3rd	L4	L5	U5
LIST	SAVE	SAVE	SAVE	SAVE	SAVE
DISK	NEW	OLD	PROGRAM	OLD	OLD}
OLD	DISK	NEW	REM	NEW}	PROGRAM}
RUN	PROGRAM	*CAT	NEW	CHAIN}	DISK
PROGRAM}	*CAT	LIST}	OLD}	DISK	LIST
REM}	OLD	PROGRAM}	INPUT		
CHAIN}	CHAIN}				
	REM }				

It has already been noted elsewhere that the REM statement is a device by which remarks/reminders can be held in a program listing without affecting the running of the program (i.e. for reader or programmer aid when reading the program listing). However, many lower school pupils believe that to include the name of the program in a REM statement is the equivalent to naming a file when it is 'saved' to disc. They also believe that failure to include it will render them unable to subsequently retrieve the program; the importance placed on the term can be seen from the figures below, Table(34), with the lowest score given by subjects from lower five, that is, first year GCSE Computer Studies candidates:

Table(34)Language Part A:Rem%

YEAR	1	2	3	L4	L5	U5
REM%	53.6	51.1	45.0	62.1	20.8	46.7

Line graphs, Appendix(31), were computed for the percentage allocation against year of subjects for each of the 'words' in the Memory category and the patterns examined in order to isolate particular year trends, (Graph1). Similar year trend patterns, indicated by Table(35) were then computed to the same graph to show the similarity of trend across the yearbands.

Table(35) Language Part A: Graph Comparisons

a. PROGRAM, LINE NUMBER, REM and INPUT
b. MODE, COLOUR, INPUT and NEXT
c. SAVE, *CAT and CLS
d. NEW and NEXT
e. OLD, LIST and RUN
f. PROGRAM, PRINT, MOVE and DRAW
g. DISK, CHAIN and FOR

Conclusion

In summary, the results show that, as a whole, we cannot expect novices to perceive the role of a computer related word in the same way as the expert. There is also a reliable similarity in the way subjects allocate computer related words to Command, Control, Input/output and Other categories regardless of their year-band all tending toward Command as a role descriptor. Where the subjects do not act similarly across the age-range is in allocation to the Memory category. Here we can expect a reliable relationship between the number of memory allocations and the age-band of the subject, indicating an age and/or experience related awareness of the need for system memory.

Language Study B

Language Part B is a progression from Part A. By presenting the material in complete sentence form rather than as unrelated items it was hoped to establish whether the catogory allocation was affected.

The number of categories was increased from four to six in the first part of this study; they comprised the original four categories from Part A plus 'Data' and 'Memory'. Data was included to accommodate data necessarily included in the program sentences and Memory as a comparator for Memory in Part A. It should be noted that due to the significant differences of age and experience between the subjects it was not possible to represent all the items used in Part A in a common core program.

There were two main aims of the investigation, firstly to collect evidence of novice understanding about 'Data' and its different role category in the program statement. Secondly, to compare common elements of studies A and B in order to establish whether the categorisation of an item or its perceived role is affected by the mode of presentation.

Method

One hundred and eighty six CLFS pupils were involved in the study; 27 first year, 44 second year, 60 third year and 55 fourth year. As an introduction each group of subjects was reminded of the work undertaken in Part A one week before, then asked to read the program as shown in Table(36).

Table(36)Language Part B: Basic Program before breakdown

```
NEW
10 REM DRAWING NAME
20 MODE 2
30 GCOL0,1
40 PRINT "FRED"
50 MOVE 500,500
60 DRAW 200,200
RUN
```

The program sentences were split into 16 segments for categorisation as seen in the sample data capture sheet, Appendix (32). The subjects were asked to put each of the sentence segments under one of the headings; Memory, Control, Command, Input/Output, Data or Other. As in Study A the sentence segments were read aloud twice i.e. each item was read and then repeated.

Results and Discussion

Following the results of Language Part A, which clearly indicated that subjects do not categorise single computer-related words as expected, I wished to know whether changing the mode of presentation would affect the outcome. Could we assume, for example, that computer-related items given in program sentence 'context' are perceived and therefore categorised in a similar fashion to items given without contextual support.

The same four 'experts' who categorised the computer related words in Part A above agreed on the expected category allocation which appears in Table(37). Again the allocation was not in total agreement with my own. It should be noted that the inclusion of a Memory category as an option raises the suggestion that all computer related items should be allocated to it. However, for the purposes of this exercise each item has been allocated according to its perceived role within the processing system as shown below. For example, item NEW has been allocated to Memory in accordance with its role as an explicit command to empty user memory (RAM) when starting a new programming task; the subjects use the command for this purpose.

Table(37)Language Study B :Expected categories:16 sentence items.

MEMORY	CONTROL	I/O PUT	DATA	COMMAND	OTHER
NEW	10	DRAWING(title)	0 1	MODE	REM
	Comma(,)	GCOL	FRED	RUN	MOVE
	Inv.commas(" ")	PRINT	200 200		
		DRAW	2		

The collected raw scores appear in Appendix(33) together with the percentage incidence.

Following the same procedure as Part A a Chi Square X^2 One Sample Test was applied separately to each of the four subject age bands to establish whether a difference exists between the observed and expected categorisation of the computer related items given in program context. It produced Chi Square X^2 ($p < 0.01$) results for each year as shown in Table(38), all exceeding the critical value, therefore we can conclude that a significant difference does exist. Appendix(34)

Table(38)Language PartB Chi Square results

YEAR	CHI X^2
1	218.15
2	235.47
3	334.42
L4	281.26

N=6 Df=5

A Friedman Two-way Analysis of Variance by Ranks (X^2_r) was applied to the data to establish whether subjects from two different year-bands perceive the role of the computer-related item with contextual clues, differently. It was again, predicted that direction to the COMMAND category would occur, Table(39). The table of deviation of observed from expected can be seen in Appendix(35).

Table(39)Language Part B: Ranked data

Year/ Category	Memory	Control	Command	I/O put	Data	Other
1	2	4	1	5	6	3
2	2	4	1	6	5	3
3	2	4	1	6	5	3
L4	2	4	1	6	5	3
SUM	**8**	**16**	**4**	**23**	**21**	**12**
	R_1	R_2	R_3	R_4	R_5	R_6

It produced $X^2_r = 19.57$ ($p<0.01$) X^2 indicating that subjects from different year-groups do not differ significantly in the way they categorise 'sentence items'. See Appendix(35). It is also observed that similarity in cross-year allocations has caused a directed result to the COMMAND category.

Finally, and in order to compare Studies A and B, the data were again tabled for a One-way Analysis of Variance for Independent Samples in order to establish whether when the memory category is one of several, the average number of allocations to it is affected by the year group of the subject.

However, it produced $F_{(3,60)} = 1.55$ indicating that the Null hypothesis stating that the average number of allocations made by subjects from different year-bands are similar, was upheld.

The role perception of 'Data' was of particular interest in Part B. Information was extracted from the DATA category and the percentage calculated for subjects allocating the four data items Line number, Mode number, Colour number and Position co-ordinates to this category as in Table(40):

Table(40)Language Part B

Percentage allocations for Line,Mode,Colour and Co-ordinate Data

DATA/YEAR	IST%	2ND%	3RD%	L4TH%
10	7.4	13.6	28.3	23.6
2	3.7	18.2	16.7	27.3
0,1	3.7	20.5	21.7	27.3
200,200	7.4	11.4	18.3	23.3

The table of results was examined for each year band to see whether the category receiving the highest allocation for the four data items had, in fact, been the DATA category; the pattern that emerged is seen in Table(41):

Table(41)Language Part B: data items

1st year	None out of the top four were Data items.
2nd year	One out of the top four: data item {(0,1)}
3rd year	Two out of the top four: data items {(10), (0,1)}
4th year	Three out of the top four: data items{(10), (01),(200,200)}

The general allocation of data items to the DATA category reveals an increasing, age-related trend although no one item achieves greater than

28.3%. This low 'maximum' percentage leads to the interpretation that the first year subject begins with a low awareness of the different role and nature of Data but that this awareness undergoes a gradual, time and experience related development. The above 'top four' allocations to the Data category, Table(41), lend further weight to indicators of the upward trend of 'DATA' recognition

Results in Table(42) for four age-bands, below, shows the preferential allocation for the MEMORY category with NEW and REM clearly leading in all years. Similarly, MOVE, PRINT and DRAW are examples least related to the memory category, again in all years.

Table(42)Language Part B:Memory category breakdown

YEAR 1%		YEAR 2%	
56.0	NEW	31.8	NEW/REM
48.0	REM	29.5	10
14.8	MODE	15.9	0 1
11.0	10/0 1 /PRINT/200 200	13.6	"" /2 /GCOL
7.4	GCOL/""/RUN	11.4	RUN
3.7	DRAWING /2 /FRED /DRAW	9.1	200 200/ FRED/,/MODE
0.0	, / MOVE	4.5	DRAWING
		2.3	DRAW
		NIL	MOVE / PRINT

YEAR%		YEAR L4	
53.3	REM	54.5	NEW
33.3	NEW	40.0	REM
25.0	10	25.5	FRED
21.7	2	10.9	MODE / GCOL / RUN
13.3	MODE/ GCOL	9.1	10 / 2/ 0 1 /200 200
11.7	FRED / 0 1	5.5	, / DRAW
6.7	RUN	3.6	DRAWING / PRINT / ""
5.0	200 200 / PRINT/ DRAWING	1.8	MOVE
3.3	"" / MOVE		
1.7	DRAW		
NIL	,		

Line graphs were computed for each year-band showing the percentage of allocations to the memory category for each of the 16 sentence segments, see Graphs(2). The results were then examined in order to establish whether a pattern of novice response existed for the same words across the first four year bands, Graphs(3). The findings were positive and as can be seen by the graph the overall trends appear similar with some peaks and troughs following closely for all age bands. Further graphs show paired year bands, Graphs(4). See Appendix (36)

A Comparison of Study A and Study B

Studies A and B were both involved with the categorisation of Computer related words. Given predetermined categories and a list of items, the subjects were required to allocate the items to categories according to their perceived understanding of the item role.

The mode of presentation in the two studies differed in that for Part A subjects were presented with a list of individual items which had no contextual clues whilst Part B items were, in the first instance, presented as a BASIC program thereby giving context, before the sentences were reduced to individual items for allocation to categories.

The first test to be undertaken on both parts established that subjects categorised the items differently to that expected. The second examined whether subjects from different year-bands differed in the way they categorised the list of words and a reliable similarity of responses was found across year-bands in Parts A and B, and they were directed to the Command category.

The third test focused on the Memory Category and revealed that the number of allocations to the memory category is influenced by the year-band of the subject when the items are given without context but not when put into context.

As can be seen from Table(43), there is a significant drop in memory allocation percentage between words/functions given as a list and given in sentences/program before being listed. Except for REM which appears to be maintained at a higher level giving further confirmation that a misconception exists.

Table(43):Memory Allocation

	FIRST YEAR	SECOND YEAR	THIRD YEAR	FOURTH YEAR
INPUT	25.0/3.7	46.0/9.1	15.0/11.7	56.9/25.5
PRINT	28.6/11.0	37.8/NIL	28.3/5.0	34.5/3.6
LINE NO.	28.6/11.0	46.7/29.5	30.0/25.0	50.0/9.1
RUN	57.0/7.4	28.9/11.4	48.3/6.7	44.8/10.9
MOVE	17.9/NIL	24.4/NIL	16.7/3.3	19.0/1.8
DRAW	17.9/3.7	26.7/2.3	21.7/1.7	24.1/5.5
COLOUR	14.3/7.4	33.3/13.6	16.7/13.3	25.9/10.9
MODE	35.7/14.8	37.8/9.1	26.7/13.3	46.6/10.9
REM	**53.6/48.0**	**51.1/31.8**	**45.0/53.3**	**62.1/40.0**

Line graphs were computed showing results for Part A and Part B for each of the sentence segments in the memory category across the four year bands, Graphs(5), Appendix(37). The results show a significant reduction in the allocation to memory category when items were given in sentence form and quite often show an inverse trend, Table(44).

Table(44):Memory Allocation
Average percentage allocation on the set of items

	STUDY B	STUDY A
Year 1	12.5%	31.7%
Year 2	12.8%	40%
Year 3	13.3%	32.6%
Year L4	13.3%	40.8%

LANGUAGE STUDY C

In Language part C subjects were asked to predict the outcome of keying in a familiar but modified piece of computer code. It was believed that if an adequate and helpful model existed, then subject explanations should be diagnostic and prescriptive. However, it was predicted that responses generally would anticipate a 'correct' program.

It has already been noted that the novice frequently fails to notice errors in programming code, tending instead to pick up on contextual clues; the trait continuing even when the program fails to work as anticipated. I believe, after observing novices and programming language problems, that many of the misconceptions are persistent and not determined by age. Thus, within the broad area of language there is a similarity between novices of different ages.

Method

A five line program was written in Basic with level and content within the common experience of the forty eight subjects in the study as Table(45).

The program, if input and executed, would cause the computer to print the word SUNFLOWERS in red 20 times down the top left hand side of the screen.

Table (45)Language Part C: Correct Program

```
10  MODE 2
20  COLOUR 1
30  FOR N=1 TO 20
40  PRINT "SUNFLOWERS"
50  NEXT N
```

The program modification comprised removal of the second half of the loop structure,i.e. the NEXT removed from the FOR-NEXT relationship, and re-arrangement so that the program statements were in an incorrect order whilst maintaining the correct line number sequence, as in Table(46).

Table(46)Language Study C:Bugged Program

```
10 COLOUR 1
20  PRINT SUNFLOWER
30  FOR N= 1 TO 20
40  MODE 2
```

Forty eight pupils of CLFS, from the first, lower fourth and upper fourth years were involved in the study. The subjects were asked to read the program and predict the output if it were used as input to the computer and executed.

The first line of the modified program, which selects Colour1(red), could not be implemented because the computer requires Mode 2 or 5 rather than its normal start-up Mode 7. This fault would not, however, stop the program from proceeding if there were no other faults, it would merely produce the text in normal sized white rather than in chunky-sized red characters.

The second line of the program goes on to a PRINT statement which can be used in three ways. Firstly, as a means of printing any alpha-numeric or numeric string, requiring no meaning to the computer in its processing task, but is syntactically required to be enclosed between pairs of inverted commas: e.g "SUNFLOWER". On execution, the contents of the inverted commas are printed to the VDU or printer as required. Secondly, the PRINT statement alone is often used to 'Print' an empty line either to space out screen text presentation or to produce a well spaced program listing, making it easier to read. Thirdly, it may be used as a means of printing out the present value of a numeric or alpha-numeric variable, the value of which has been previously declared, e.g PRINT SUNFLOWER if, for example, SUNFLOWER=20.

Because the word SUNFLOWER has not been placed between inverted commas in the modified program and was not assigned a value as a variable the first error message will notify the user of "No such variable at line 20".

Results and discussion

On completion of the exercise it was found that, although they varied in content and length, each explanation included one or more of the 14 statements list in Table(47).

Table(47)Language Part C: Subject predictive statements

1. Inverted commas required
2. Nothing
3. Print (or clearly implied)
4. Draw
5. Red
6. A colour
7. A coloured background
8. A sunflower
9. Sunflowers
10. All over the screen
11. 20 times
12. time in seconds
13. 20 lines down on the screen
14. Mode 2

To establish whether a significant relationship existed between the year-band of the subject and the ranked program responses, a Kendall's Coefficient of Concordance 'W' was applied with the result, $'W'_{(13)} = 0.636$ and $X^2 = 24.801$ ($p<0.05$) indicating that the overall rankings of the subject responses are not independent nor unbiassed. The ranked scores have shown a similarity of thought across the age-bands. Appendix(38)

The large percentage of subjects predicting a correctly written and running program led to the allocation of a five point scale, 0 to 4, according to the number of correct parts from PRINT on the screen, Red, SUNFLOWER, 20 times, mentioned in each of the subject descriptions. A One-way Analysis of Variance was then applied to establish whether there is any difference between the way the different year groups predict the program effect on execution. See Appendix(39) for data tables and Analysis of VarianceTable. It produced $F_{(2,46)}=12.93$ ($p<0.01$) indicating that subjects from different year-groups do predict the outcome of a 'bugged ' computer program differently.

To determine which of the year-bands differed significantly, a Scheffe Test for comparing means was applied with result, see also Appendix(39), I = 1.838537 for $p<0.05$ see Means Table(43):

Table(48) Language Part C Table of means:

M1= 3.2	M1-M2 = 0.06	M2-M3=2.05
M2= 3.1	M1-M3= 2.11	
M3= 1.09		

The result indicates a significant difference of means between Years 1 and U4 and again between years L4 and U4. Showing that the first year subjects, in their assumption that they are dealing with a 'working' program, predict more of the correct elements that would result from its excution, the lower fourth subjects follow very closely so that both years achieve a significantly higher mean score than the upper fourth.

Setting a problem, such as in Part C, has proved to be a rapid method of finding weaknesses in the program structure models held by novices. Misconceptions appear to be deeply embedded and have resisted modification through feedback during the many weeks of practice.

The table of 14 program parts, with the percentage incidence for each part was ranked as seen in Table(49).

Table(49) Study C:Modified Program Ranked by number of incidences

1ST YEAR	L4%	U4%
%(part)	%(part)	%(part)
92(9)	92(9/3)	46(9)
88(3)	85(5)	36(3/2)
83(5)	69(11)	27(6)
67(11)	23(14)	18(1/5/14/8)
21(12)	8(12/13)	9(4/7/10/11)
13(14)	8(8)	15(1)
4(1/4/13)		

Showing, in order of priority, that the most likely prediction is that:

a). the word SUNFLOWER will be used,

b). the words will appear on the screen,

c). the words will appear in red, and,

d). they will be repeated 20 times.

One of the commonest problem areas is in the use of a loop for repetition of an instruction. The FOR-NEXT 'Loop' structure is an obvious example and it is appropriate to recall the weak relational model held for FOR and NEXT and the structure's low percentage allocation incidence in the Control category Table(50). Apart from the repeat loop for printing "Sunflowers' 20 times, reference is made in the novice explanations to "20 seconds" as example 21, Year1 and "20 lines down from the top of the screen" as example 11, year L4. Appendix(40) gives a breakdown of the loop type incidence.

Table(50)Language Part C:Loop example

For reference the three types of loop are illustrated:
1. FOR loop = 1 to 20:PRINT "SUNFLOWER" : NEXT loop
To print the word SUNFLOWER on the screen 20 times
2. FOR loop = 1 TO 20 : PRINT :NEXT loop
To print 20 rows of 'nothing' i.e. move the cursor down the screen 20 lines
3. FOR loop = 1 TO 20 : NEXT loop
Note there is no PRINT statement -a simple counting or delay loop

Twenty three of the twenty four first year subjects saw no error in the program as can be seen from the following explanations:

Year 1
1. Line 20 needs inverted commers and if it did have them it would print in red sunflowers for 20 seconds.
2. It will have 20 lines saying "sunflowers" in red.
3. Type sunflowers 20 times in red.
4. It will make the word sunflowers in red appear on the screen 20 times.
5. It will write sunflowers come out 20 times in colour one(red)
6. In red it will print sunflowers 20 times
7. Make the computer print 20 red sunflowers
8. I think it will print sunflowers in red 20 times
9. It will print in red on the screen "Sunflowers" 20 times
10. I think it will print sunflowers in red 20 times
11. This program will print in red in Mode 2 sunflowers 20 times
12. It will print in red sunflowers 20 times on the screen but it will be in Mode 2
13. Print Sunflowers in red 20 times
14. It would have a sunflower in red 20 times
15. It would have sun sunflowers in red 20 times
16. It will print the word sunflower in red
17. It would print the word sunflowers 20 times
18. I think it is going to print sunflowers 20 times
19. I think it will print "sunflowers" in red then go down 20 lines
20. It will print Sunflowers in red letters in mode 2 for 20 secs.
21. When you press run sunflowers writen in red will appear for 20 secs.
22. It will print sunflowers in red for 10 seconds
23. It will print sunflowers 10 secs.
24. It will draw a sunflower.

Year L4

Eleven of the thirteen lower 4 year pupil saw no error in the program as can be seen from the following explanations:

1. It will print synax error because there are no quotation marks on 20.
2. It will not print sunflowers 20 times down the screan because sunflowers is not in " "
3. Sunflowers will apear in red in the middle of the screen at the top.
4. It will print sunflowers 20 times in mode 2 in red
5. It will print 20 times sunflower in red
6. It will print sunflowers 20 times in red, and in Mode 2.
7. It will print red sunflowers on the screen 20 times
8. It will print sunflowers 20 times in red
9. It will print sunflowers 20X in red in Mode 2
10. It will print sunflower in read 20 times
11. This program will print sunflower 20 times in red
12. It will print sunflowers 20 lines down from the top of the screen in red
13. Sunflowers will be written in red and will stay on for 20 seconds

Year U4

Four of the eleven Upper 4 year used the expression 'nothing' indicating that the program would not run i.e. no error message suggested.

1. Nothing
2. Nothing as a mistake has been made
3. Nothing, no quotes
4. Nothing will happen because there are no quotes mark around sunflowers
5. It will print sunflowers
6. It will print sunflowers 20 times in a certain colour and in Mode2
7. Draw the sunflower in one colour
8. Sunflowers would be printed in red first but it can be changed into different colours as it is in mode 2 from colours 1 to 20
9. It will write sunflowers all over the screen in colour
10. The word sunflowers will be in red and flash down the screen
11. Have a coloured background with a sunflower.

In the two year groups 1st and L4th there appears to be high anticipation of the 'correct' version, 58% and 62% therefore totally ignoring incomplete structures such as the loop and incorrect order of commands. Not one subject gave the incomplete loop as an error. Errors were detected by 4% in the first year(quotation marks not present), L4 (quotation marks not present) 23% and U4 36%(50% giving no reason for error the other 50% giving quotation marks as error).

In summary the year-band and ranked responses show a similarity of thought across the age-bands. An analysis of data predicting a correctly running program i.e. the four items PRINT, SUNFLOWER, RED , 20 TIMES showed a reliable move away from agreement to a year-band influenced relationship. That is, different age-bands predict the outcome of the bugged program differently.

Finally, it has been seen above that, in the opinion of the subjects, the most significant error of the program is omission of inverted commas, therefore a further short study, based on a simple question was added.

The Role of Inverted Commas

Since more is happening than could ever be explained, explanation involves selection of what to explain. If we knew how selection is done we could do a better job of designing the information that systems provide so that explanation- building is made easier and more accurate(Clayton Lewis). To explain a computer interaction, the explanation is a description of a sequence of events, including what the user did and how the system responded.

For example, from the earliest exercises on the computer the pupils have carried out simple tasks including programming in BASIC. They are taught from the outset that a requirement for the computer to PRINT out a name or any other 'item' the name or item must be enclosed by inverted commas. In the bugged program study it was the only error identified and, as a result, it was thought to be an ideal opportunity to examine the novice explanation of what happens to the inverted commas between the keying in of the string enclosed by them and the string appearing on the screen without them. How would the novice explain their disappearance i.e. how would they equate what they, as the user, did and the resulting system response.

Seventy two pupils were asked to consider a small BASIC program which, on execution, printed a name onto the screen. They were reminded that in order to have the name printed on the screen it must be enclosed by inverted commas when input but when output to the screen the name was seen without the inverted commas - could they explain what had happened to them.

Results and discussion

The responses to the question from each subject can be seen in Appendix(41). On examination it was noted that the explanations could be grouped into five categories as can be seen in Table(51)in which examples of the category are also given:

Table(51)Language Part C:Inverted Commas Summary:

1. An explanation of why the inverted commas are present in the program:

   Yr1(5)"The speech marks tell the computer which part to print and not to print"

   Yr1(16)"I think that the speech marks just help in the printing and...."

   Yr2(20)".........because it is a command and is telling want to print on the screen"

2. The computer 'forgets' the inverted commas or 'they disappear'

   Yr2(43) "They disappear when you press return"

   Yr1(25)"I think that they are speech marks that dissapear because they are not part of the wording you are trying to print".

3. The inverted commas go back into the computer

   Yr2(19) "They go inside the computer because we dont want to see them"

   Yr1(12)"They don't keep them because it won't understand if you didn't. There essential. They go back into the computer".

4. The inverted commas go back into, or are stored in, the memory

   Yr1(11)"I think they go into the memory bank but it won't come back until we do a special thing"

   Yr1(20) "The "" go into the computers memory but you must put them there out of the computers memory"

   Yr1(24)"The speech marks dissapear into the computers memory and come back when we press the right button".

   Yr2(28)"Disapear in its memory banks to be stored for the programme"

5. Other responses

   Yr1(4)" The speech marks get blotted out"

   Year1(15)"Their blots go behind the screen when you press curtain buttons"

   Yr2(36)"I think the inverted commas becomes an error"

   Yr2(39)"They move off the screen sideways"

I wished to establish whether a significant relationship existed between the category of explanation response about the disappearance of the inverted commas and the yearband of the subject. In other words, can we expect a different mode of thought between the first and second year responses.

Categories 1 and 5, i.e. 'explanation' and 'other' were combined to keep the expected values above five and then the data applied to Chi Square kxn table. It produced $S_{(3)}$= 8.43 ($p<0.05$), X^2 indicating a reliable relationship between year-band of the subject and his/her explanation about the inverted commas, Appendix(42). Means produced by categories (1&5), (2), (3) and (4) across all subjects: 0.22, 0.25, 0.24 and 0.29 respectively.

If we accept that explanation plays an important role in understanding the conceptual model held (Clayton Lewis), then it is equally important that sufficient experience is gained in order for accurate interpretation of explanations to take place. In this particular study, for example, the two 'visible' steps are recalled; a name is typed as input at the keyboard 'marked' by inverted commas, it is followed by evidence of the system processing i.e. the appearance of the name on the screen without inverted commas. When the two steps are taken in isolation, it is understandable that the novice explanations about the disappearing inverted commas include "they disappear" or " they go back into the computer".

It is also recalled that during an earlier drawing study the subjects were required to explain graphically the 'unseen' computer process between keyboard and VDU, this similar task also included the printing a name onto the VDU. In that study no first year drawings of the system process included the removal or explanation of the inverted commas, one second year drawing contained an "inverted comma remover" Drawing(1) and a second drawing talks of picking up 10, print and "" but it is not established that they are the only items picked,Drawing(2). Finally, one third year subject also used a PRINT filter which includes removal of the inverted commas, Drawing (3).

The results of a Chi Square test established a reliable relationship between the type of response and the year-band of the subject. The most significant difference between the two yearbands is seen in the largest and only significant result in each year i.e., 41%, of first year subjects believed that the inverted commas go back into, or are stored in, the memory thus implying 're-use' matched against 36% of second year subjects who believed the computer 'forgets' the inverted commas or 'they disappear'.

The first drawing studies allowed freedom of interpretation and so contrasted greatly in content detail with the subsequent unseen processes study where the drawings became task specific. However, as can be seen from the above discussion the greater detail included very few clues about the novice beliefs on the role of inverted commas. Given the nature of the written explanations further support is given to the advantages gained by using graphical explanations. In this case a graphical study might have provided answers as to where in the computer the inverted commas 'disappeared' and, indeed, the process of restoration.

In the teaching and learning about the hidden processes of a computer system it is clear that language and appropriate terminology cannot be avoided. What we can do, however, is to look more closely at the choice of language employed and possible ways of linking the 'words' to imagable graphics.

The following part of this chapter looks firstly at whether the very young novice can, not only use an induced mental model to solve a problem, but also show recognition of the method used in his/her explanation. Secondly, it looks at the imagability of some of the terminology commonly used in the computer room.

LANGUAGE AND IMAGERY

Much of the earlier work of this Thesis is based on novice drawings which promise to project a view of how the subject, at different ages, is modelling his interpretation of the computer system.

To be able to develop and modify these ideas we need to establish that an effective line of communication exists between the learner and the educator. Among the pre-requisites for such communication must be, firstly, the subjects' awareness of his ability to generate and use images from the past to solve problems and, secondly, his ability to 'explain' the method of solution. If such a method of communication does exist then it can be used to develop an interactive learning pattern directed specifically at the effective evaluating, shaping and developing of existing mental models. Interactive, beyond the normally accepted sense required for voluntary learning, to the sense of the less used facilities, introspection and explanation.

The educator is likely, as a result, to find himself evaluating novice understanding of an 'event' or 'process' from the tentatively described, internalised account. Seen in these terms it would be useful, at least, to know something about the child's ability to work with and think about images cued by language; in particular, computer-related terms.

An appropriate method of investigation is suggested by Paivio(1975) who discusses the use of induced images or models in order to solve problems. Paivio said, for example, that he went about listing the names of colleagues working in the same building by imagining an internal view of the building and then walking along the corridors looking at the name on each door. He similarly recalled the number of windows in his house, this time imagining the exterior of the house; walking , in his mind's eye, to each side in turn counting the number of windows.

It was Paivio's statement of adult awareness about the conscious or unconscious use of mental images that led to the suggestion that image evocation could play an important part in the development, examination and shaping of mental models.

By using these ideas, the first survey simply asked the subjects to write down the number of doors they would pass on a particular, known route within the school building. No clues were given as to how they might solve the problem. This was an investigation, at a simple level, to establish whether children, when using a mental model, in this case a familiar route, would volunteer the information about their use of the model in the process of solving the problem in the explanation.

The second study recalls that the subject drawings included such objects as Keyboards, VDUs and Memory raising the question of whether their inclusion was affected by their imagability. It is possible, for example, that imagability constraints have a circular affect. That is, constraints may be implicit in the common components found in the drawings, in the whole terminology surrounding the novice and in the developing mental-model; the motivator of the subject's system drawing. In short, if much of the learning-teaching dialogue is language based, the actual existence or form of system components and the quality of mental model explanations may be dependent on their imagability.

Therefore, following the imaged 'route' survey, this second and final study looks at the imagability of a set of computer-related terms. The set of words is examined for the type of image they may evoke in the subject: 'Picture', 'Word' or 'Nothing' evoking.

Using and Describing the use of Mental Models

The objectives of this survey were to design and administer a questionnaire to examine whether young children are able, without prompting, to use and describe their use of a mental model to solve a problem.

Method

The subjects taking part in the survey were 181 pupils of Lomond School, Helensburgh who were of both sexes. They were drawn from four age-bands: 37 first year, 36 second year, 55 third year and 53 fourth year.

On introduction, the exercise was explained to the subjects who were then handed a questionnaire, Appendix(43), and asked to read the two questions carefully. To avoid any mis-understandings, the instructions were repeated verbally. The two questions were based on a known and familar route, within the school building, which began at the main door of the school and led to a room very close to the computer room:

(1)"How many doors are there on the corridor between the staff room and the steps to the reprographic room ?" and

(2) "How did you arrive at your answer to question (1)?

Results and discussion

Only the responses to questions (2), 'How did you arrive at your answer to question(1)' were classified. From the subject responses five categories were made possible (1) Guessed, (2)Remembered, (3)Visualisation of the route clearly indicated, (4) Actual route given, (5)spoiled paper or answer not classified elsewhere. Of the classifications category(3) only was accepted as a positive response.

The full table of Raw scores may be seen in Appendix(44) and the percentage incidence tables in Appendix(46). They show that 83.4% of the 181 pupil population included in their explanation the use of a mental model to select and retrieve required information.

It is assumed that the two categories 'Remembered' and 'Actual route given' were also visualised to bring about a solution, but, because the subjects had failed to identify the step in the problem solving process, it could not be treated as the correct and required communication. A selection of 40 positive, category (3) answers, may be seen in Appendix(45).

The high percentage result of 83.4% across the whole population is clarified in Table(52) which shows the breakdown for each year-band:

Table(52)Image use Survey:Percentage Category(3)

YEAR	PERCENTAGE
1	78.4
2	75.0
3	89.1
4	90.6

This survey has clearly shown that a high percentage of subjects of all four age-bands are able to explain the action of generating a model from memory and using it to 'pull out' or 'retrieve' information required to solve a problem. In addition, a large percentage of all age-bands do, without the need to be cued, include in their explanation the method used. i.e. generation of the model.

Table(52) above, shows a clear age-related development of explicit image inclusion in explanation except for year two, which dropped 3.4% against year 1. This drop appears to be due to the explanation 'from Memory' which, although it probably carries implicit-image use for these subjects, it could not be classified as a positive category (3). If, for example, the 'from Memory' explanations had been included the percentage result for year two would have been 83.3%, thus maintaining a continuous upward trend.

It was noted, from the experience of the survey, that school children have particular expectations of lesson format and generally feel no sense of anxiety about the task presented whilst the normal pattern is maintained. However, the survey seemed to cause some unexpected concern to pupils who regarded the actual 'number of doors' they gave in answer to the first question as a test of their knowledge and became anxious as a result when they were unsure of its correctness.

In conclusion, the survey which pre-supposed the use of mental images revealed that :

(1)Children as early as 9 years of age are able to consciously retrieve mental models of past 'events' for the analysis and solving of problems,

(2)A significant number of the children, 85% out of the population of 181, were able to write down a recognisable and vivid descriptive account of their use of a mental model. Indicating that, albeit at an elementary level, proof exists of a viable means of communication for mental model development.

Evoking Images From Words

The objectives of this study were to design and administer a questionnaire to determine whether subjects report the generation of a visual image in response to a selection of words used in computer studies. To allocate the responses to the three categories according to Picture, Word or Nothing evoking qualities of the word. To determine whether the reports are age-related and whether patterns of imagability are significantly common across the agebands, if thy are, to determine particularly directed words.

Method

A questionnaire was designed which contained ten words, as in Table(53) and three columns allowing the allocation of image description choice for each word, Appendix(47). The ten words were selected for the exercise on the basis that they had been previously introduced, and were in common use, during computer study classes:

Table(53)Images from Words:List

COMPUTER
VDU
KEYBOARD
MEMORY
VARIABLE
BASIC
ECONET
SYNTAX
STORAGE
BRAIN

The subjects, were those taking part in the earlier survey above.

Prior to the exercise, the subjects of each group were encouraged to discuss some every-day words and the way they re-acted to them. A variety of highly graphical words were chosen to encourage interaction within the groups. For example - 'snake' was selected and the subjects had very little difficulty in describing the snake they 'saw' when they heard the word. 'Kangaroo' was similarly chosen and also produced vivid descriptions of kangaroos including such details as 'babies in the pouch' and "mine's got very large legs!" 'Truth', however, proved more difficult because of its abstract nature; although pupils understood the meaning of the word well enough they were unable to 'picture' anything as a result. The one exception was the pupil who said that she saw an 'angel'. The general opinion was that when hearing the word 'Truth' they understood what it meant and the 'word ' was 'seen'.

It was emphasised that there was no right or wrong way for the subjects to respond; explaining that it was to be an observation of how they reacted to a word, a personal reaction. The question was raised at this point about what they should do if they saw a picture and a word. The advice given was that they should decide which was the first or stronger of the two and to select that; in general to select the first impression i.e. not to sit and think about it. Next a 'nonsense' word was explored. The general feeling was that the pupils had no reaction other than that they 'knew' they didn't know what the word meant and had not heard it before. We discussed the possibility of 'Other' as a response if the word did not produce a 'picture' or a 'word'.

The pupils were given the questionnaire, asked to read each word and to allocate it to one of the three categories, picture, word or nothing.

Results and discussion

The raw scores were tabled and ranked, Appendix(48). In order to establish whether the subjects were significantly like-minded across the whole population in their choice of picture,word or nothing evoking responses to computer related words, the data were subjected to a Kendall's Coefficient of Concordance 'W' . It produced W=0.8503 $(p<0.01)$ Appendix(49),indicating that the subjects across four age-bands are significantly like-minded in the way they classify lists of computer-related words to the three categories. As a results the top three most-picture imaged words are Computer 94%, Keyboard 88.4% and Brain 83.4% and the three least-picture imaged words are Syntax 12.7%, Basic 6.6% and Variable 5.5%.

Table(54) shows the 'picture evoking' words in order of percentage incidence:

Table(54)Images From Words: Percentage Imagability

Computer	94.0
Keyboard	88.4
Brain	83.4
VDU	41.0
Storage	26.0
Memory	16.6
Econet	14.4
Syntax	12.7
Basic	6.6
Variable	5.5

The particular words, Memory, Storage and Variable were chosen for the study because of the concept 'data retrieval, and its related concepts of 'variable' and of 'labelling' an item to be stored in Memory. They are concepts which have proved particularly difficult for many pupils to understand and to use correctly. Many GCSE candidates in their final year, for example, are still unable to explain the purpose of a variable and appear not to be helped by its common use in mathematics. It is clear that, in spite of regularly using small programs which contain variables as labels in loop structures and for data storage, approximately 25% of the subjects responded as if it were a new, therefore, unknown word.

If we look more closely at the results for these words in particular, see Appendix(50), the total non-picture evoking is :

Memory 83.4% Variable 94.5% Storage 74.%

taken from the two categories, 'word' and 'nothing' evoking :

Word : Memory 70.7% Variable 69.6% Storage 53.6% and

Nothing: Memory 12.7% Variable 24.9% Storage 20.4%

The high word-evoking scores for memory may be related to the way memory has been represented in the subject drawings. In almost all cases of components attributed to memory the component has been referenced by a 'word' label. For example, Memory Bank, Memory, Ram, Rom or by a label such as 32K; in no case has memory been represented in terms of its function. In contrast Brain, which cannot be defined as having 'memory' intent scored highly as picture-evoking at 83.4% and raises the question as to the form of picture evoked.

The results for VDU, SYNTAX and ECONET have also proved interesting. Many young, some older pupils and even a few adults still occasionally call the 'VDU' the 'television'. It is concluded that this is the case only because many home computers are linked to and use the television as a monitor. Whilst it is class policy to call the visual display unit the 'VDU' it is also called, the 'screen' or the 'monitor'. In view of the variety of names in use for the same item, it is surprising that one name does not act as trigger to evoke a visual image of another, if it is the more familiar name. However, 'VDU', expected to be highly picture-imagable, maintained a low 41%, lower than Brain, for example, which, it is assumed, the subjects have never seen in reality

The second example, SYNTAX, is part of a commonly generated screen error-message given on the BBC micro yet approximately 35% of the population gave a 'nothing' response and, in total, 87.3% gave a non-picture response.

Similarly, ECONET, the name of the net-work system used by the novices is frequently included in general discussion and also appears on posters in the computer room illustrating a ring of linked computers and yet 46% gave a 'nothing' response, and again, a high non-picture total of 85.6%.

In conclusion, this study examined a further dimension applicable to the mental model, that of the possible effects or constraints exerted by the terminology we use in the computing context. Firstly, for example, if some of the terms we use are preferred by the novice because they are more familiar, i.e. part of every-day language, then problems may arise if those terms have a hidden (to the novice) specialised meaning. Secondly, the imaginability of the terms used may play a part in the conception of the system and its components.

It is clear from the study that very few words used have clear qualities of picture-imagability; only three of the tenwords gained over 41% response from the 181 subjects. In addition, it is also clear from the study that we cannot predict an improvement with age, i.e. year-band of the subject does not appear to significantly affect the picture-imagability of the set of words.

SECTION THREE

MENTAL MODELS OF A COMPUTER SYSTEM: CONCLUSION

CONTENTS PAGE

SECTION THREE

CHAPTER 7

MENTAL MODELS OF A COMPUTER SYSTEM

The principle aim of this Thesis was to establish whether a mental model, i.e. a conceptual model, of a computer system exists in the mind of the child-novice, to investigate its nature and to establish teaching constraints capable of directing development and exerting modification.

The final chapter is presented in two parts. The first relates the findings of the empirical investigations to a useful, generalised conceptual model of a computer system. It extracts the essential 'pegs' from which to hang the relevant developmental constraints necessary to minimise the effects caused by mis-conceptions about the system. Finally, the generalised model is applied to a first level novice problem, the explanation of how a character pressed at the keyboard appears on the visual display unit, i.e. the problem that was set in the third practical drawing exercise.

PART 1 The Existing Model of the System

Drawings as a method of data capture

It was observed above that to seek information from very young learners about their 'mental models' of a computer system would present the researcher with a formidable task. The idea of trying to capture a 'snap-shot' of such a concept is, in itself, new and research methods available to carry out the task few and inappropriate. Drawing offers a 'safe' and enjoyable alternative; a method of expression which allows the novice-subject to transcend the barriers of language, spoken and written, as well as problems related to low confidence and self-consciousness about existing knowledge and understanding.

The three drawing studies carried out during the development of this Thesis have left no doubt that, as a method of data capture, it has established itself as possessing great value and potential. It provides a source of data which is unaffected by many of the inhibitions imposed by alternative methods and the value of data so collected has, equally, been shown in this Thesis.

These conclusions were further reinforced during the later stages of the research when, in the language based studies, the information gained from the subjects' written explanations appeared tauntingly inadequate by contrast. For example, when the subjects were asked to explain the disappearance of the syntactically required inverted commas, the answers were varied and gave insight into the modes of thinking, but, it was strongly felt that an opportunity had been a missed; a drawing may have revealed how and where in the computer inverted commas are returned and how in 'Memory ' they are stored.

The Subjects

This Thesis has dealt largely with subjects in the age-range nine to fourteen years, therefore, a large proportion of the population's cognitive development is firmly rooted in the concrete operations stage, (Piaget 1952). The implication, in terms of their imaging ability for example, is that they are progressing from the limitations of the static image to a higher level of imagery offering greater flexibility and mobility; capable of anticipating the successive movements of a 'yet-to-be' transformation from one state to another.

Further developmental effects are expected in the subjects' perception of picture-study problems and through it their method of solution. As a result, a large proportion of the population, as concrete operationals, are pre-occupied with the immediate realilty and the observable. They are capable of orderly and systematic observation, can observe and record a limited number of associations between occurances and non-occurance. A smaller proportion can be expected to exhibit some transitional characteristics of the formal operational, and they will be better equipped to handle the 'probable' and hypothetical projections about the system and its capabililties.

The transitional stages of development of the subjects are for example suggested in the growing 'complexity' of perceived and subsequently drawn system models; the increased number of items including objects from every-day life.

The Emerging General Mental Model

The three sets of novice drawings have revealed a Mental Model with distinct characteristics. It is a model which, I believe, forms the basis of a viable, and contextual interactive system . The features of the model have clear age-related, developmental features as described in the discussion.

The generalised subject model contained five 'top' components which maintained a 'highest-percentage' incidence during two different population studies. Their inclusion, that is, the use of 1,2,3,4 or 5 of the components in the model, is shown to increase with age, whilst the same ranked order of preference, Communication-link, Input/Output, Memory, Chip and Transport, is maintained throughout the three year age-bands.

It can be seen that the subjects' object set identifies with the three basic parts required by any computer system as seen in diagram(1) below, where the 'process', at this stage, remains concealed.

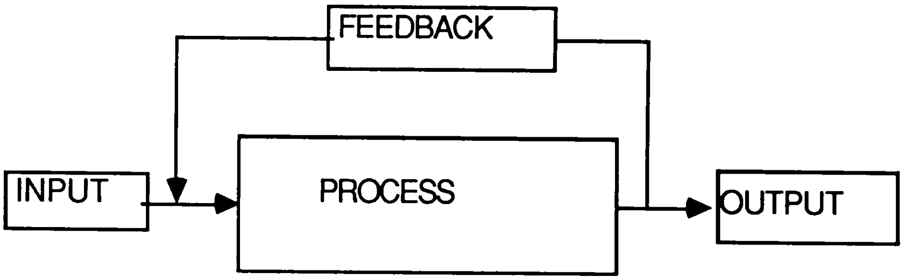
The 'processing' capabilities of the system illustrated in drawing(1) will be dependent on the components available to the model, but if, for example, we take an 'expected' model which contains a CPU(central processing unit) then the second simplest level might take the form of schematic diagram(2). In the second diagram we see that the first layer of the CPU has been peeled away to reveal the ALU, IAS and CU as well as a unit marked memory. This last part illustrating the deliberate intent to differentiate between volatile and non-volatile storage.

To briefly explain the units mentioned, the ALU, arithmetic and logic unit, is a set of electronic circuits where arithmetic and logical operations are performed. The IAS, immediate access store, is the volatile memory provision, in which each location can be written to or read from immediately, and is where the user program and data are stored during use. The CU, control unit, is another group of electronic circuits that has the all important function of controlling all the operations within the computer. Where not shown, the assumptions of diagrams include feedback and internal communication.

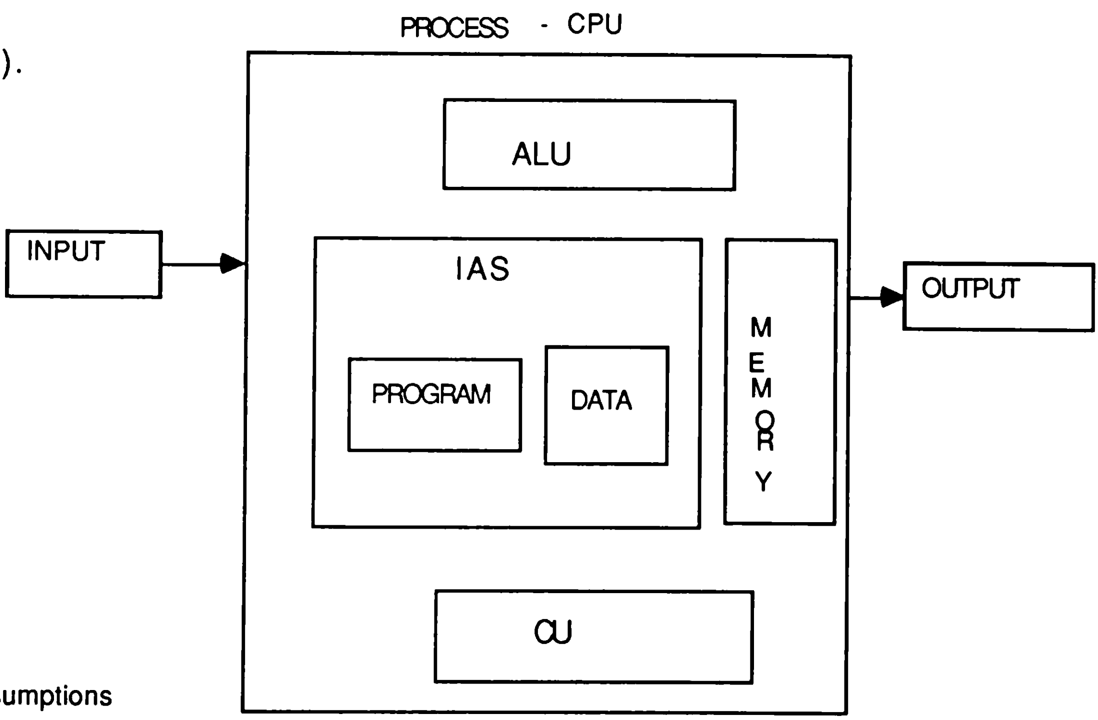
The general subjects' model components included Communication-links and transportation, not identified in diagram(2), leaving Memory and Chip, where 'Chip' has yet to be defined. Because the novice concept of memory is

SIMPLE SCHEMATIC COMPUTER SYSTEM

1).



2).



Assumptions

Feedback
Communication

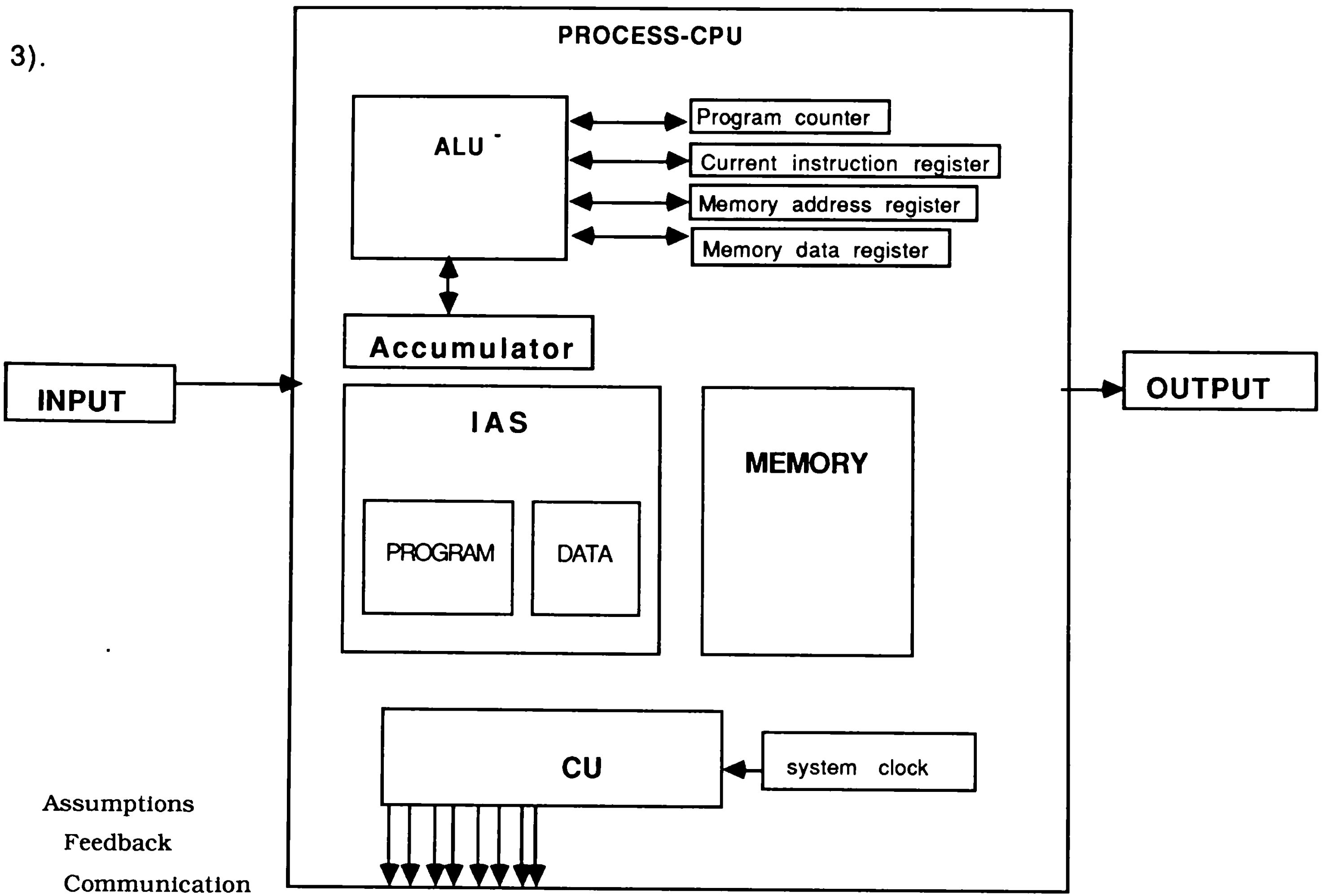
weak, and the processing is entirely dependent on memory, any initial models created for the novice must emphasise its important role, as indicated above.

Diagram(3) is suggested as the final complexity required of the general model because it reaches the detail necessary for GCSE Computer Studies. Here we are considering the task of the processor as it carries out a repeated cycle of operations. Nearly all computers today work on the Von Neumann stored program concept. That is to say, an instruction is fetched from memory, decoded(i.e. the computer finds out what the instruction wants it to do) and then executed. All operations inside the CPU involve this fetch-decode-excute cycle being repeated over and over again.

The CPU in diagram(3) has been opened up further to reveal the four registers interconnectiong with the ALU(defined above). A register is simply the name given to an electronic circuit that is capable of storing several binary digits. If you imagine that data travels in trains along little railway lines inside the CPU then the registers can be regarded as the stations where data is temporarily stored. It is the job of the program counter register to keep count of the place from which the next instruction is to be fetched. When an instruction is fetched from memory it is the job of the current instruction register to store this instruction. The memory address register holds the address of the data that is being accessed in main

SIMPLE SCHEMATIC COMPUTER SYSTEM page2

3).



Assumptions
Feedback
Communication

memory and the memory data register holds the contents of the last selected word read from or written to main memory.

Also interconnecting with the ALU, is a special register, the Accumulator, which appears for the first time in diagram(3). The ALU works in close conjuction with the accumulator which is so called because it is used to accumulate intermediate results. For example, it can be used to hold a number that has just been loaded from a specific location inside the computer's memory and various operations can be perfored on it.

The final new object to be introduced in diagram(3) is the system clock and its relationship with the CU. It is an accurate, very fast electronic timer which ensures that all chips which need to be synchronised are synchronised inside the computer. Because the BBC micro-computer is an 8-bit processor, all operations, including input and output, and internal transfers of data are in units of 8 bits, as indicated schematically from the CU in diagram(3).

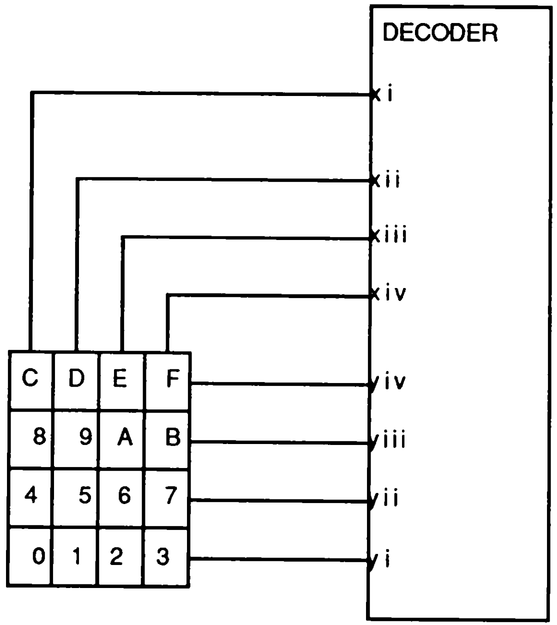
Having examined the basic objects required for the 'expected' model processing, and before communication is tackled, it is appropriate to add the Input/Output functions to the schematic model; it will be recalled that this element was one of the 'top' five subject component list.

The standard input device is the Qwerty keyboard but, for the purposes of this exploratory discussion, the hexadecimal keypad, which offers a manageable 4x4 keyboard, will be used, see diagram(4). The principle remains the same for other keyboards.

At this point attention is directed to subject Drawings(4,18,19,21,22&25) which clearly illustrate the concept of communication between input device, i.e. keyboard, and the computer. Drawings(4,18,19&21), for example, clearly demonstrate communication from individual keys, thereby attempting to relate cause and effect. Drawings(22&25) show lines of communicaiton which remain separate from one another whilst in Drawings(4,19&21), lines of communication are seen later to join, forming a single 'track' within the computer. Drawing(18) demonstrates another alternative with individual tracks coming together to run in parallel - although it is not known whether the use of eight tracks is by accident or design.

If we compare the subjects' concept of individual key communication to the 'expected' in the hexadecimal example which has lines of communication leading to the 'decoder', the similarities are clear. What is not made obvious by the subject drawings is how, when lines are merged to form one, identification of a selected character is maintained during communication, for example, to the VDU. The method in Drawing(22), applies an individual 'umbilical chord' which identifies the character and its route, whilst Diagram(11) shows an uncomplicated serial transmission of characters from keyboard to screen, both examples avoiding the problem.

4). INPUT: SCHEMATIC KEYBOARD REPRESENTATION

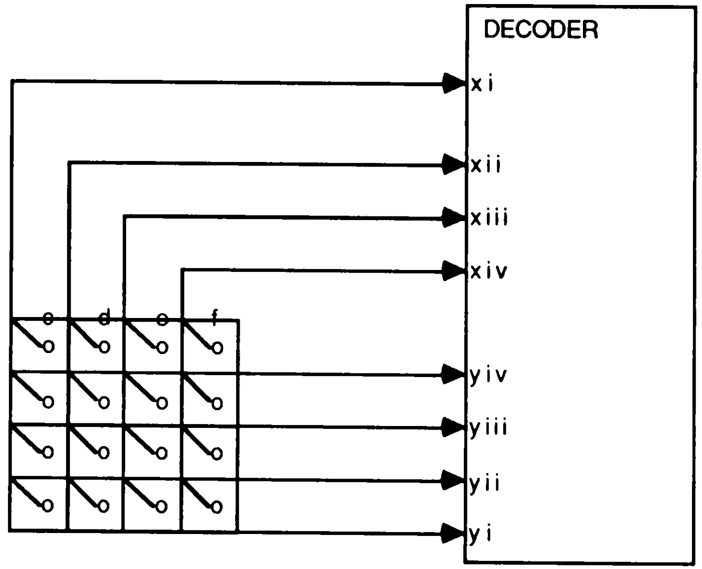


If a satisfactory model is to be developed to explain the cause and effect relationship of pressing a keyboard character, the matter of how the key press is identified and how that identity is maintained until it is transmitted onto the visual display unit is important. By removing the top layer of the hexadecimal keyboard, as in diagram(5), the method of identification becomes more obvious and is seen to be based on an XY matrix of communication lines which detect and identify keys depressed.

A keyboard is a set of switches arranged so that data can be input to the microcomputer system. The output from any keyboard or pad has to be encoded into a suitable logic signal before being applied to the microcomputer input. A qwerty keyboard as used by the BBC microcomputer will be encoded into 8-bit ASCII(plus 1 paritiy bit) code, using a special integrated circuit, a decoder, with built in ROM. In the case of our hexadecimal pad, shown in diagram(5), the decoder contains all the logic required to encode the array of 16 push-to-make switches into 4-bit binary . A Data Available output goes high to indicate that a key has been operated and returns to low when the key is released, even if another key is depressed. If , for example, the character 'A' is pressed on the BBC micro-computer keyboard, the character identification code would be 1000001 plus parity bit.

In some of the subject drawings a parallel is found for the decoder as an intermediate process between keyboard and screen For example Drawing(12) shows a sorting and a problem box, drawing(23) two

5). INPUT: SCHEMATIC KEYBOARD UNDERLAY



separate storage points are shown for (i)numbers and symbols and (ii) alphabetical characters. Drawing(17), in a further example, shows boxes of characters inside the computer which, after selection, require sorting-to-order or reduction in size before presentation to the visual display unit.

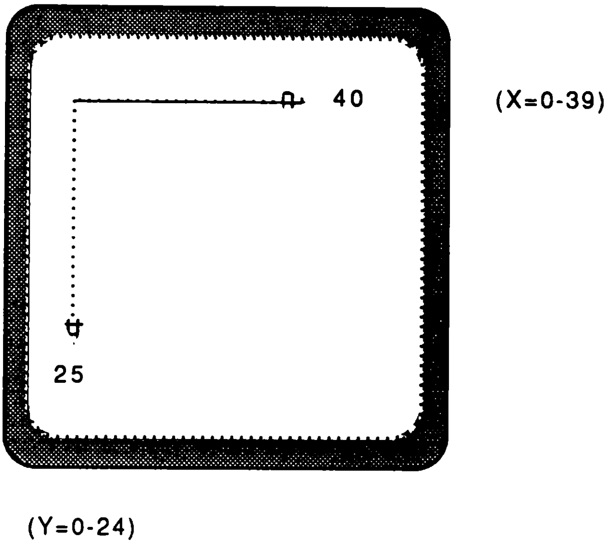
It is appropriate at this stage to move to the other end of the process model, to the Output function, because the nature and design of a communication model will be dependent on both Input and Output models. Unlike keyboard representation, the 'useful' subject explanations of Output to the visual display unit are very thin on the ground. At one level, illustrated by subject Drawing(11), a direct 'tube-link' runs from keyboard to visual display unit and characters are seen tumbling along a transparent tube between keyboard and visual display unit. When the characters meet the end of the tube ,which is behind the screen, they are 'expelled' from it and blown onto the screen. Drawing(21) illustrates the joining together of communication lines from key underlays, the single line continuing onto the screen where the input 'name' is delivered as a complete text-string.

The indications are that the 'natural' subject mental model carries a better sense of 'input' with implied communication than the output function and that the basis of weakness probably originates from the method of communication adopted by the subject. That is, the simplistic model which transports individual characters overtly, directly and serially from the keyboard to the screen whether by tube, wires or conveyor belt, is more readily able to explain the presentation at the screen than the 'mature' subject model of input which matches more closely the expected model.

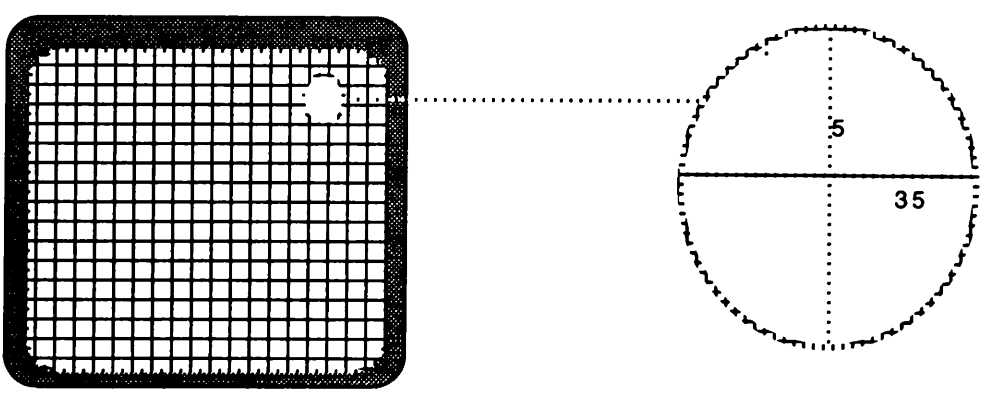
The 'expected' model of the visual display unit in diagram(6a) shows the approximate textual dimensions of the screen, (the 'X, Y' range) as {0-39x0-30} with the origin housed at the top left-hand corner. Diagram(6b) illustrates a schematic view of the grid underlay of the same text screen which is distinct from the graphics addressable system. The addressable screen grid becomes important when a particular position is required for the output being sent to the screen , that is, other than default values. For example, to print "hello" on a clear screen with default values produces the output at the top left corner of the screen, and each subsequent item would appear on the line below, i.e. Y+1. If output is required at a specific location, whether top-centered or bottom left of the screen, the 'print' item must be specifically addressed to that position using tabulation points. The enlarged section of diagram(6b) illustrates text postion(35,5) which would be addressed as PRINTTAB(35,5) "item", the actual location translated as 35 along the x-axis and 5 down the y-axis.

At this point in the discussion the emerging model possesses a keyboard, where a number of input lines of communication from depressed keys are channelled to a 'decoder' and the selected character is identified. The data identifiying the depressed character, the binary or ASCII code, is

6a).OUTPUT: SCHEMATIC VDU(mode7)



g).OUTPUT: SCHEMATIC VDU UNDERLAY



transmitted from the input decoder to the output decoder which prepares the data for transmission to the output device. At the output end the character is identified and encoded according to its seven segment display binary equivalent. With seven segments, labelled a to g, it is possible to display all the hexadecimal numbers and letters. For example, the maximum 7 segments are lit, i.e. 1111111, to create the number '8' output display where a loglical 1 is required to light one segment of the character.

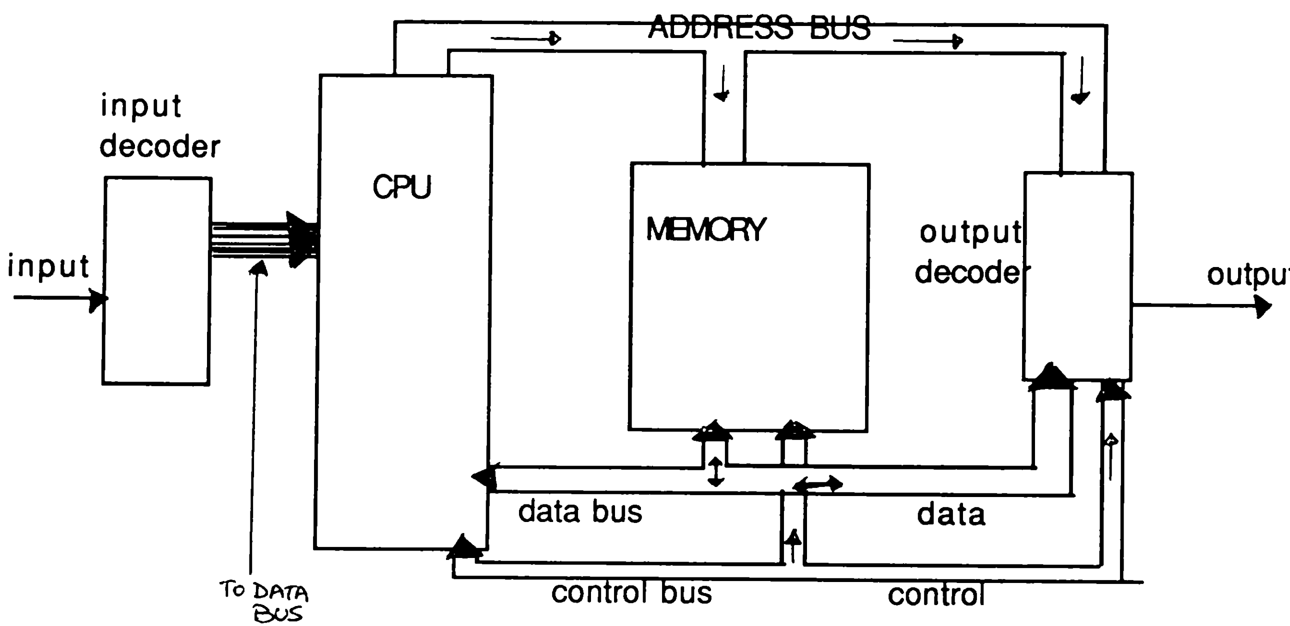
The character 'A', which was transmitted from the keyboard earlier in the discussion, requires 1110111, (segment d is switched off), as the required pattern of bits to produce its seven segment display. Each of the output codes is unique and is held in a 'lookup' table either in RAM or ROM ready to be outputted as required. The output code for our character 'A' is $77, the hexadecimal equivalent of 1110111, and it is $77 which is transmitted to the output device, in this case the visual display unit. On arrival, the hexadecimal value is decoded and further encoded to binary for screen instruction.

The 'expected' model is represented by schematic input, output sections and processing objects, and now requires the addition of communication links. The subjects' concept of communication within the system was strong for all age-groups, whether they chose to use roads, rail-tracks, conveyor-belts, tubes or wires, therefore a version of the 'expected' model should be suitable for all levels. The essential communication links within a computer system are established by the use of 'buses'. A bus is a group of parallel conductors inside a printed circuit board or wires which together form a digital highway, terminology which lends itself very well to the 'real world' interpretation made by many of the subjects in their drawings. The communications to and from the microprocessor can be divided into three groups, Control bus, Address bus and Data bus. The Control bus consists of a variety of important control and timing signals used between the microprocessor and the other devices which make up the system. It consists of lines such as RESET, ENABLE, READ/WRITE and so on, and controls the data flow. The address bus is uni-directional and is typically 16 bits wide, conveying address information from the microprocessor unit to memory as binary patterns. The data bus, on the other hand, is bi-directional and links all the main chips of a system to the data lines of the CPU. See diagram(7).

A Specific Model

In the drawing studies, the choice of components made by subjects for the system model was seen to change between the unconstrained and constrained model; a difference which carries important implications for the creation of any single conceptual model.

Generally the models showed, that as the subjects become older, they tended to 'clothe' the model with more components; reflecting changing perceptions of the computer's internal complexity. The additional components, where labelled, were identified as drawn from familiar, every-day objects

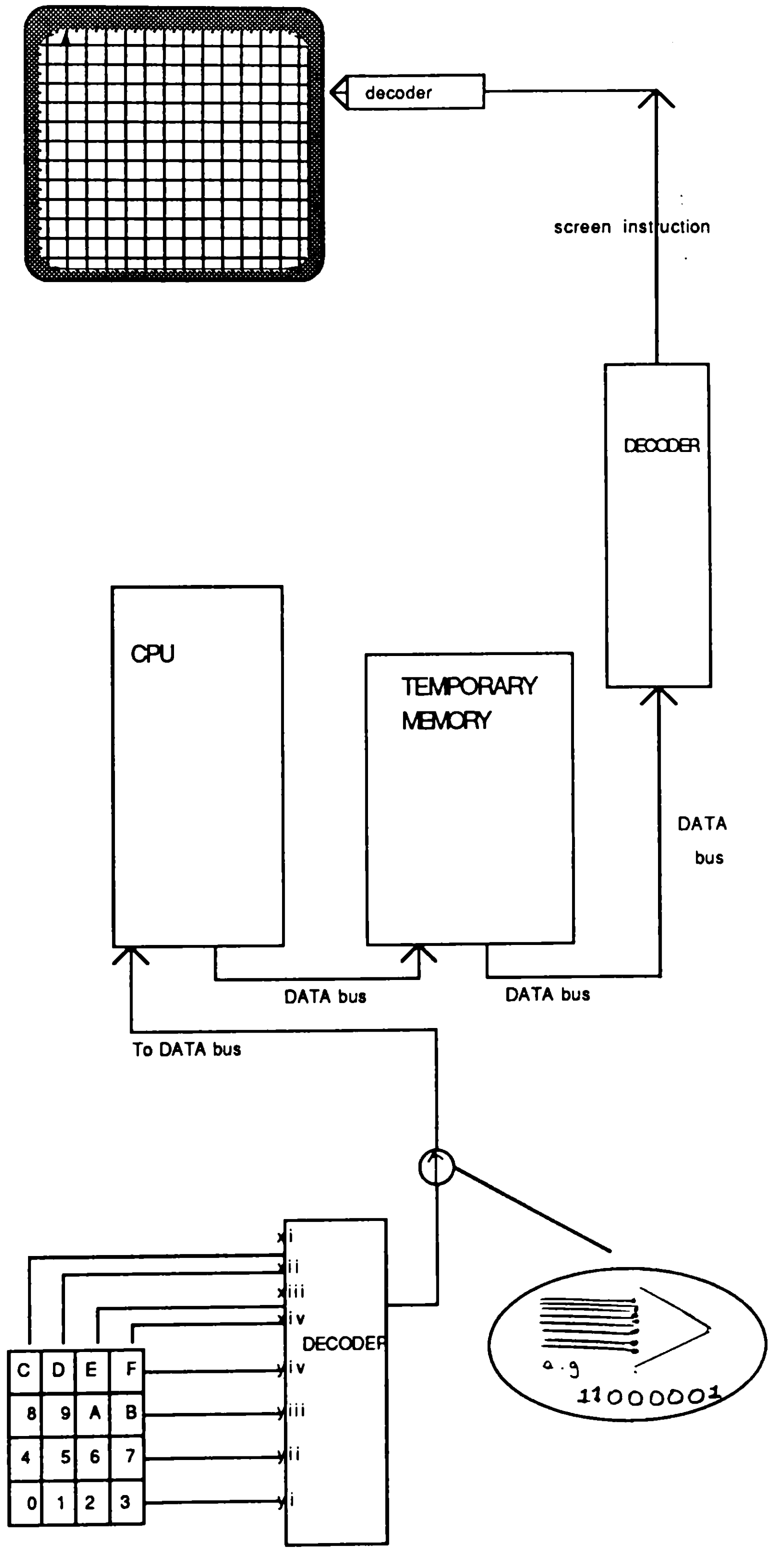

ADDRESS BUS
input
decoder
CPU
MEMORY
output
decoder
input
output
data bus
data
TO DATA BUS
control bus
control

including a wide style-range of fuses, power and internal screens. However, in the task constrained model the pictures changed in character becoming task- specific and, in general, included only those components directly related to the task set.

The indications are that no one model will suffice for the general computer user, it appears more likely that a conceptual model is constructed from a fundamental system base which is developed within the context of the application undertaken. A possible set of components for the general model has been assembled and described above and, from this standpoint, the actual form of the age-linked , task-specific model will be determined by the task itself and the target population, the components being covered, revealed and expanded as appropriate.

For the exploratory model, the problem is that set in the third drawing study, the explanation of the the transmission of a keyboard character from input to output onto the visual display unit which is seen as a first level model. Diagram(8) sets out a possible schematic format which required minor adjustment to simplify the keyboard selection and visual display models. The task of the decoders at both ends remain essentially hidden processes, interpreted simply as receivers and senders of character identifiers. At this 'first' level the role of volatile memory is introduced.

Memory allocation(not user addressed) is used here to give temporary storage of the character(it is assumed that program line numbers are not introduced at this level). The system using the memory component as a



8). KEY TO SCREEN TRANSMISSION - LEVEL 1

COMPONENTS: INPUT/OUTPUT, MEMORY, AND COMMUNICATION VIA DATA BUS scratch-pad which is easily wiped off when the next 'message' is sent. The cycle of the model involves the press of a key. The decoder receives the location code of the key pressed. The input decoder looks up the the code for that character and sets the new code on the tracks to the output decoder and a copy is sent to the memory unit. The output decoder receives the character code, decodes it, then encodes the 7-segment display equivalent of that character and sends it on the last part of its journey to the screen. This 'model' lends itself well to role-play with groups of children taking the part of the components, enacting temporary memory and using self-designed 'look-up' tables for the decoding, encoding and 'bussing' of data around the system.

In concluding the discussion it should be noted that any model development is incomplete unless, firstly, it includes relevant research into its validity as a behaviour predictor, that is, its components have reliable causal relationships and the system does what is expected of it. Secondly, that the model is capable of development from the simplest to the most complext required levels whilst maintaining its integrity. And, lastly, that the appropriate descriptive language is established. The concepts embedded in the system have much of their foundations rooted in language, and, because concepts emerge and are perceived through language, the ability to describe and predict the behaviour of a system using the appropriate language or terminology is essential. The 'specific' model discussed above is, in these terms, incomplete and has been used purely as a vehicle for model-component discussion purposes.

Conclusion

This research has established that a basic mental model of a computer system exists in a usable form, and that it is probably contextually affected.

The examination of drawings, predictions and computer related terminologly has revealed several important factors about the conceptions and misconceptions held by the child novice about the computer system. There are clear indications that cognitive development, attitude and belief structures play a more important role in the growth of novice understanding than most other school-based subjects. However, there are also indications that aspects of it may be positively influenced by appropriate experience.

A general system model has been suggested which is largely based on subject system components, supported by 'expected' model components where their inclusion is essential for the explanation and a reliable equivalent did not exist in the novice model, e.g.transmission of data to the visual display unit. Finally, a first level task-specific system projection was made from the general model as an illustration of its possible application.

PART 2: The Way Ahead With Mental Models

"The computer is incredibly fast, accurate and stupid. Man is unbelievably slow, inaccurate and brilliant. The marriage of the two is a force beyond calculation".

Cherne(1977).

It is the task of this, the concluding, section to relate the findings of the empirical studies to the problem as identified in the opening chapter.

The research results have shown that a mental model of a computer system does exist in the mind of the young novice and, that its form is suitable for use in the promotion, development and modification of system concepts. However, it has also been clearly indicated that there can be no 'one' conceptual system model which can be developed to aid learner understanding.

The results suggest the existence of a 'basic' model which is contextually influenced and adapted according to the task in hand. For example, the perceived system model for a wordprocessor is expected to be quite different to that of a Programming language model. Added to this are other variables exerting their influences on the novice model; for example, the stage of cognitive development, attitudes, beliefs, (including their explanation), and the effects of language.

A Computer Literacy Policy

It is clear that a complex problem with inter-dependent factors exists suggesting that a global approach and an adjustment of the way we think generally about the subject of 'computer education' is required.

This recalls an observation made within the first few pages of this Thesis. It was noted there that, because a first year secondary school pupil may expect to receive four or more years of computer experience above that gained at his or her primary school, we should recognise the responsibility for developing Computer Literacy.

Attempts to define this currently popular label were unsuccessful because no common set of rules or agreed selection of topics appear to exist. The definition "whatever a person needs to know about computers and, be able to do with computers in order to function completely in society (Hunter Seidel and Murphy) was typical but unhelpful. However, five of the list of ten computer literacy topics used by Cheng & Stevens, do appear to contain seeds relevant to points raised in the research. They are:

Computer terminology,
Parts of computers,
Having positive attitudes toward computers,
Being able to run a computer which includes running the computer software and,
Knowing programming language commands

suggesting that the set of twelve basic Constraints as outlined below could form the basis of a Computer Literacy policy.

An Examination of the Constraints

The grouped Constraints are followed by a related discussion and, although the approach appears 'linear' or 'sequential', it is only so for the purposes of exposition.

Firstly, Beliefs and Assumptions,

CONSTRAINT(1) The nature of the machine, its limitations and capabilities should be made explicit.

CONSTRAINT(2) Emphasis should be placed on the major role played by Memory for system and application software.

CONSTRAINT(3) Software used with novices should be selected on its ability to supply positive information about the system.

Some attitudes and beliefs exist before the novice ever meets the computer formally for the first time. Implicit in the definition of the conceptual model is the user's prior knowledge, beliefs, assumptions and expectations and it is clear that there are some powerful variables involved which can mitigate for or against the development of the desired robust mental model.

Beliefs and assumptions, good and bad, can become deeply entrenched and difficult to modify unless identified early in the 'system-learning' process. For example, given a CD player or 'music centre', the novice user may agree that the equipment is complex and difficult to use for the first time but he is unlikely to make the assumption or hold the belief that the equipment possesses an intelligence greater than or equal to that of a human being. From studies carried out we can predict that as many as 60% of children between the ages of 9 and 13 years of age do hold such beliefs about the computer. The surprising factor being that the percentage of pupils who hold this view increases with age i.e. 52% in the first year followed by 63.2% in the second year and 65.1% in the third.

Such beliefs, together with the lack of awareness about system memory, are thought to have a significant and detrimental influence on the novice user's sense of being 'in control' and are actually nurtured by system 'evidence'. My particular experience suggests that the belief is perpetuated by the "second-personness" of the interaction required for logging-on to a system with shared resources, and is further promoted by the nature of the interaction employed in most application software.

For example, in the school environment the wordprocessor is normally the first application software used with young novices, making available ample opportunity to lay down negative as well as positive attitudes and beliefs about the system. Owen(1986) points out that even the youngest pupils will be familiar wilth the concepts of characters and their fixed positions in relation to one another on a page, a line and how long it can be, the limits of the paper in terms of size and shape and also the physical management of paper and pencil.

Not least in importance is the child's manipulation of the actual characters on the writing medium which is direct and with the child in immediate control. In contrast the computer system alters the parameters as it manipulates the character representation; the software rather than the novice is in direct control. In addition, the novice has to discover the volatile nature of their work representation as it is manipulated on the screen; often proving to be a painful and demotivating lesson.

The list of negative points is incomplete but sufficient to support the argument for avoiding the use of a word-processor during the earliest stages of novice computer use unless sufficient preparation can be carried out in advance. For young novices the software itself should always, in my opinion, be viewed as a vehicle for the development of computer literacy rather than as an end in itself. The content should be subordinate to the information it makes available to the novice in order to build understanding about the system.

The syllabus designed for the young novice computer user should be accountable for the points introduced above. It should be based on deliberate and explicit guidelines to firmly place the computer into context as a realistic machine with clear cut capabilities and, perhaps more importantly, limitations. In use it should be emphasised as a resource with major dependency characteristics based on memory and instruction.

In conclusion, the development of healthy attitudes and beliefs about computers will be influenced by two complimentary areas. Firstly, the creation and maintenance of a psychologically 'safe' environment (Rogers1954) to enable the novice to learn freely and creatively, and secondly, the observation of all that is done with , and all that is said about, computers by others in relation to the novice learning.

Secondly, the System Appearance:

CONSTRAINT(4) Whenever possible novices should be introduced to computer concepts on a system representing the best available design for promoting a good system image.

CONSTRAINT(5) Full introduction and familiarisation procedures should be a standard to the introduction of a computer system. Each object of the configuration being covered in terms of its external attributes. There has to be understanding of the tool before it can be successfully used as a resource.

Experience of the system includes all aspects of it including the physical appearance, the nature of the interaction as well as the documention and instruction.

For most of us the choice of system is a 'fait accompli' and the constraints of the existing machine design must be accommodated. Even those in a position to make a choice about which system to purchase for school use will find little choice and no solution yet to the current problem.

At the simplest configuration level, the physical differences between systems may be very obvious. For example, CPU combined with keyboard and separate monitor and disk drives, or CPU combined with disk drives and separate monitor, keyboard and mouse, or CPU combined with monitor and disk drives with separate keyboard and mouse or, any of the preceding plus network facilities.

Systems which have combined CPU and keyboard(BBC micro computer for example) can cause confusion as to where the 'computer' actually is and leads to the common situation of children referring to the VDU as the computer; a point supported by some of the drawings. The error is understandable when comparing the appearance of the commonly used school VDU to the standard television which is a self-contained unit needing no supporting peripherals.

Whilst young novices are less likely to be concerned with the basic ergonomics of keyboard design than their adult counterpart, its layout is a problem high on the 'external differences' list. Apart from the 'concept' or similarly simplified and modified keyboard for the very young and disabled, little has been done to research and improve either the appearance, function or descriptive labels of the keys. In terms of keyboard layout, the young novice is faced with a bewildering assortment of keys; different shapes, sizes and names. The fact that amongst them exists an array of alpha-numeric keys standard to most typewriters is of no significance to the novice nor are the concepts of uppercase characters and dual, or even triple, function keys requiring the user to learn special key combinations.

Rather than the combined CPU and keyboard used by the majority of primary and junior schools the 'clip-on' style of keybard used by most modern business systems would allow the interchange of a simpler key board for the early years or stages with a more complex but still 'sensible' keyboard for the more experienced user.

Accepting the conditions which might exist in any teaching environment, very little time is normally devoted to allowing the novice to become familiar with the computer equipment. As mentioned above, the pupil brings to his first experience the 'attitudes and beliefs that existed before he ever met the computer' and the meeting will modify those attitudes and beliefs negatively or positively. The concrete-operational, in particular, will benfit from a careful introduction to the system.

The system appearance, that is the external objects and behaviour which make up the stand-alone system, should be seen as the first step on a continuous training programme in observation. For example, the basic three parts of the system input =>process =>output are concretely observable as are the effects of system cues and the loss of any one of the system parts. Learning to link cause and effect can be promoted. The concrete objects, their names, role within the system and overt behaviour may be used to lay the foundations of useful discussion, development of correct terminolgy and the sense of 'knowing' designed to enhance the novice user's self-image and being 'in control'.

CONSTRAINT (6): A component model should be developed to supply the first level concrete, domain-specific framework, which leads to other levels of complexity.

CONSTRAINT (7) : Where appropriate the component model should reflect simplicity and visibility

CONSTRAINT (8): The component model should contain clear boundary conditions.

CONSTRAINT (9): The component model should contain all essential and available clues.

It is said that novices tend to lack domain-specific knowledge(Greeno 1980, Simon 1980;Spilichy, Vesonder, Chiese & Voss 1979) and provision of a domain-specific framework is thought to supply the necessary 'availablity' for effective assimilation of new information. Concrete models were demonstrated as effective advance organisors in learning new scientific information by Roger & Cabel(1975,76).

For example, subsequent learning was influenced by the use of a chain of falling dominoes as a concrete analogy of electrical conduction. A simple idea generating a powerful metaphor.

Any idea developed for clarifying the hidden operation of the system should have the two properties suggested by duBoulay et al(1980):

> (1) Simplicity: there should be a 'small number of parts that interact in ways that can be easily understood and
> (2) Visibility: novices should be able to 'view' selected parts and processes of this notional machine in action'

The component model must also be valid (Riley) where validity does not depend on the user model being identical to the system model but it must be sufficiently robust to enable the user to explain and predict correctly, i.e. the user's knowledge must be consistent with the system behaviour. This 'how-it-works-knowledge' selected to be relevant to the user's task, facilitates learning, provides predictive and explanatory power, increases the likelihood that procedures will be remembered or can be regenerated and enables the transfer of skills.

A simple concrete model of the observable system, is envisaged as a second phase which follows the physical introduction of the system. Seen as a model of 'what is already known', this component model of the conventional standalone system should be capable of displaying on/off states and cue states by appropriately positioned illumination. The specific conceptual development required falls into four categories:

1.The ability to explain the system: its dependencies and inter-dependencies
2. The ability to predict an outcome i.e. ready to Run, or not ready to Run, given a set of system conditions.
3. Given that the system is not ready to Run, the ability to analyse and solve a conditional system problem.
4. The ability to explain visual, aural and temporal system clues.

For the third phase the novice user appears to be required to make a substantial conceptual leap into the dark when moving from the relative safety and control of a stand-along system to a network. However, with an understanding of on/off states and cues as discussed above, it is proposed that the knowledge gained can be transferred as a unit onto the network with only a minor adjustment.

Physical examination and experience similar to that used to introduce the stand-alone configuration should be carried out with the network prior to the introduction of a 'concrete model'. The second 'global' component model, should be clearly recognisable as the system it models and so constructed to accommodate re-arrangement of the model to suit varying situations.

The specific conceptual developments required from the second model stem directly from the 2-way nature of data transmission; the 'conversational' mode used on a network between the stations and a 'distant' control point, the resulting difference in the relationship of a one-to-one and the many-to-one interface, and the identification of stations on the network as separate from user-identification. The requirements are:

1. The ability to explain the system: its dependencies and inter-dependencies
2. The ability to predict an outcome, system ready to Run or not ready to Run, given a set of system conditions
3. Given that the system is not ready to Run, the ability to analyse and solve a conditional system problem
4. The ability to interpret and explain system cues.

It is hoped that the models developed will enable the novice to extend existing knowledge but more importantly, identify and modify gaps in understanding. The aim is to enable growth of the ability to handle the unexpected and 'what-if' situations which can be exploited fully only when the novice feels safe and in control of the situation. In addition, a more positively active and experimental role in the user/system partnership will enable the vital 'repair' functions to effect growth and error-correction of the mental model of the system. The conceptual model is intended to set the scene and direction for the construction of a useful mental model which can then be revised by recursive and iterative methods.

Opaque systems such as computer systems, where the essential processes are hidden, have already been described elsewhere as conceptually multi-layered, rather like the skins of an onion, a Russian doll or a nest of 'black boxes'. The particular opaque system considered here is a network of computers, with shared resources, managed by a file-server.

As a result of system opacity we are faced with what Seely Brown(1986) describes, in his discussion about informational systems, as a lack of causal metaphors for explanation. He clearly differentiates between the kinds of explanatory metaphors required firstly for mental models of mechanical systems, which can be constructed with metaphors built on every-day physical life experience, and secondly for the informational system, which has little in the way of ready-made conceptual foundations on which to build. Initially the novice is presented with Owen's(1986) concept of a 'bag of groceries' which, as he points out, would be of limited value if we could not also unpack elements of that concept. However, it is clear that if the system functions are layered in terms of conceptual complexity or 'shells of competency' (Hollan), then the system must be revealed in timely and manageable conceptual 'bites' (Piaget).

At the stage when the novice is exhibiting transitional characteristics of formal operations opportunities will be created to extend system understanding. When he or she is thoroughly familiar with and is able to discuss, analyse and predict the overt behaviour of the system it is time to introduce the concept of what the component parts do.

The problem of how best to present a conceptual 'model' of the hidden processes within a system, particularly for young novice learners, is a research topic in its own right. However, as a result of this research the 'expected' components and the 'weak' characteristics are known and can, therefore, be promoted.

At an elementary level, and if the correct terminology has been used, it would seem a natural progression to 'open up' the familiar model to a first level revealing the fundamental components communication, memory and input/output functions, already predicted to exist in the mental model.

The problem clearly becomes more complex as cause and effect relationships are taken down through the conceptual layers of the model and characteristics of the model themselves are influenced and changed by contextual use. It is most likely that future research will investigate methods of using the computer with methods of artificial intelligence to manage complex learning in suitable 'bite-sized chunks'. The method would enable the user to investigate conceptual levels appropriately with the content, pace of progress and monitoring carried out interactively on a one-to-one basis by the computer. "The computer is very patient and provides no opportunity for personality conflict with its user" (Ahl 1976).

For example, one technique is already available which uses the Apple computer and the software "Hypercard". It allows learning materials to be created which are founded on databases of information. Following the concept of 'layered' information, it offers illustrated 'cards' of information which contain 'buttons' . A button, previously programmed, can be selected to obtain further screens of graphical, textual and aural information related to the selected area. For example, the first layer model would contain a Memory component with an attached button. By selecting the button the user would be taken 'into' the Memory unit, which may contain components of its own designed, for example, to descriminate between various memory tasks(system and user etc.) each, again, may have attached buttons to reach further levels containing specific registers and so on.

Thirdly, the role of language and imagery

CONSTRAINT (10): The component model should be developed hand-in-hand with a sense of the role-related nature of specific computer-related terms and how the role could affect the model image.

CONSTRAINT (11): The component model should be developed with the use of images: using induced images to solve problems, in verbal and role play form, giving full opportunity of 'talking through' the problem-solving activity.

CONSTRAINT (12):The component model should be developed with the use of appropriate words of high imagability and novices encouraged to recognise and discuss the system in terms of mental images.

Johnson-Laird(1983) draws attention to the fact that the human communication is complex and tends to suffer from literal failings which often depend on the listener recovering the speaker's referential intentions.

Referential intention is not in isolation under the umbrella of language related phenomena. The language used to explain computer related objects or processes to the novice will contain a mixture of vocabulary; some carrying prior meaning and some new. Whilst ease of learning is dependent on the extent to which objects, relations and language in the new system can be integrated into the existing components of knowledge it is not necessarily beneficial. Lewis(1986) points out that learners unfamiliar with a domain often make connections that are not valid and students may mis-interpret or distort information to fit their naive views.

It was found, for example that the role-sense about computer-related words, in terms of Control, Command, Input/Output, Memory and Data was weak with the majority of terms regarded at a simple level, that is, as Commands. Fewer, fundamental mis-conceptions may occur if the important terminology or language structure where relevant is set in the context of the system model appropriate to the stage of novice-development.

The implications of 'language' are extensive and include the use of all vocabulary whether verbal or written and beyond to areas such as keyboard labels as illustrated above. Studies carried out have shown that young novice are well able to use their strong imaging abilities with objects given in verbal form, but will be adversely affected by words which cannot be picture-imaged.

The response to such words as computer and keyboard evoking strong 94% and 88.4% picture responses respectively, whilst a concept such as 'memory', a proven weak link in the mental models held by novices, was evoked by only 16.6%. This in itself suggests that memory is an 'empty' concept, portrayed as it is, a labelled shape or box, with no apparent functional reason for its inclusion in the drawings. Similarly, Econet, the system most commonly used in School networks, is also weak at 14.4% but 'SYNTAX ERROR' the commonest error message produced by the system produced an even lower score of 12.7%.

One method of enhancing the development and reinforcement of accurate early models is seen in the use of induced images and role-play; thereby combining the natural strengths shown in the former with an activity which comes naturally to the younger novices and at a time, for them, of least inhibition. The study of induced images for retrieving information for problem solving showed that over 80% of the novices examined were able to describe their use of images without direction. Therefore, to overcome the problem of low or non-imagability of key words there should be a deliberate policy to 'create' and to 'use' relevant images; making their use explicit at all times.

Conclusion

As established in this Thesis the acquisition of user mental models of a mechanistic system is a natural process arising from 'making sense' of the physical structure of the device and observing it in action. However, the complex nature of the mental models required for a computer which has an opaque system , is conceptually multi-layered and multi-process suggesting that the natural processes need to be given a helping hand.

There appear to be many contributing factors in the development of a mental model. For example, the conceptual layers phenomenom is one of them and is clearly manifested in the relationship which exists within the system, that is, between the physical components and the software. This relationship is responsible for giving visual, auditory and temporal clues about the opaque system.

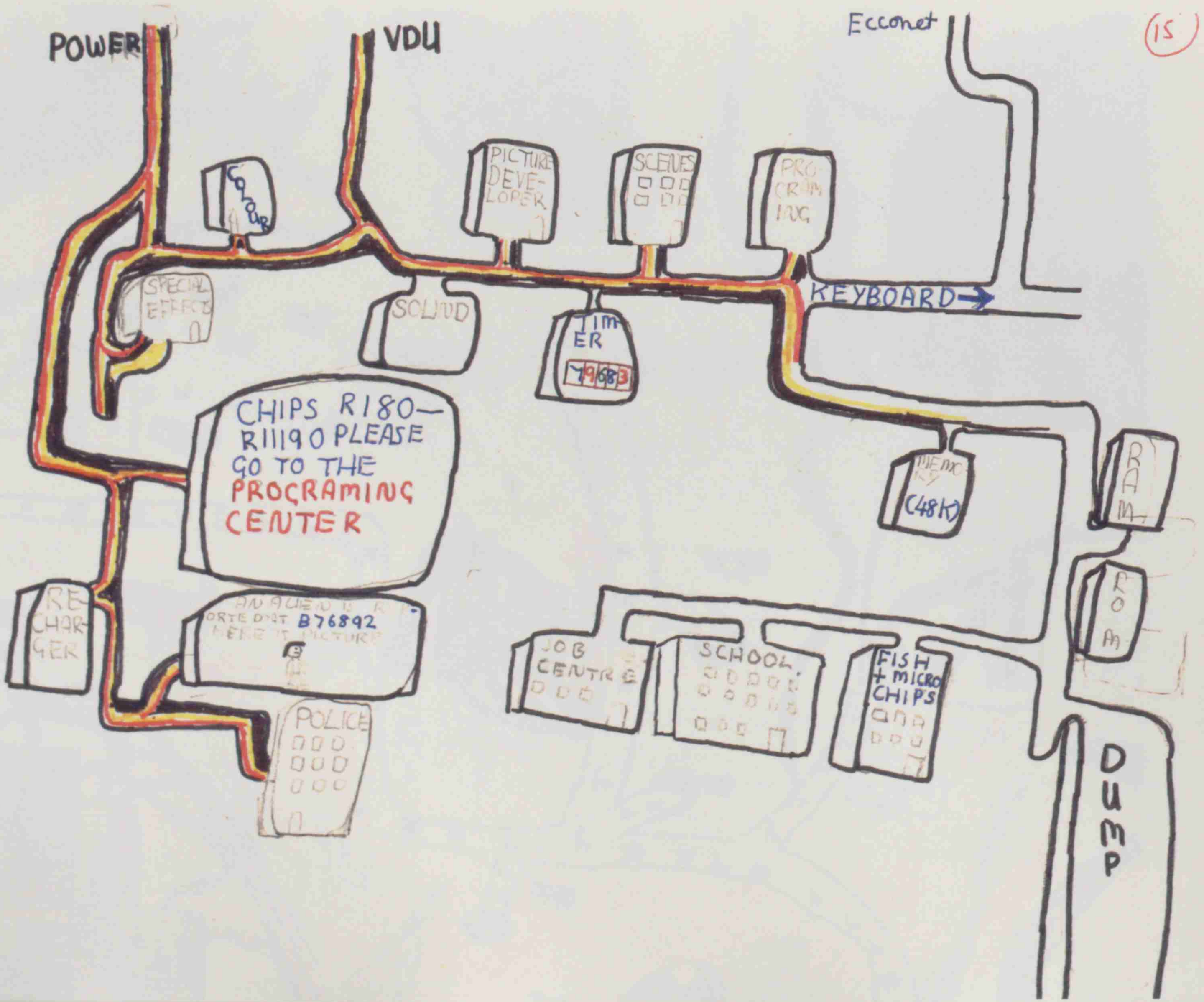
As a result of this research the conclusion has been reached that the way computer education is perceived and taught in school requires modification if learning with greater understanding is to be achieved.

The generation of healthy and realistic belief structures about the computer is essential, as is an environment in which the novice feels in 'control' and is encouraged to understand and investigate. Concrete models, development of language through role-play, imaging and explanation are combined to enable the concrete-operational child in particular to develop an under- standing of the 'total' system.

The under-standing and development of useful conceptual models is reflected in the growing ability to explain the system component dependencies and inter-dependencies; beginning with the observable system. The novice should develop the ability to predict action outcome, analyse and solve problems based on conditions of the system and to interpret and explain the system cues: visual, auditory and temporal.

DRAWING REFERENCES


CHiP 513
ROBOT WARDEN
32K MEMORY
WorD FACTORY


POWER
VDU
Ecconet
Colour
PICTURE DEVE-LOPER
SCENES
PRO-GRAM-ING
KEYBOARD
SPECIAL EFFECTS
SOUND
TIM-ER
MEMO-RY (48K)
CHIPS R180—R11190 PLEASE GO TO THE PROGRAMING CENTER
RE-CHAR-GER
B76892
POLICE
JOB CENTRE
SCHOOL
FISH + MICRO CHIPS
RAM
ROM
DUMP

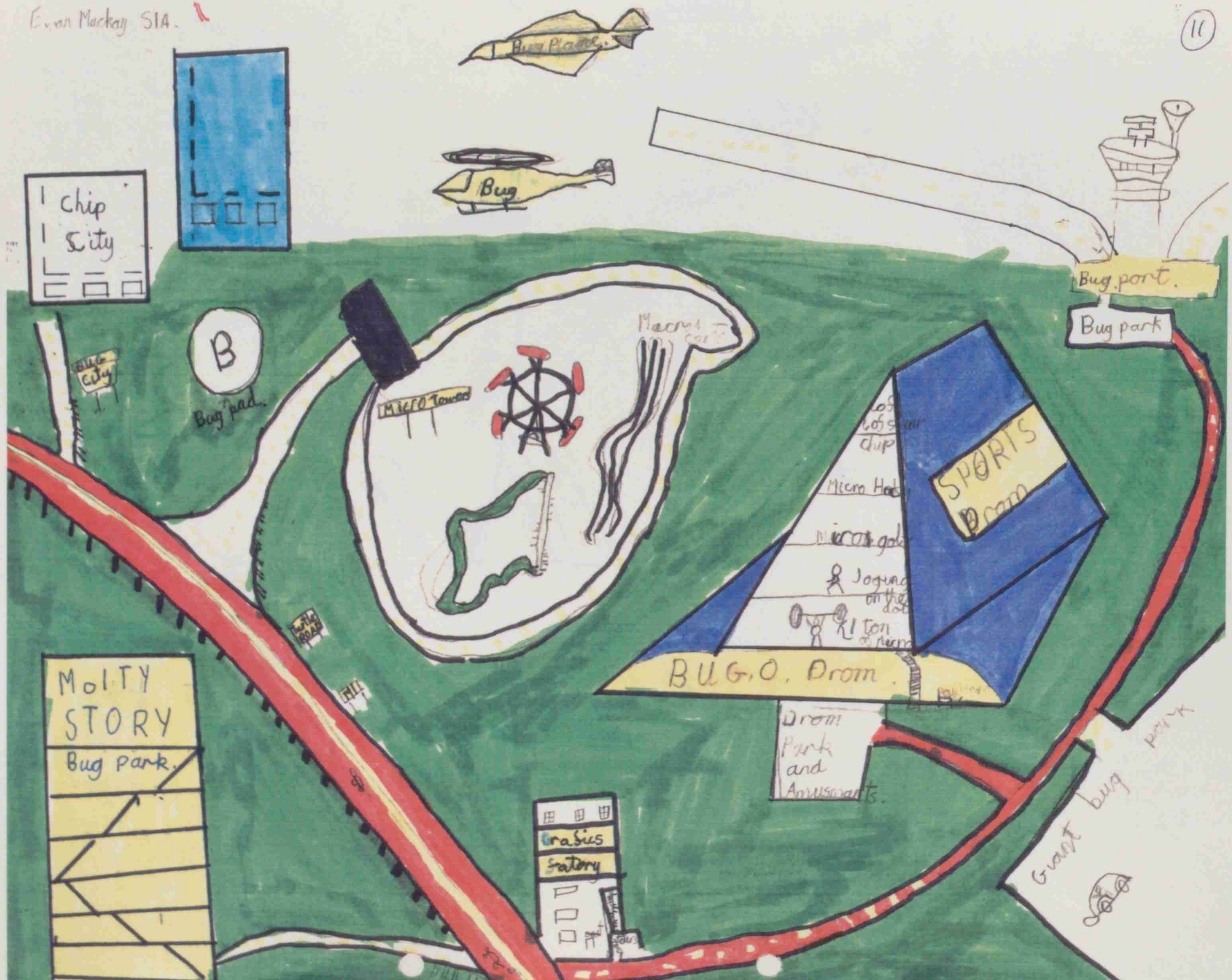
15

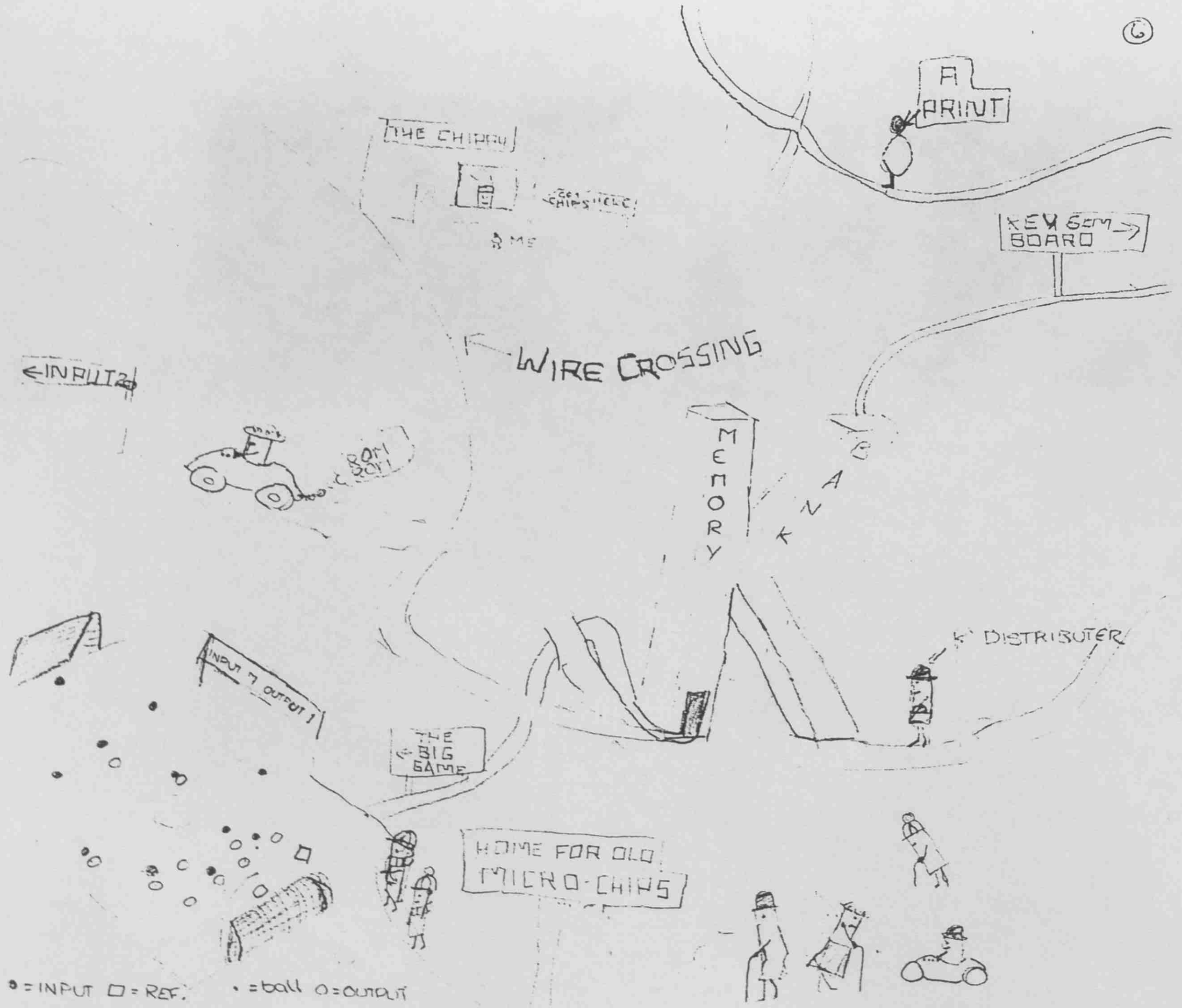

TO DUMP
TO HELL
TO HEAVEN BANK
CHILDRENS PLAYGROUND
TO HEAVEN


SORTER
GRAPHICS
MONOBUS
BASIC
ROM
RAM

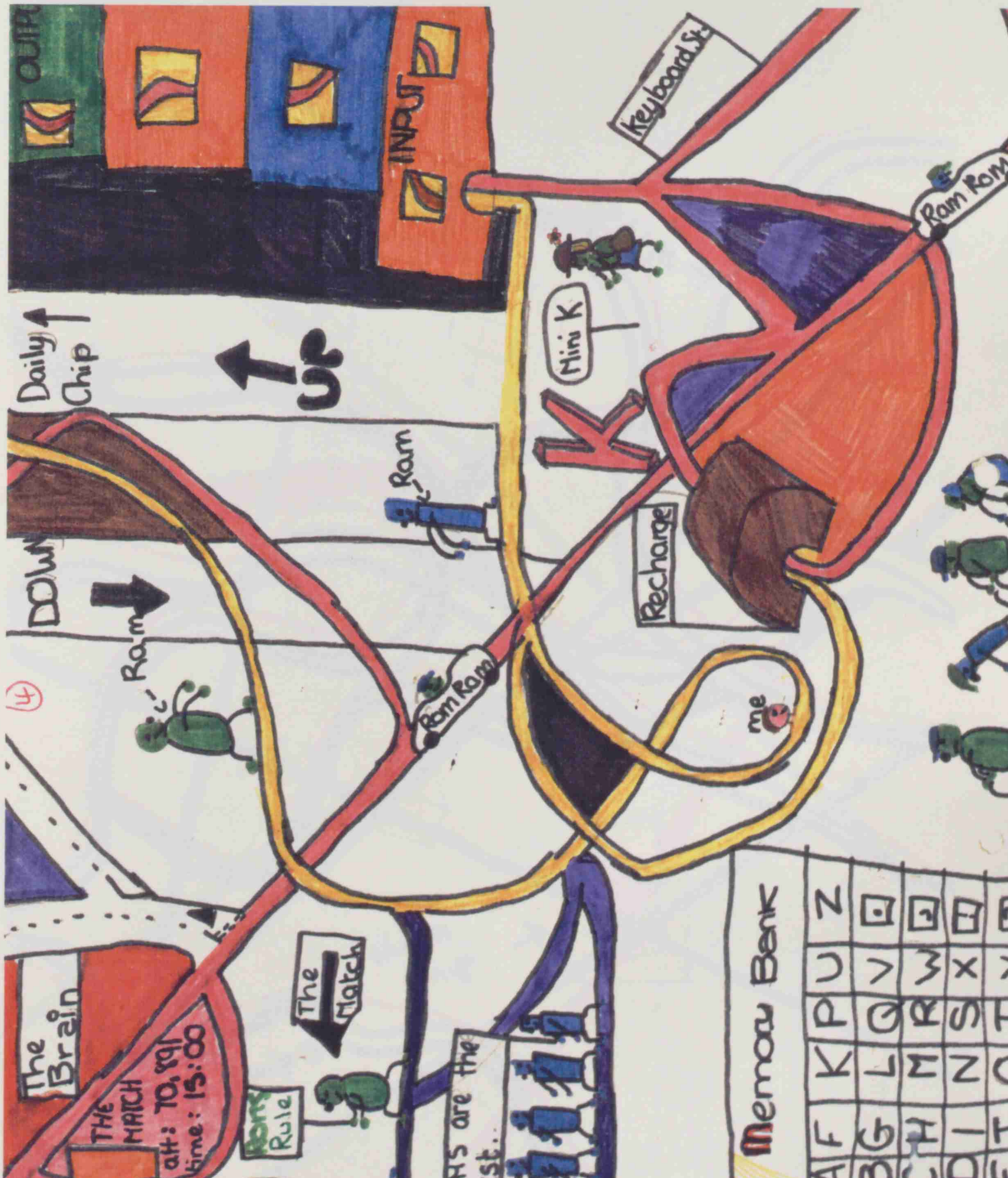
Evan Mackay 51A.

(11)



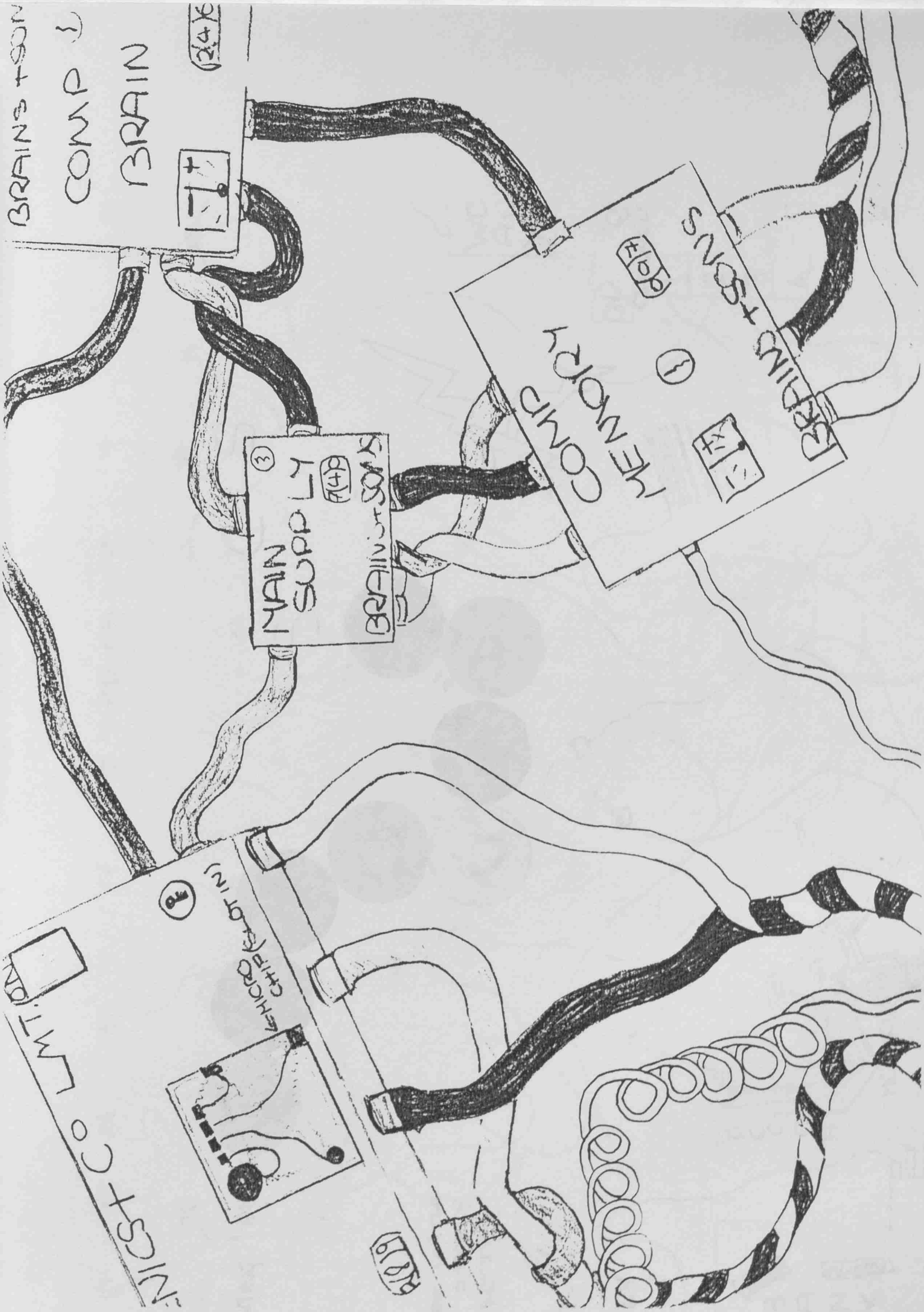


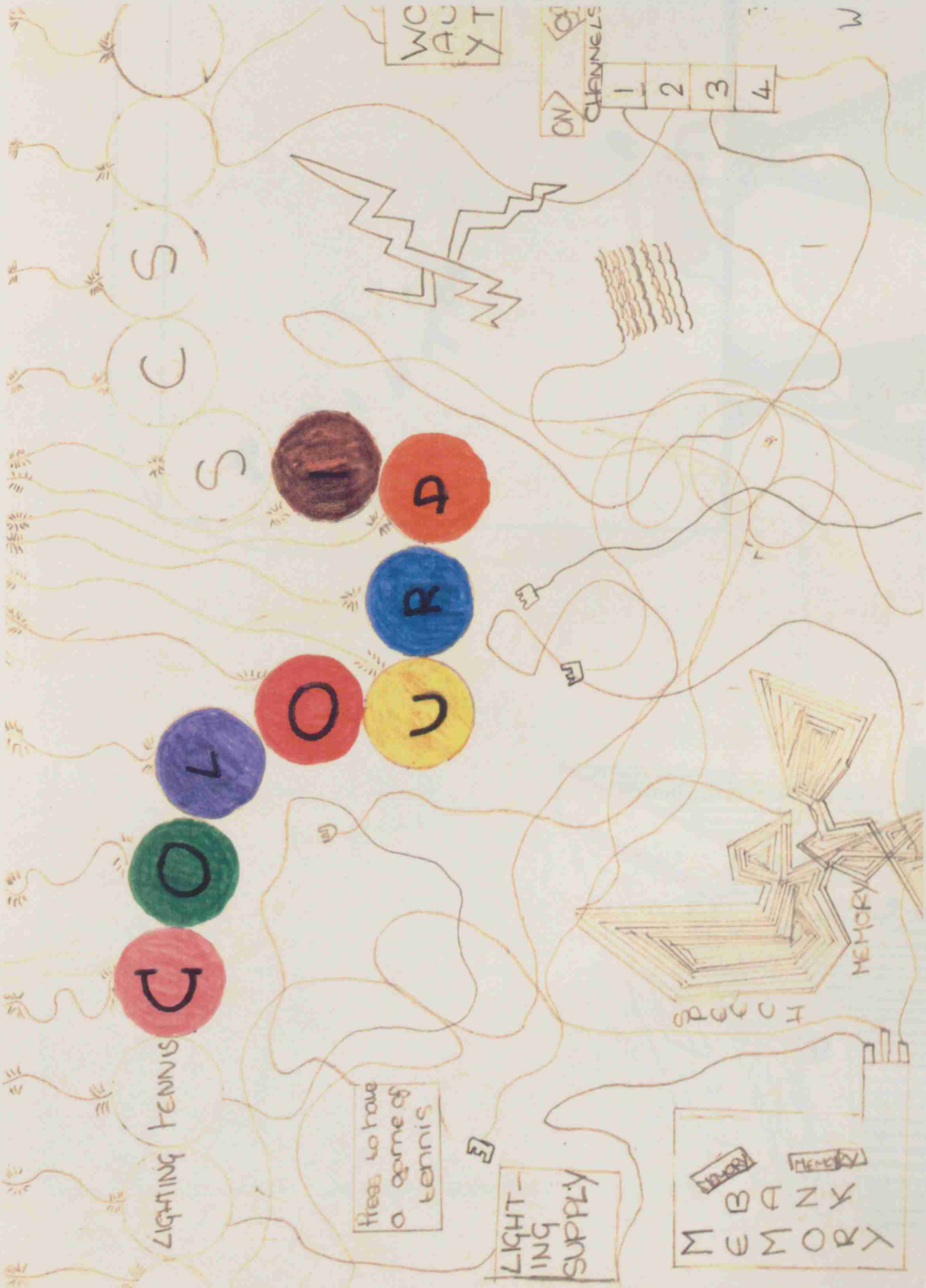
A PRINT
KEY 6em BOARD
THE CHIPPY
ME
WIRE CROSSING
INPUT 2
MEMORY
BANK
DISTRIBUTER
INPUT 7 OUTPUT 1
THE BIG GAME
HOME FOR OLD MICRO-CHIPS
= INPUT □ = REF. • = ball o = OUTPUT



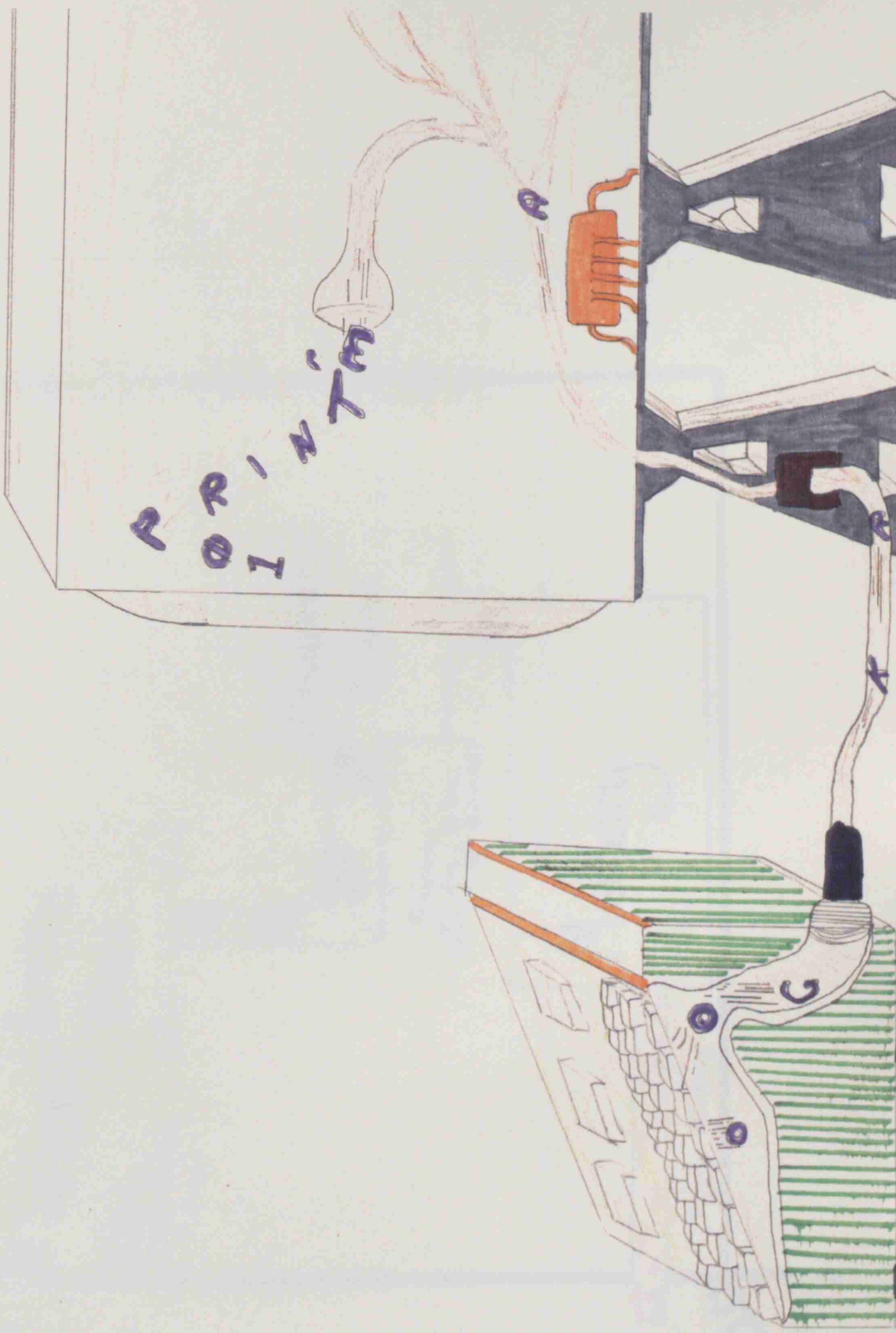
Keyboard St
INPUT
Daily Chip
UP
DOWN
Mini K
K
Ram Ram
Ram
Recharge
me
The Brain
The Match
Memory Bank

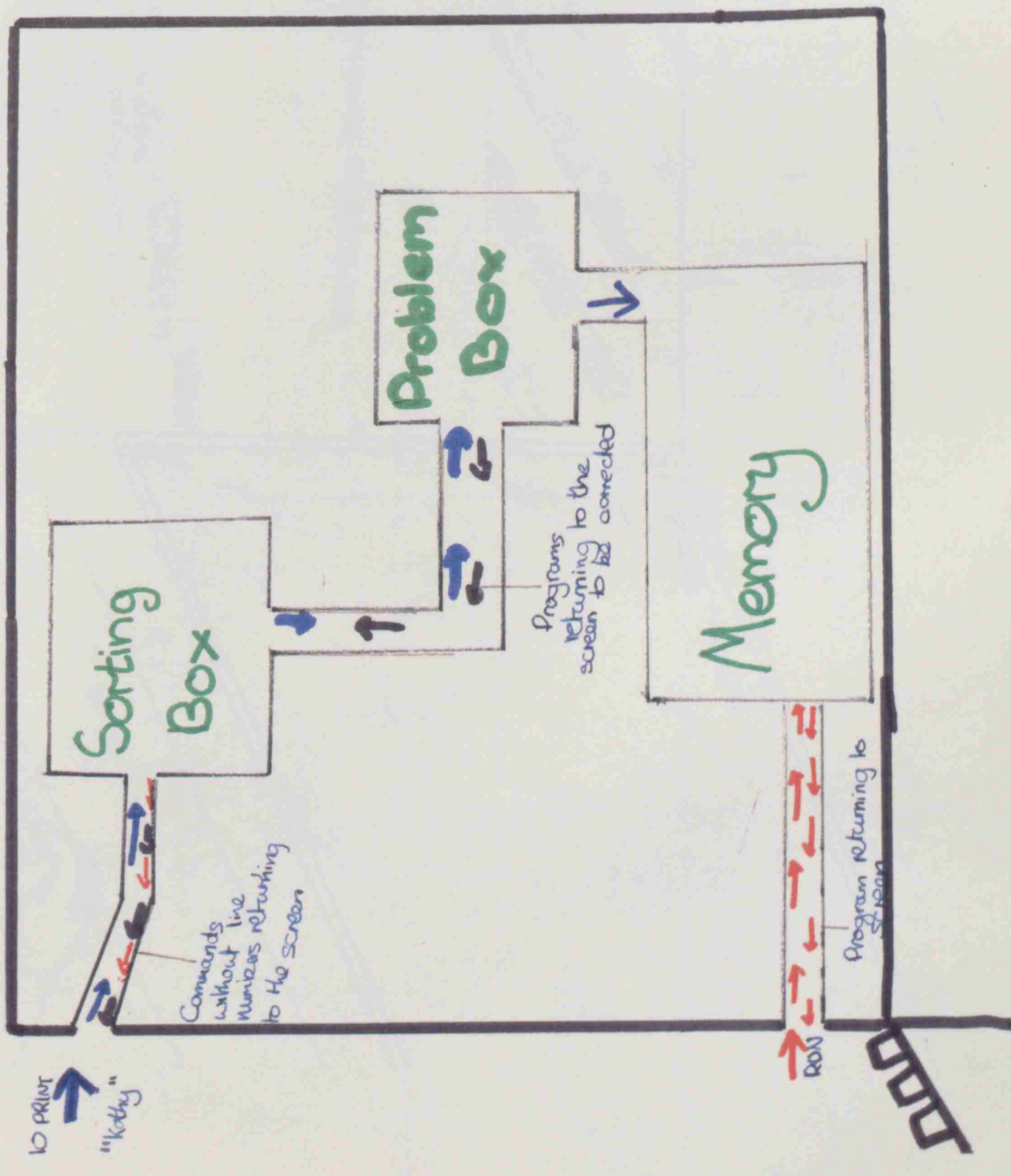

chip
wires
string chips
main chip
chips

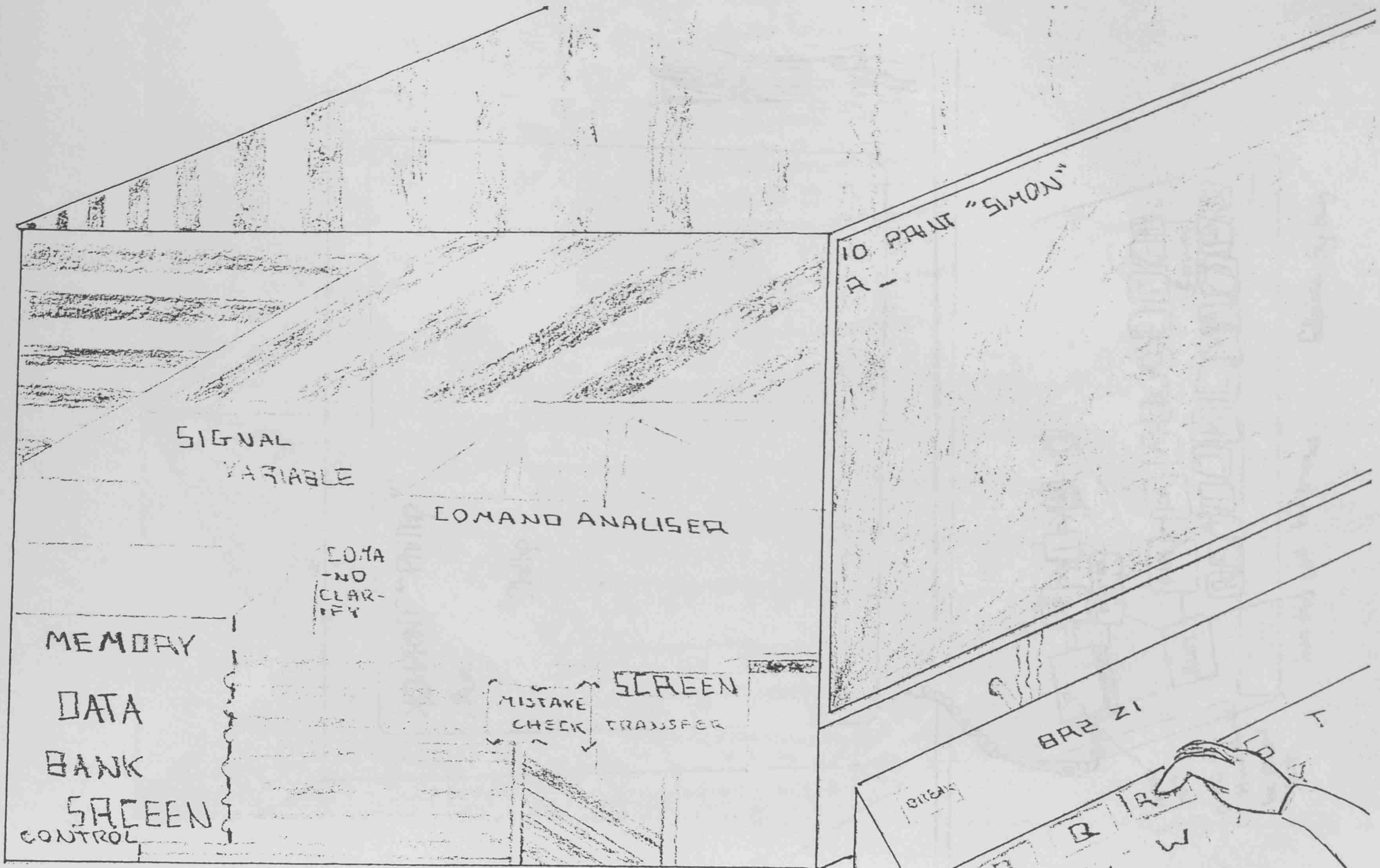

BRAINS + SON
COMP 1
BRAIN
MAIN SUPPLY
BRAINS+SONS
MEMORY COMP
BRAIND+SONS
NICS+C° LMT.
MICRO CHIP (1 of 12)



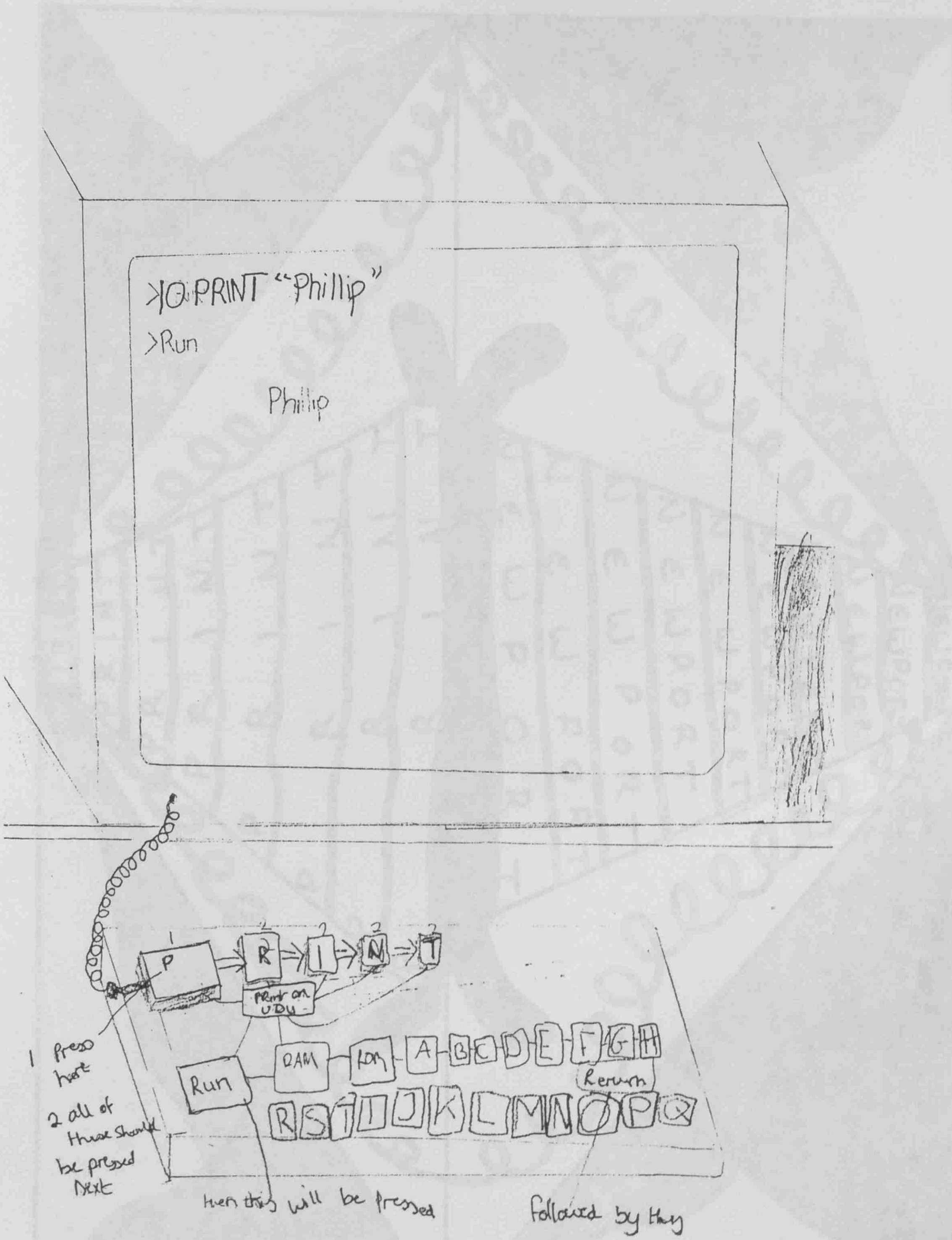
LIGHTING
TENNIS
COLOUR
CHANNELS
ON
1
2
3
4
SPEECH
MEMORY
LIGHT-ING SUPPLY
Press to have a game of tennis
MEMORY

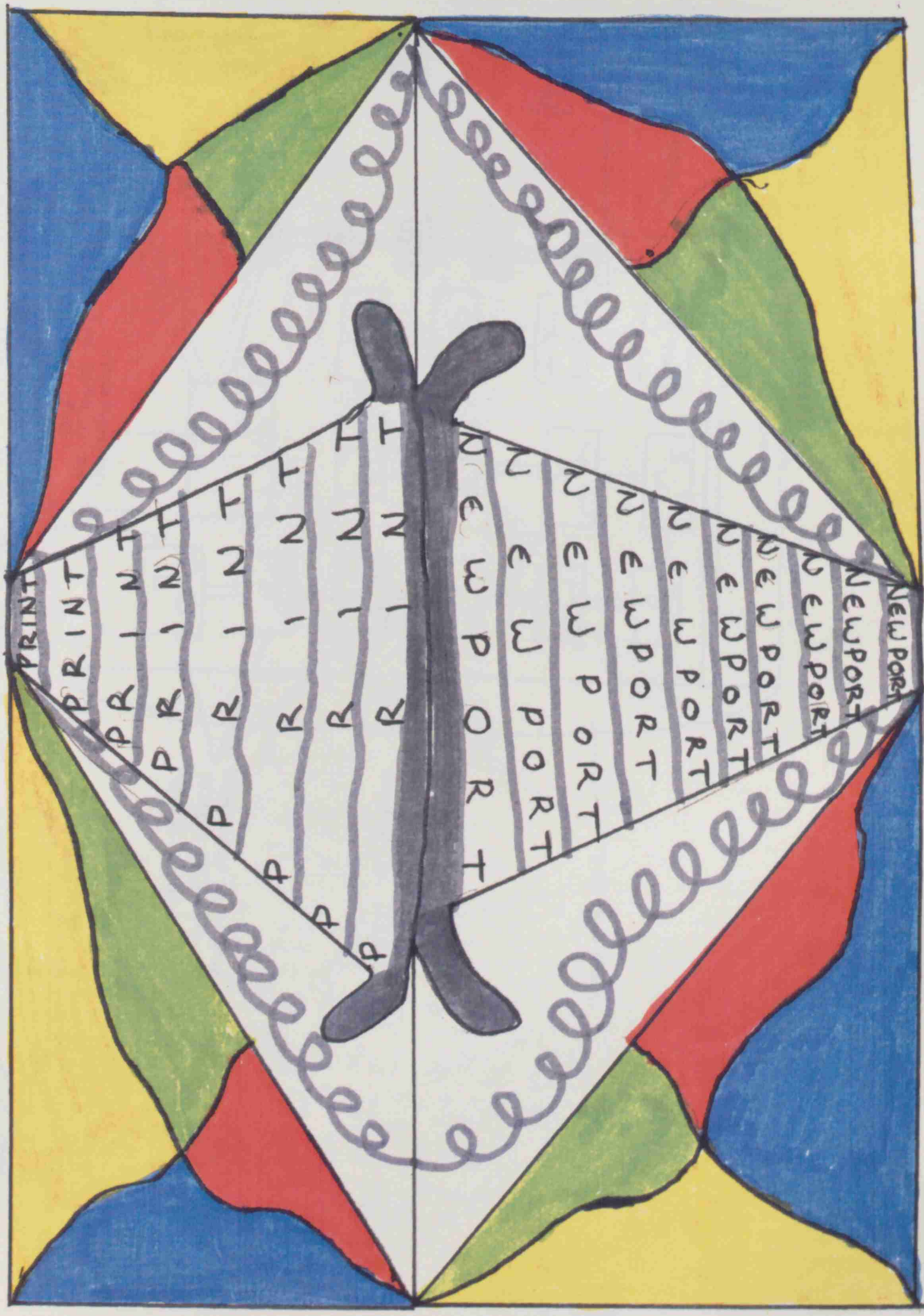


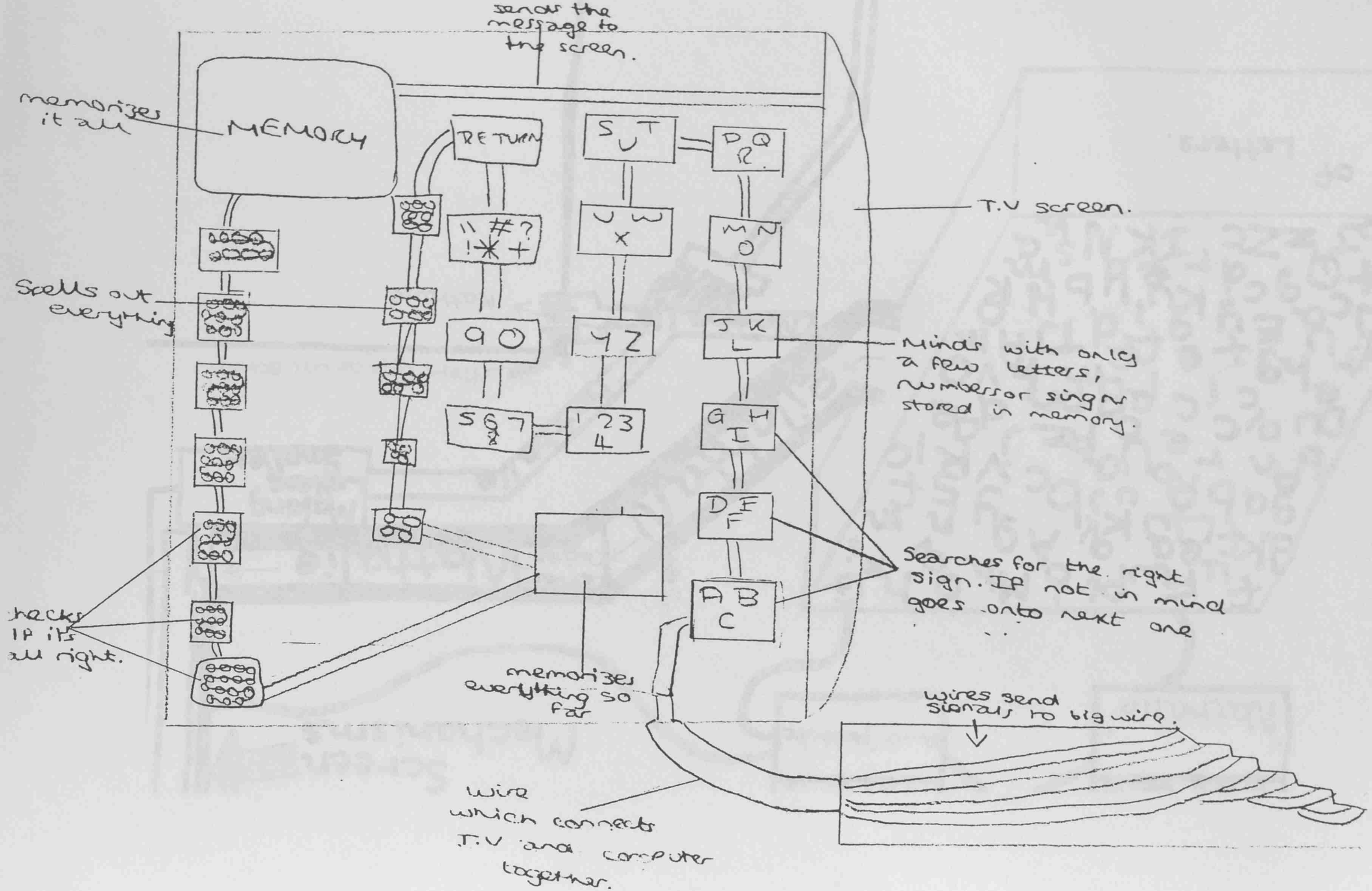

Sorting Box
Problem Box
Memory
Commands without line numbers returning to the screen
Programs returning to the screen to be corrected
Program returning to screen
10 PRINT "Kathy"
RUN

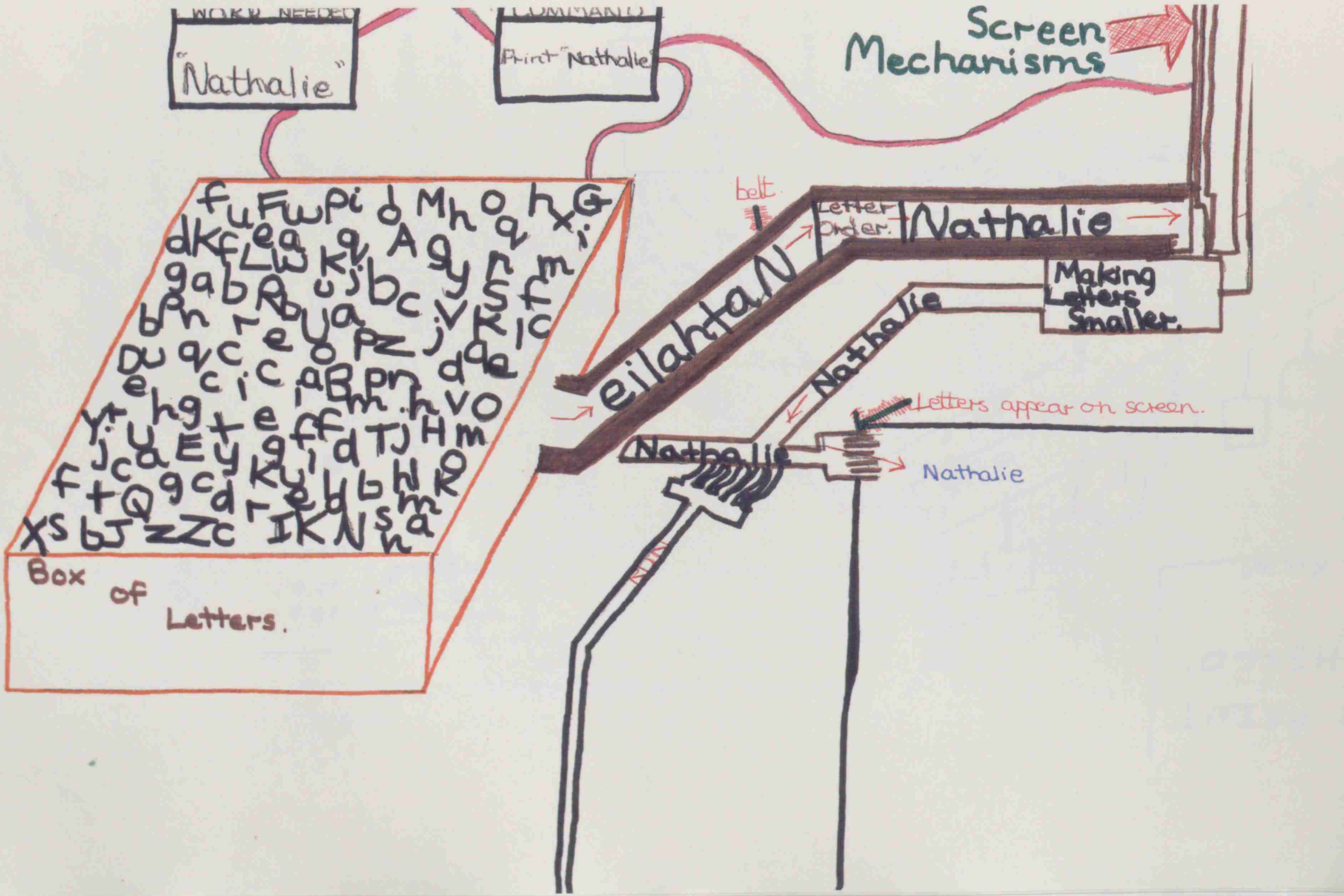


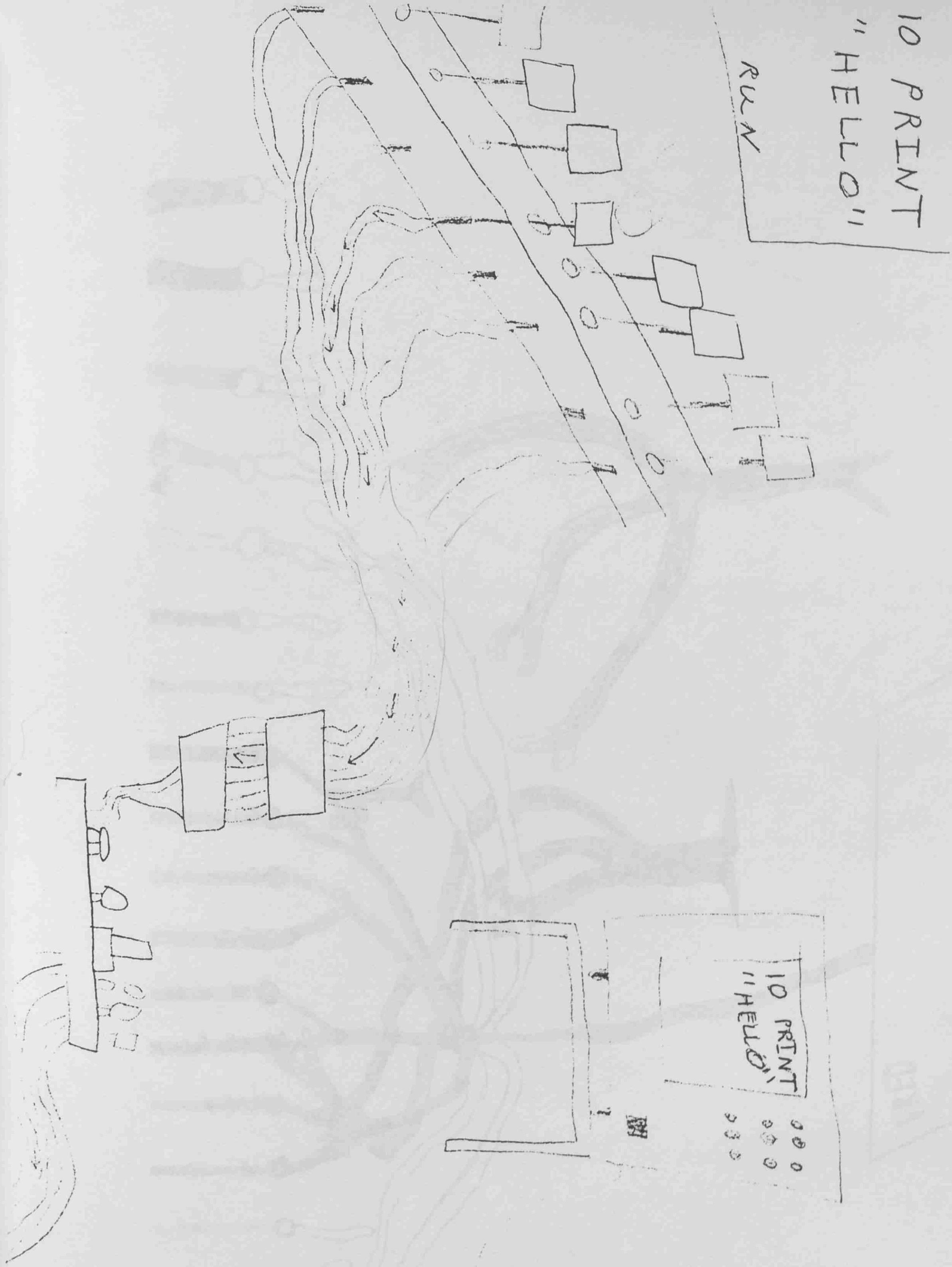
I think that when type in the comand it is checked for mistakes, varified and passed. I then think it is transfered to the screen.

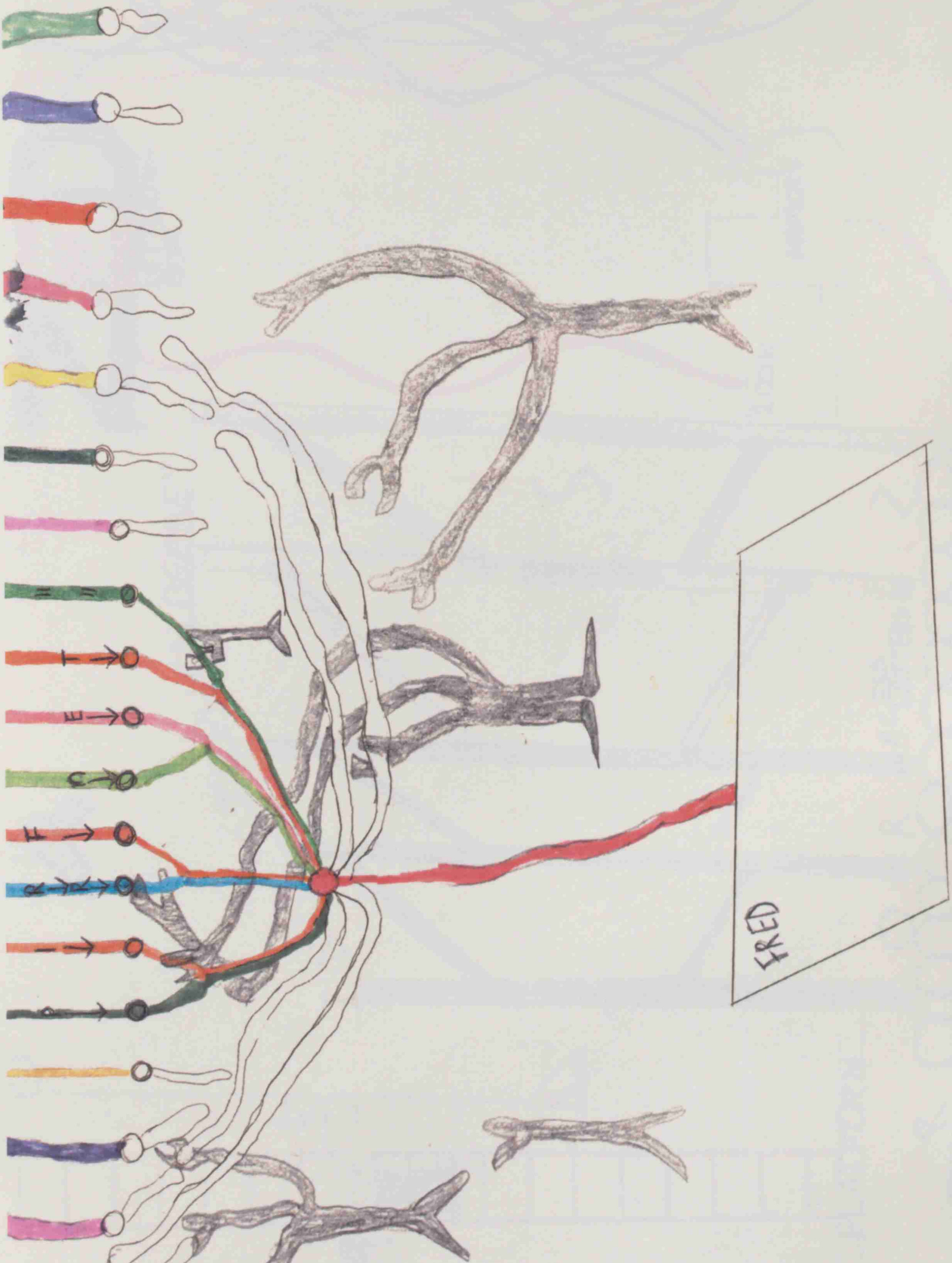

>10 PRINT "Phillip"
>Run
Phillip
1 press here
2 all of these should be pressed Next
Run
Return
then this will be pressed
followed by this

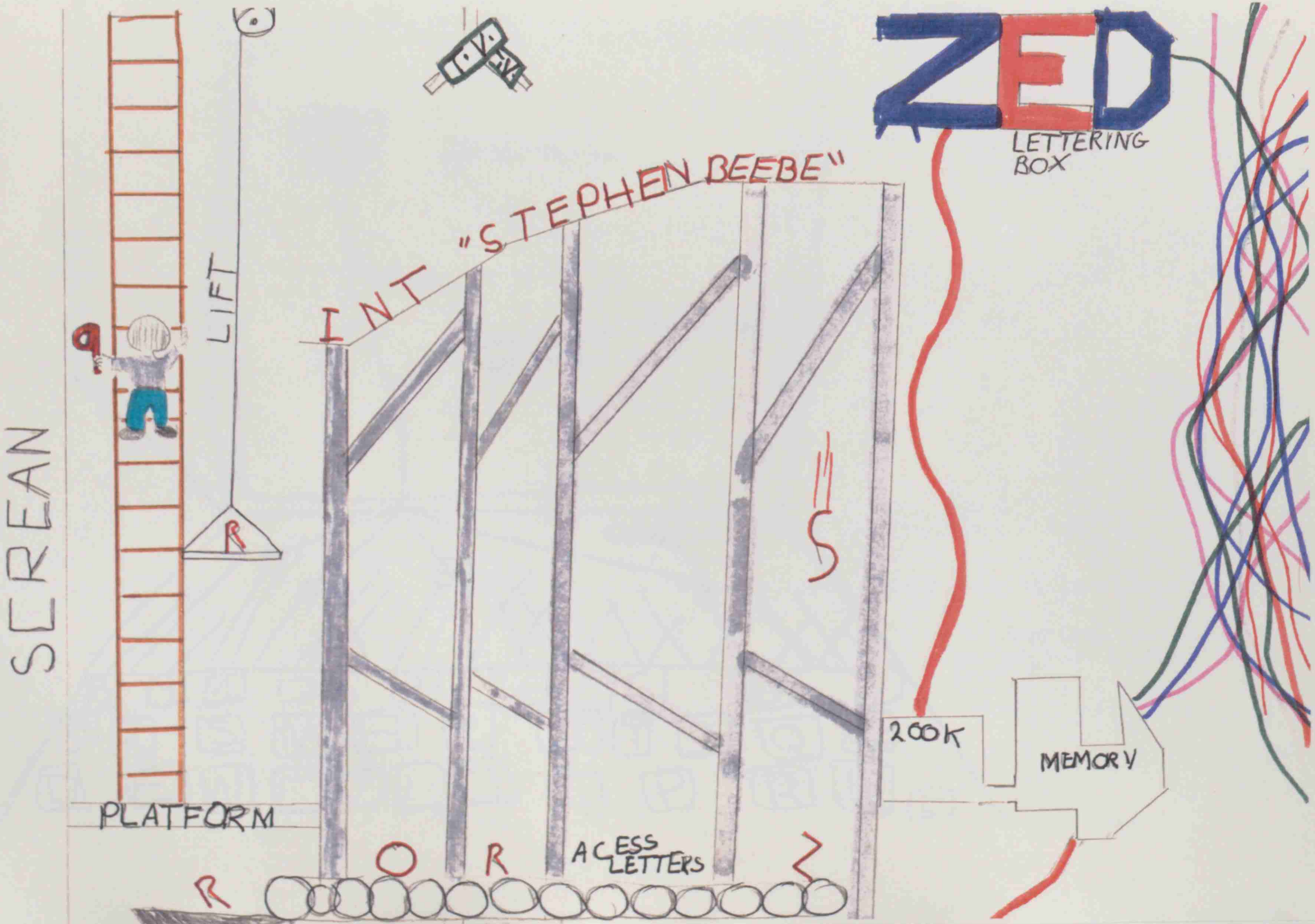

PRINT
PRINT
NEWPORT
NEWPORT

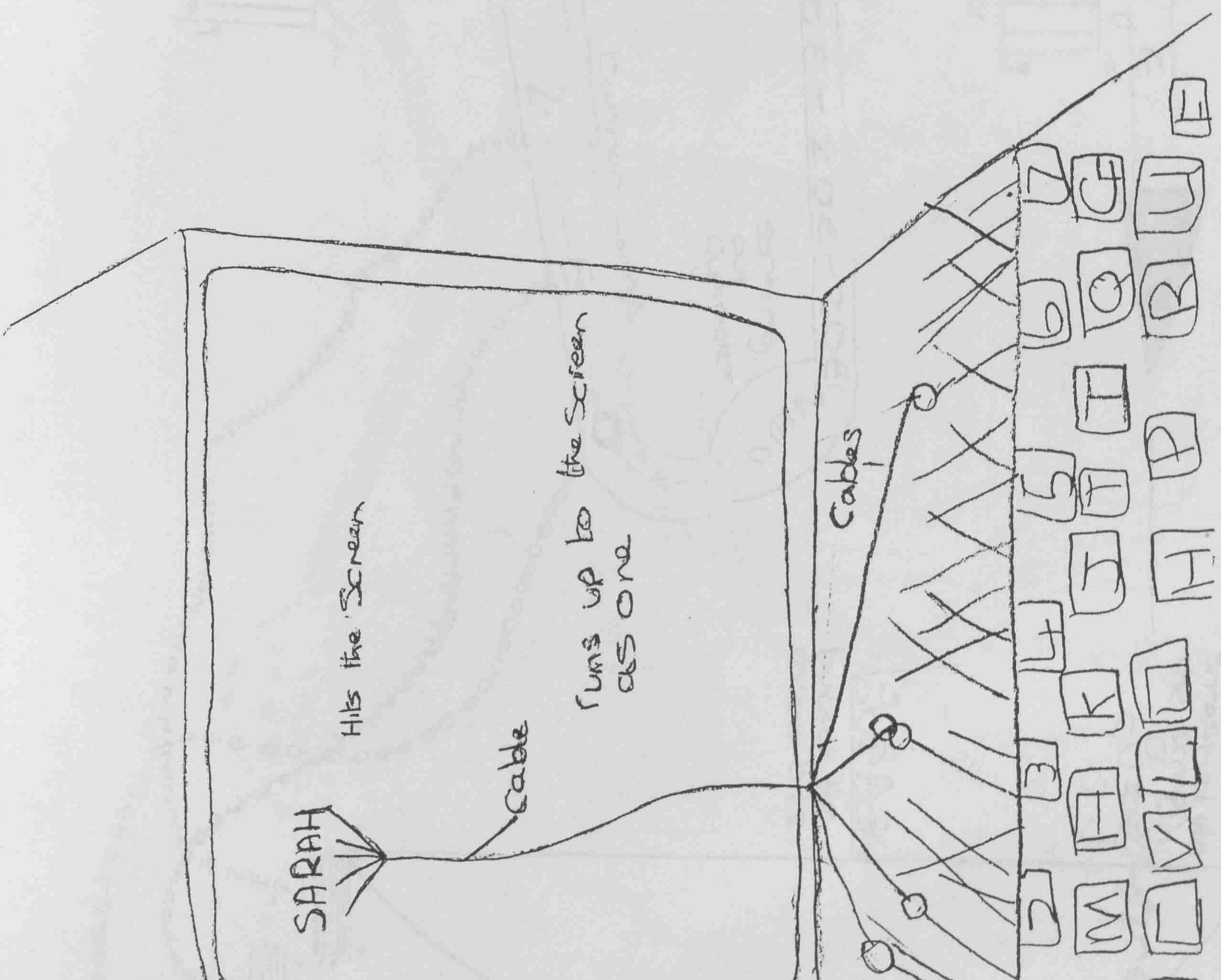

sends the message to the screen.
memorizes it all
MEMORY
RETURN
S T U
P Q R.
T.V screen.
" # ?
! * +
U W X
M N O
Spells out everything
9 0
Y Z
J K L
Minds with only a few letters, numbers or signs stored in memory.
5 6 7 &
' 2 3 4
G H I
D E F
Searches for the right sign. If not in mind goes onto next one
A B C
checks if it's all right.
memorizes everything so far
wires send signals to big wire.
wire which connects T.V and computer together.

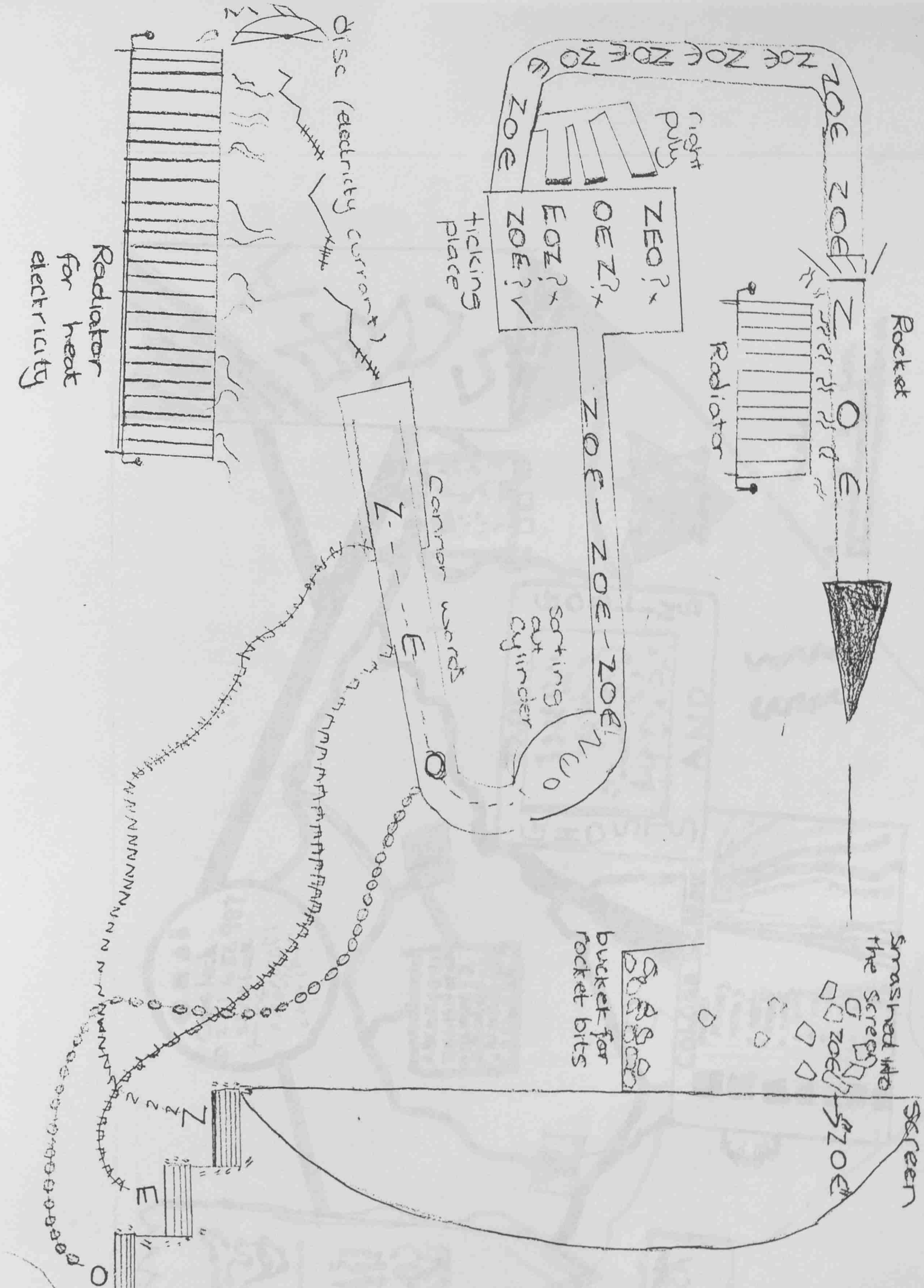

WORD NEEDED
"Nathalie"
COMMAND
Print "Nathalie"
Screen Mechanisms
belt
Letter Order.
Nathalie
eilahtaN
Making Letters Smaller.
Nathalie
Nathalie
Nathalie
Letters appear on screen.
Nathalie
Box of Letters.

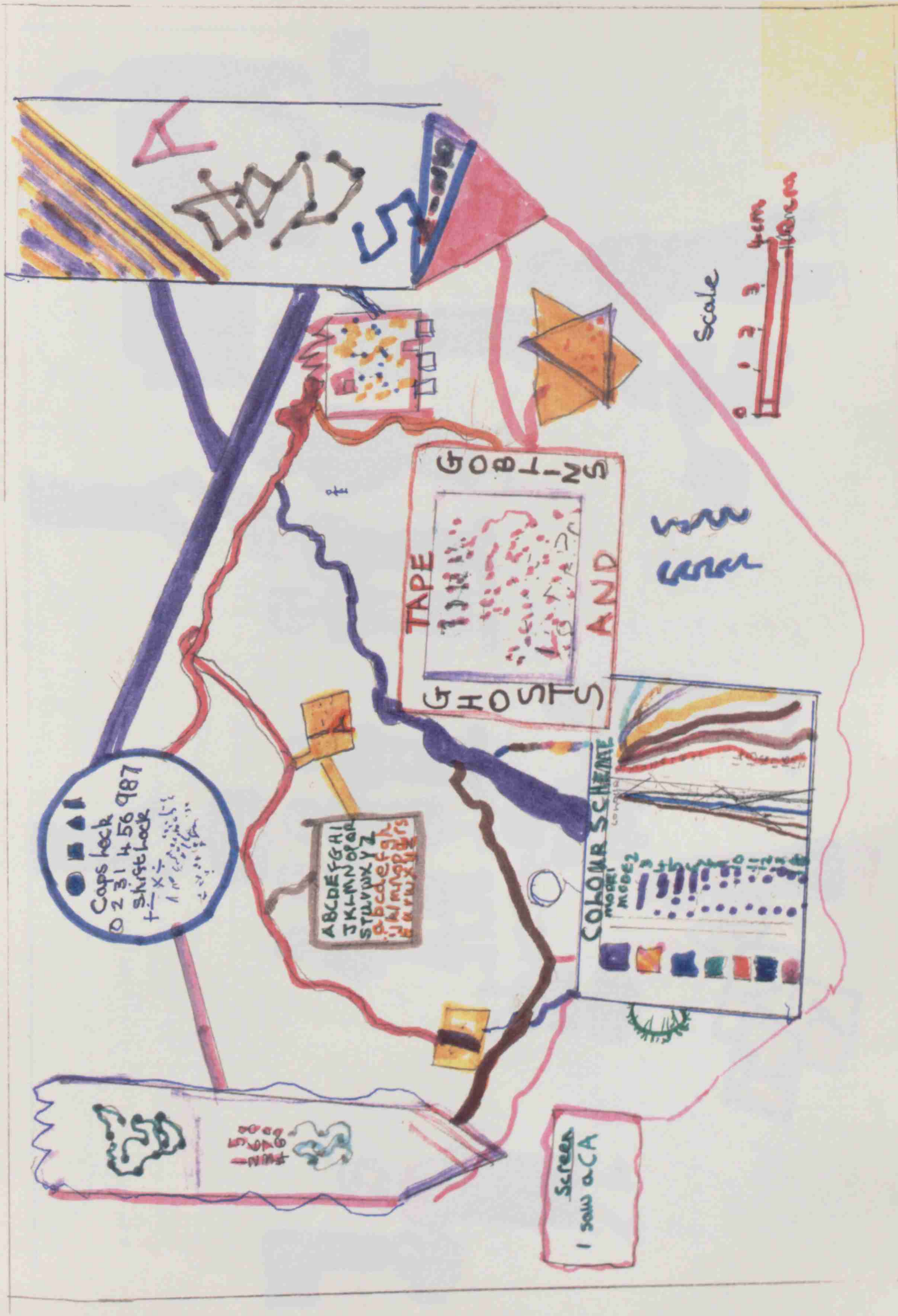

10 PRINT "HELLO"
RUN
10 PRINT "HELLO"


FRED

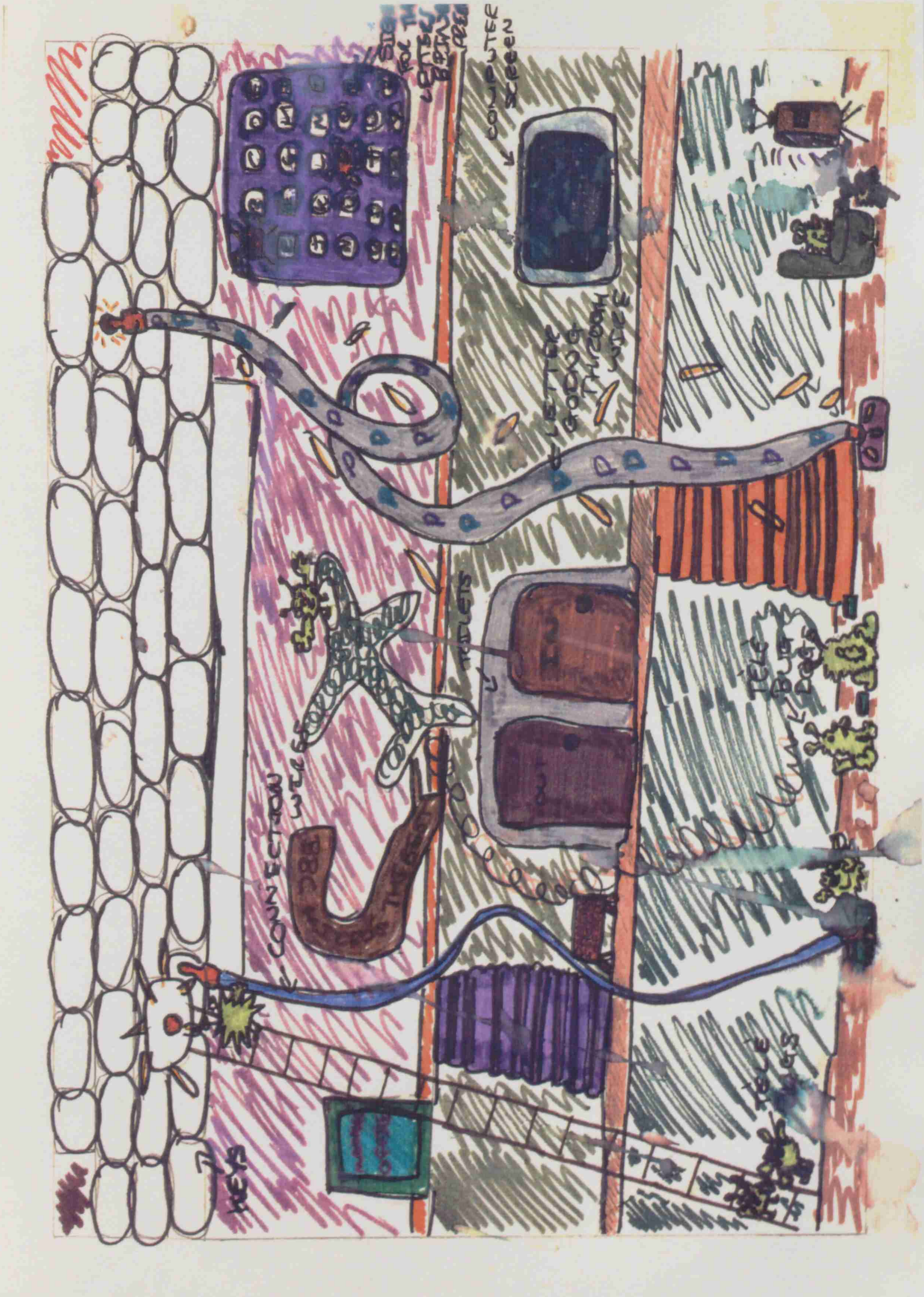

ZED
LETTERING BOX
INT "STEPHEN BEEBE"
LIFT
SCREAN
PLATFORM
ACESS LETTERS
200K
MEMORY

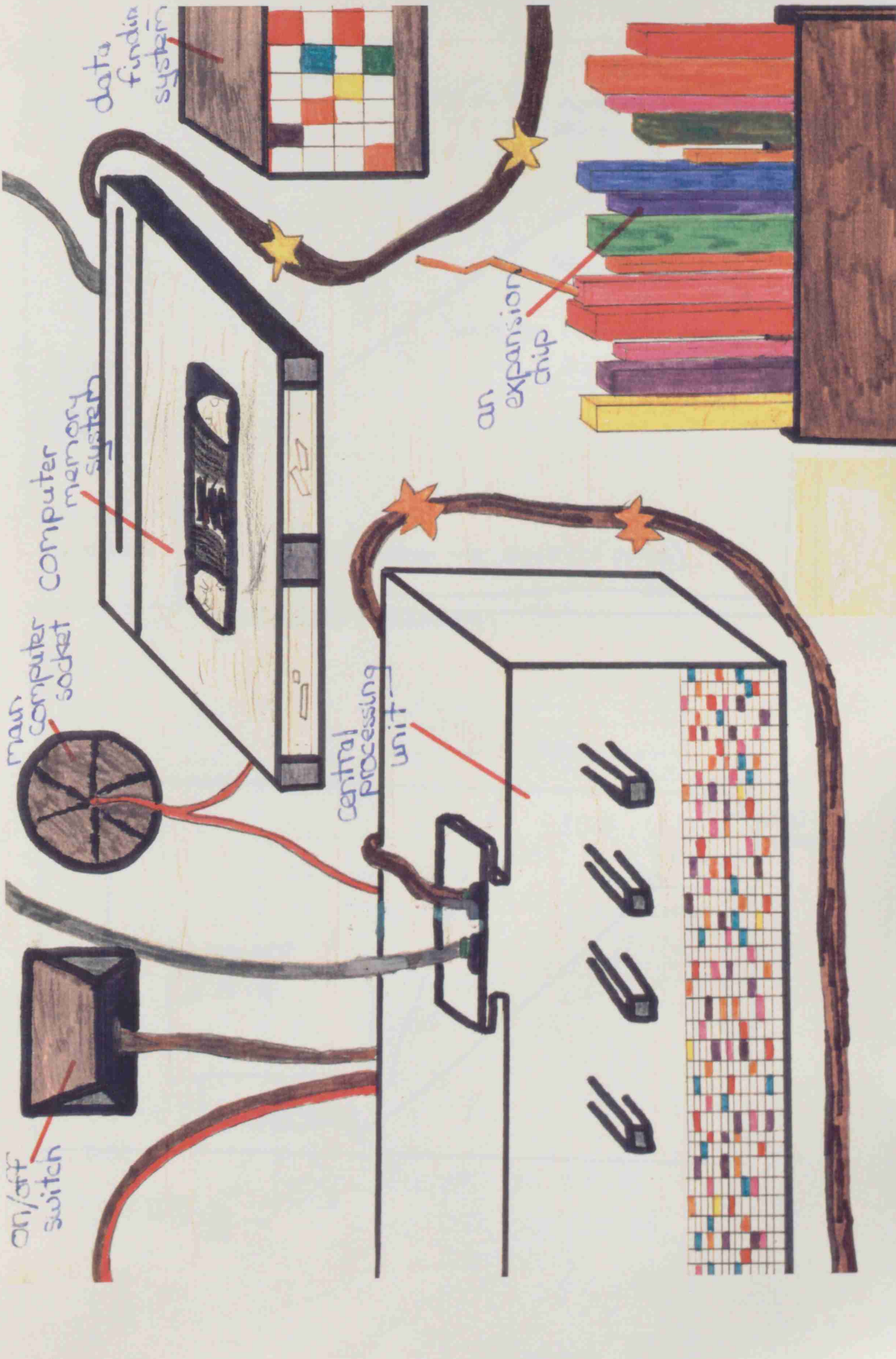

SARAH
Hits the Screen
Cable
runs up to the Screen as one
Cables


Rocket
Radiator
right pully
ZEO?x
OEZ?x
EOZ?x
ZOE?✓
ticking place
ZOE-ZOE-ZOE!
sorting out cylinder
common words
disc (electricity current)
Radiator for heat electricity
smashed into the screen
bucket for rocket bits
Screen


TAPE
GOBLINS
GHOSTS
AND
COLOUR SCHEME
MODE 1
MODE 2
Scale
Screen
I saw a CA
Caps lock
0 2 3 1 4 5 6 9 8 7
shift lock
ABCDEFGHI
JKLMNOPQR
STUVWXYZ
abcdefghi
jklmnopqrs


FRED


COMPUTER SCREEN
LETTER GOING THROUGH WIRE
TOILETS
CONNECTION WIRES
KEYS


on/off switch
main computer socket
computer memory system
central processing unit
data findin system
an expansion chip

S2B

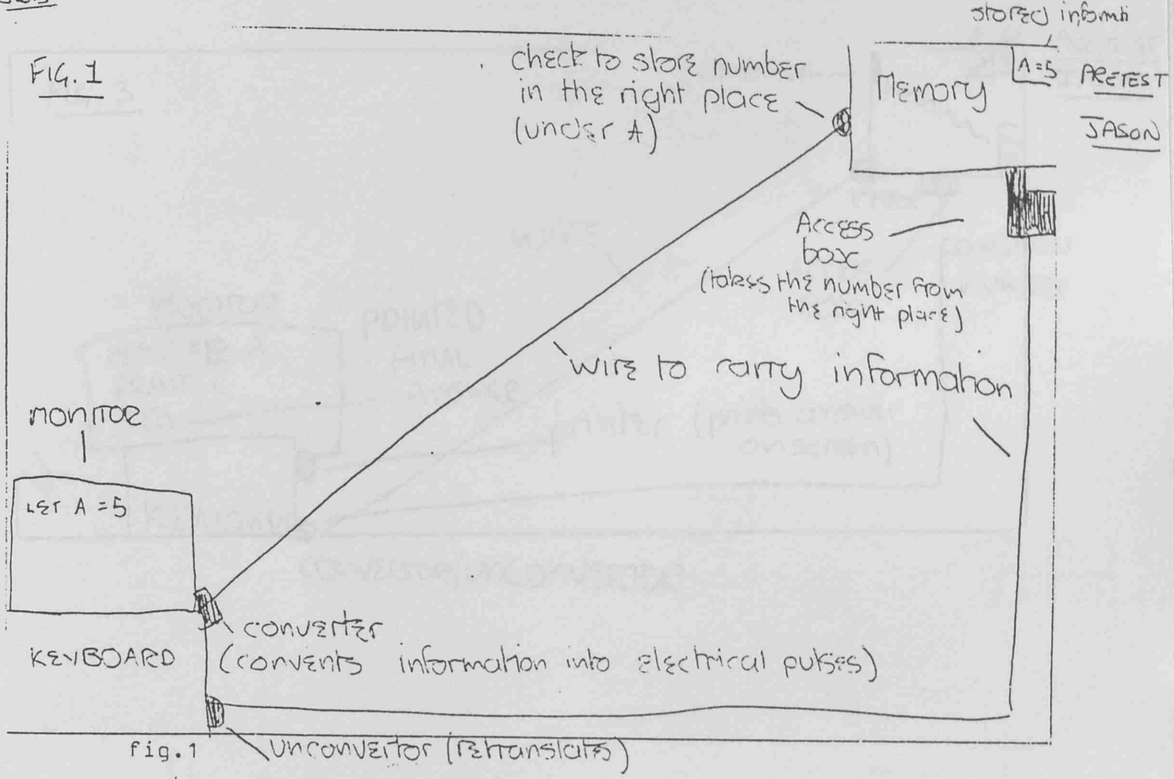


Fig. 1
stored informati
A=5
PRETEST
JASON
check to store number in the right place (under A)
Memory
Access box (takes the number from the right place)
wire to carry information
monitor
LET A = 5
KEYBOARD
converter (converts information into electrical pulses)
Unconvertor (retranslates)

fig.1

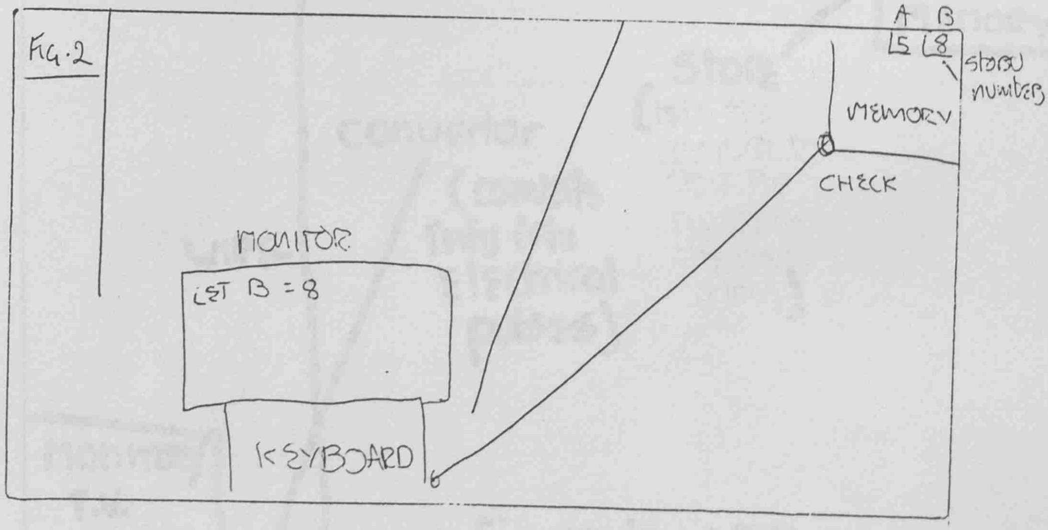
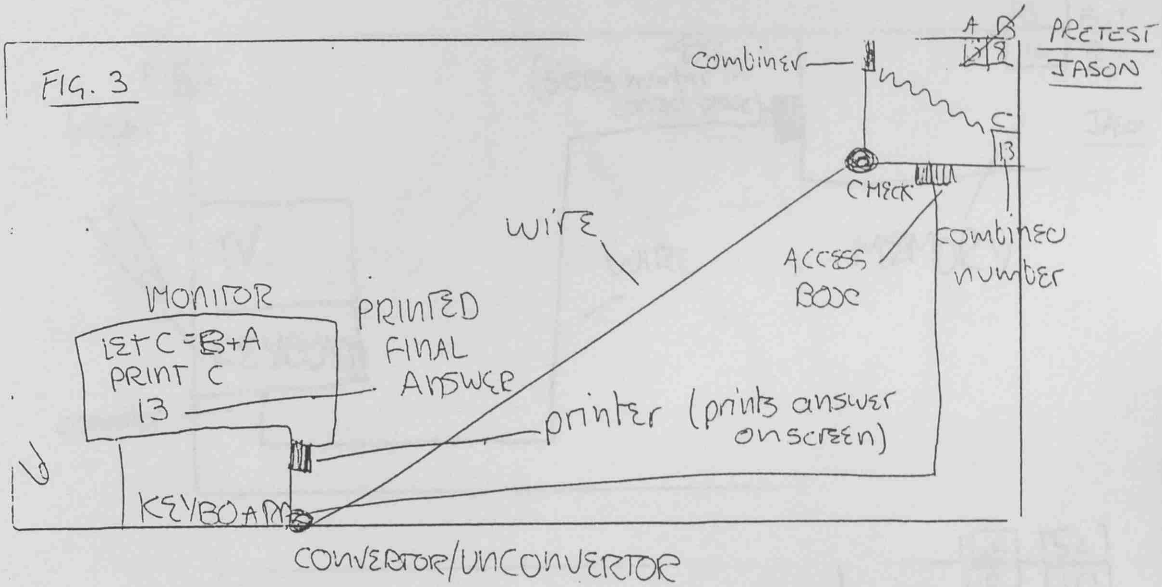


Fig. 2
A B
5 8
stored numbers
MEMORY
CHECK
monitor
LET B = 8
KEYBOARD


PRETEST
JASON
FIG. 3
combiner
A
B
C
13
CHECK
WIRE
ACCESS BOX
combined number
MONITOR
LET C = B+A
PRINT C
13
PRINTED FINAL ANSWER
printer (prints answer onscreen)
KEYBOARD
CONVERTOR/UNCONVERTOR

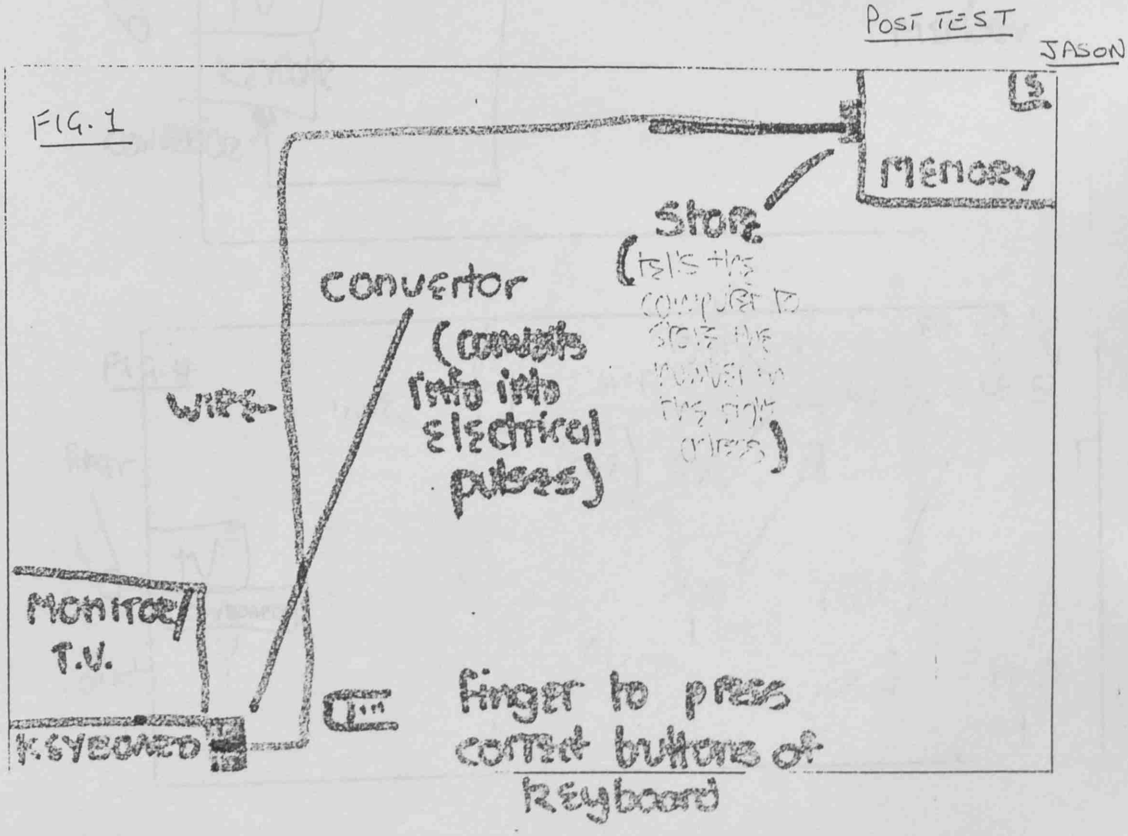

POST TEST
JASON
FIG. 1
Memory
Store
convertor
(converts info into electrical pulses)
wire
Monitor/ T.V.
KEYBOARD
finger to press correct buttons of Keyboard

FIG 2

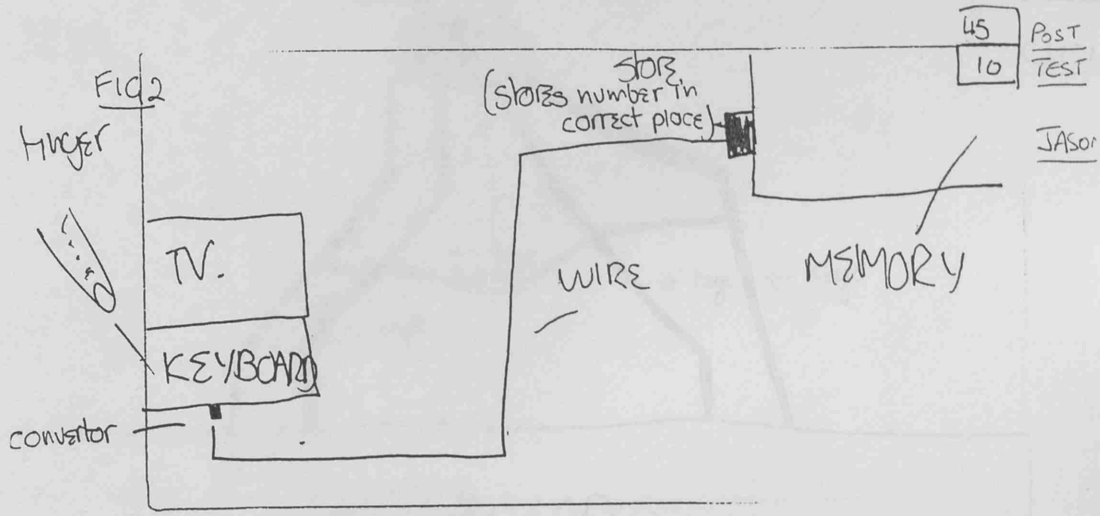


FIG. 3

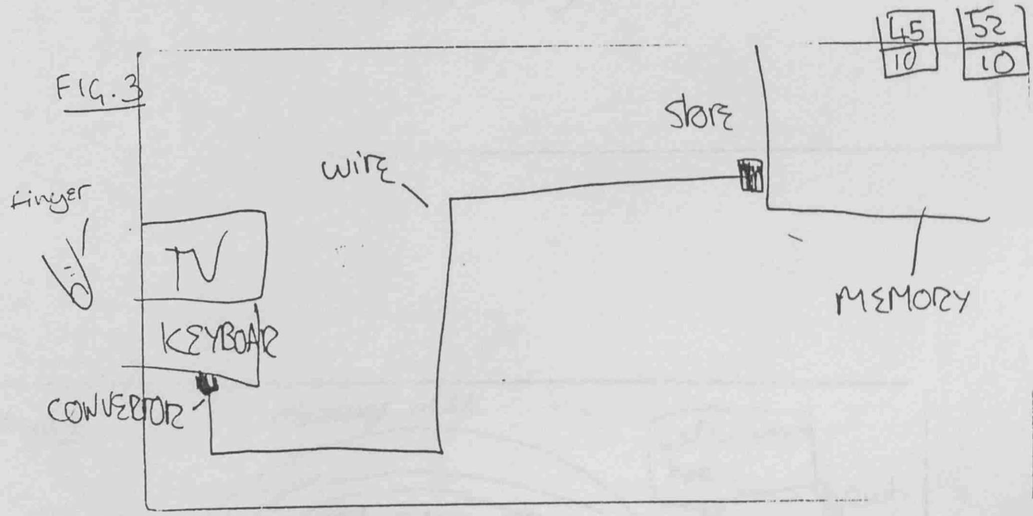
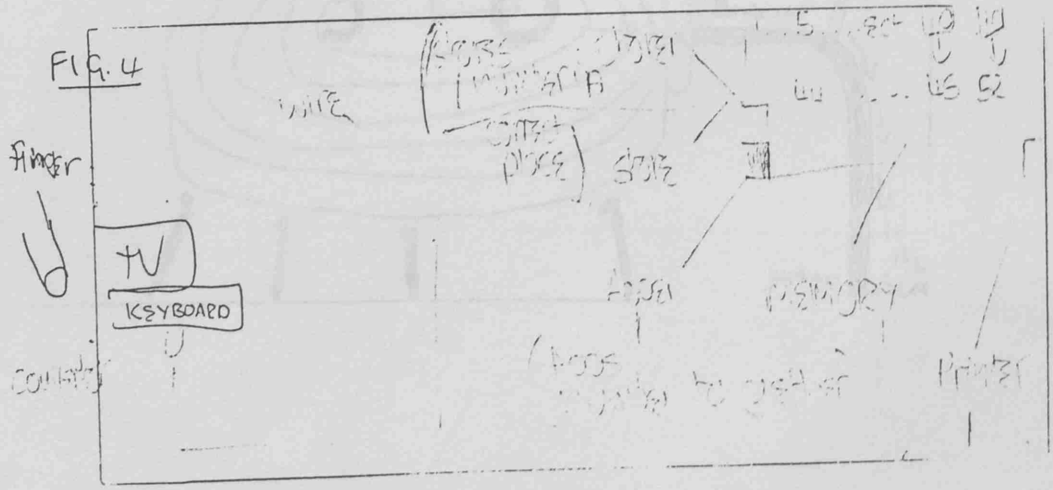


FIG. 4



S2A

PRE TES

CHAN

FIG. 1

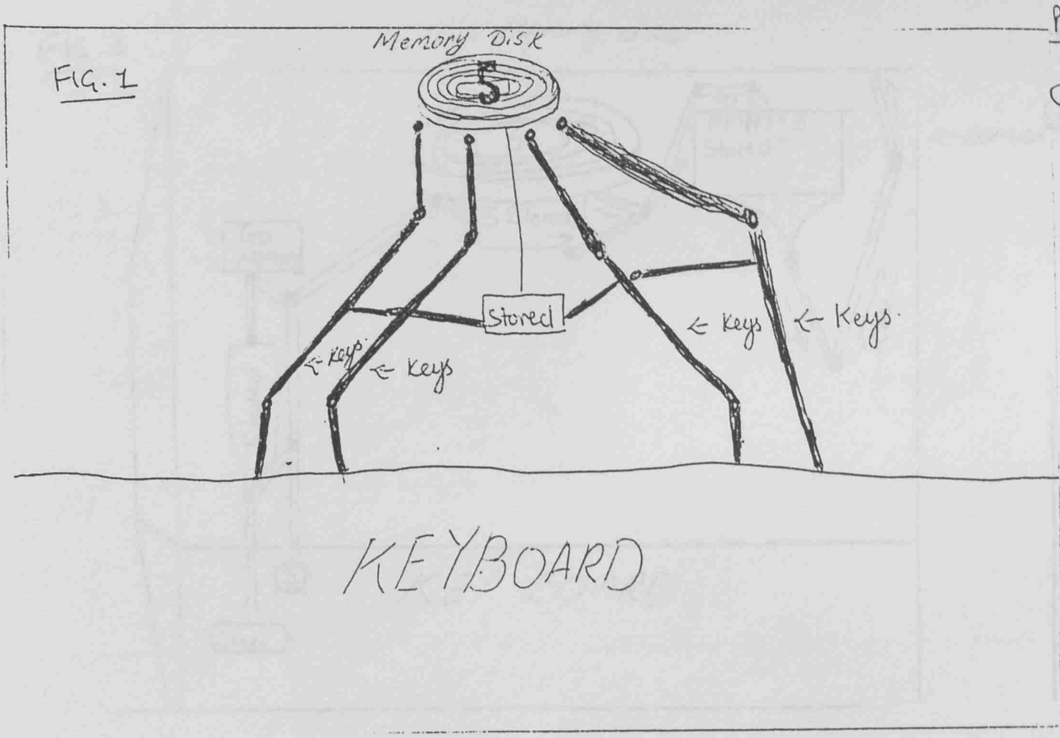


FIG. 2

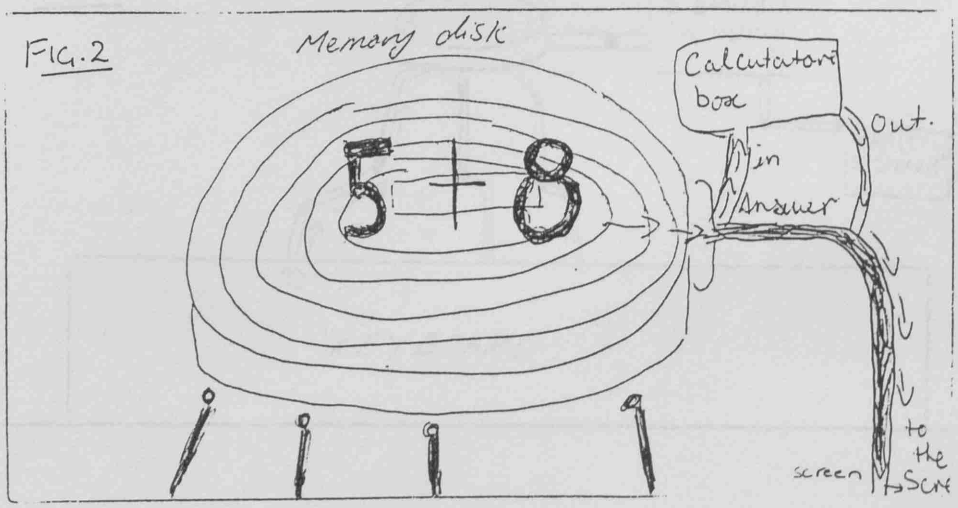
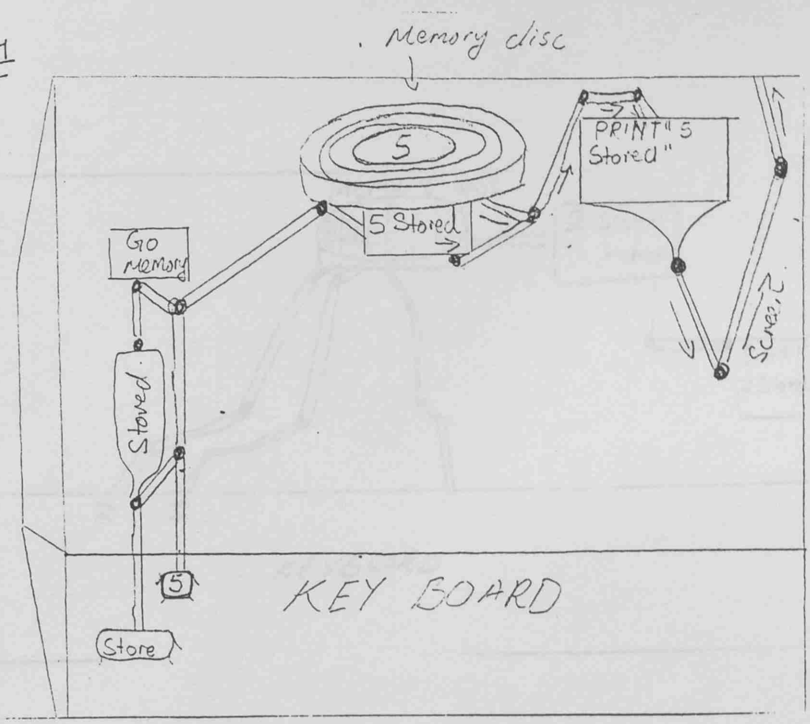
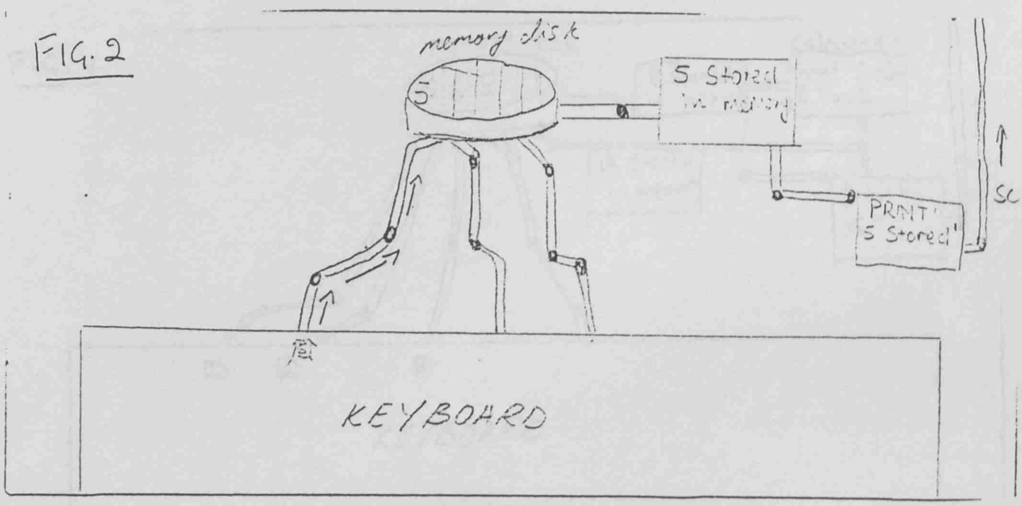


FIG. 1



POST TEST
CHAN

FIG. 2



POST TEST

CHAN

FIG. 3

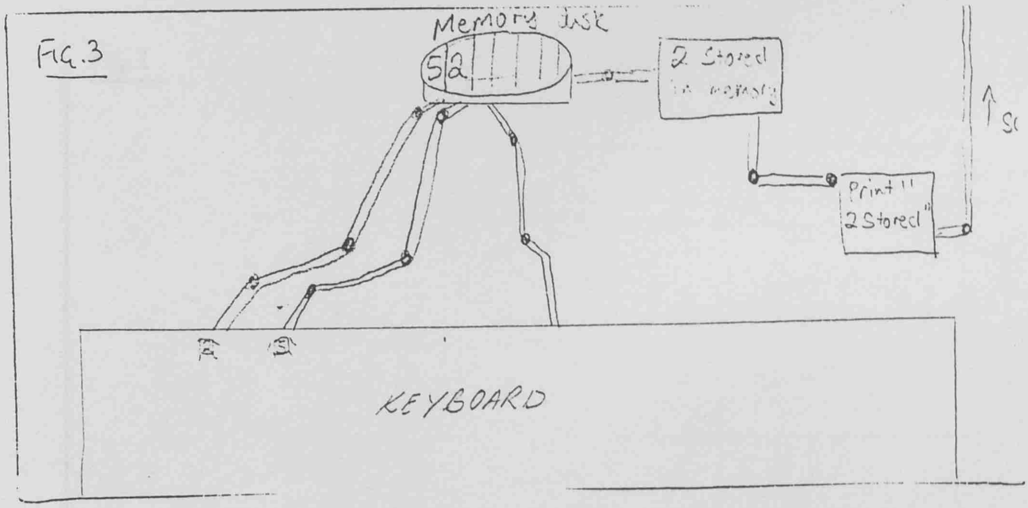
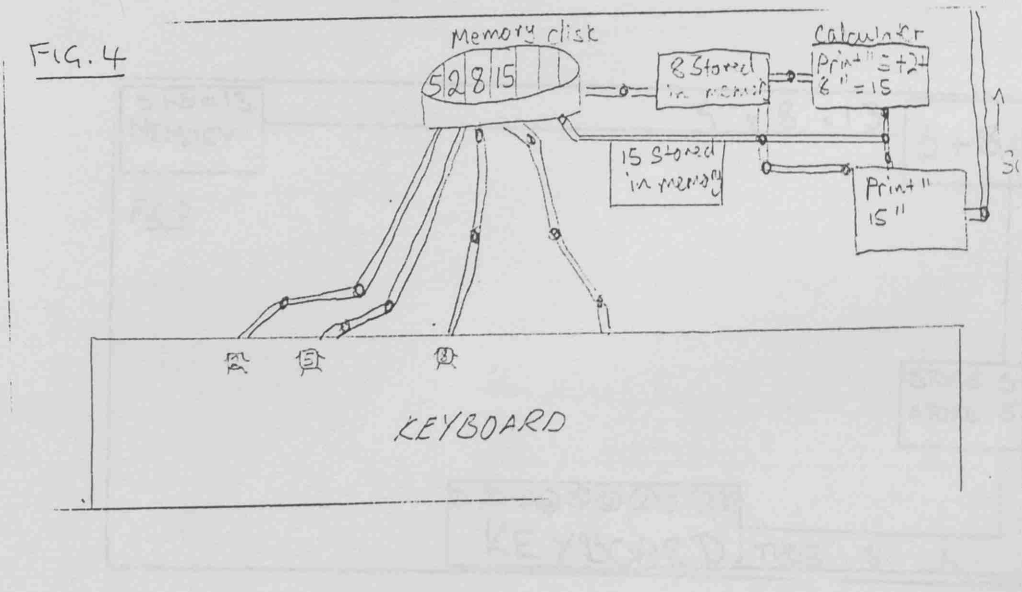
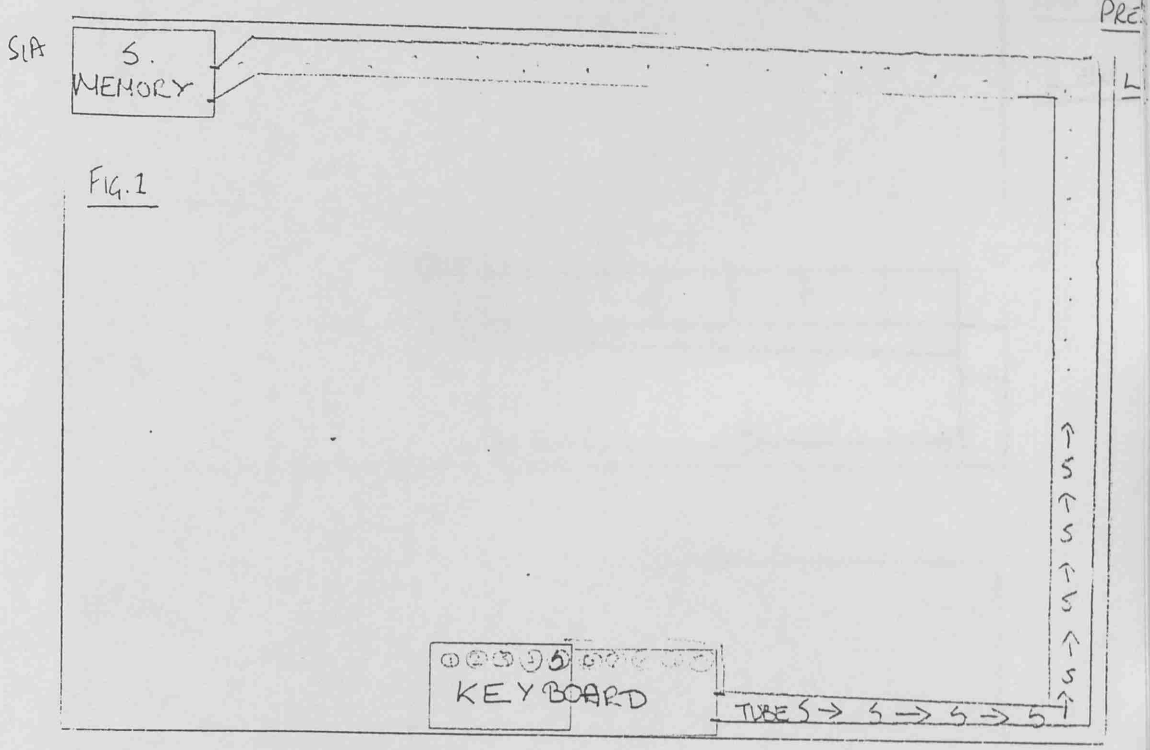


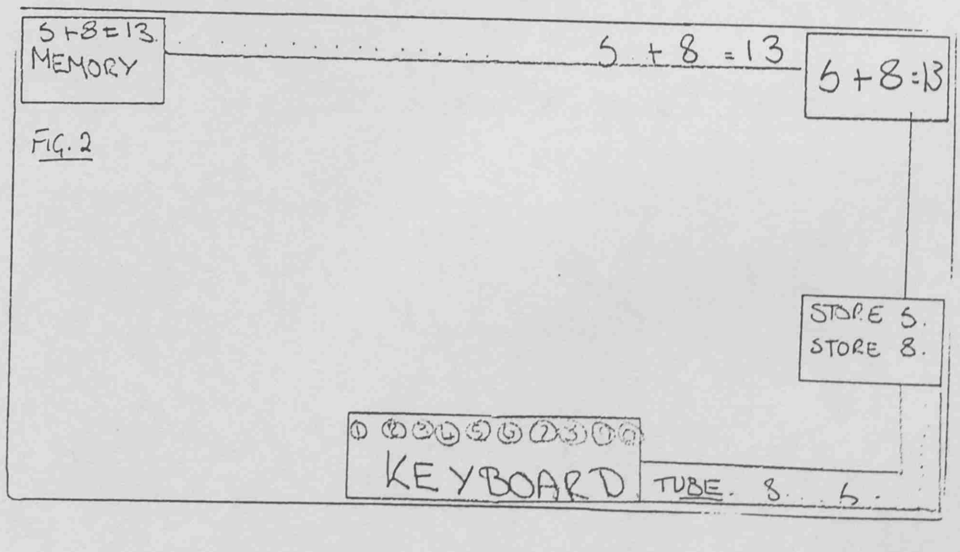
FIG. 4





S(A
5.
MEMORY
KEYBOARD
TUBE 5 → 5 → 5 → 5
5 ↑ 5 ↑ 5 ↑ 5 ↑ 5 ↑
PRE

Fig. 1



5+8=13.
MEMORY
5 + 8 = 13
5+8=13
STORE 5.
STORE 8.
KEYBOARD
TUBE. 8. 5.

Fig. 2

POST TEST

LEE

FIG. 1

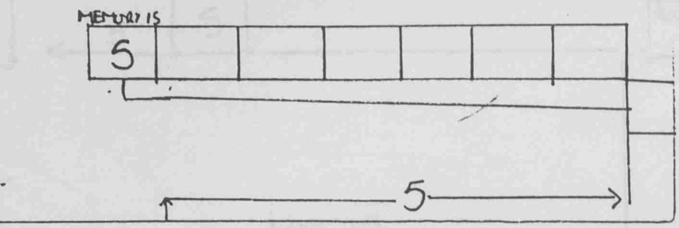


FIG. 2

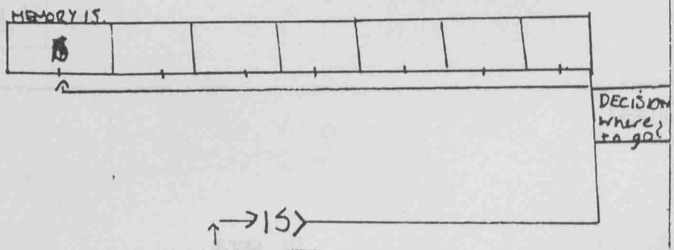
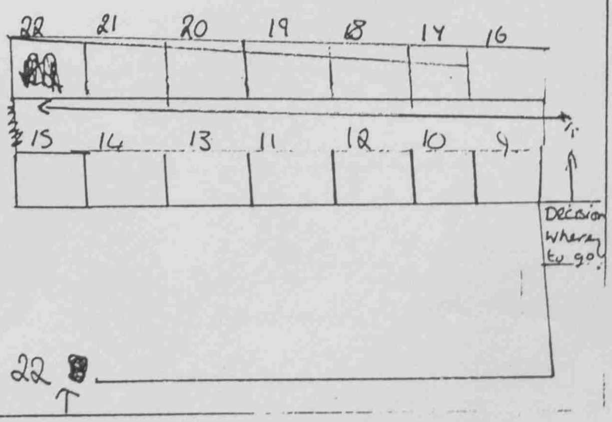


FIG. 3



POST T

LE

FIG. 6

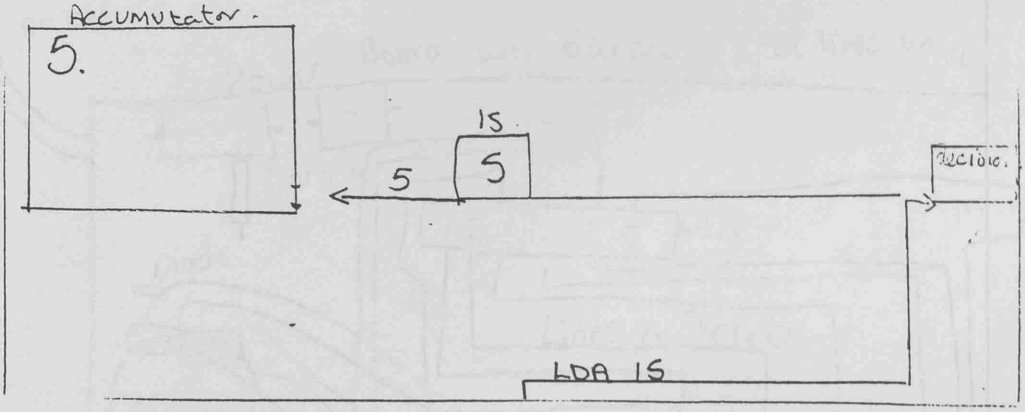


FIG. 8

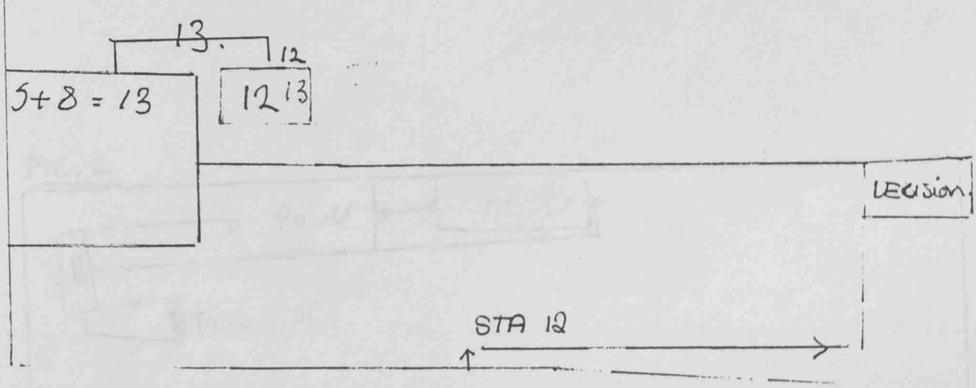


FIG. 1

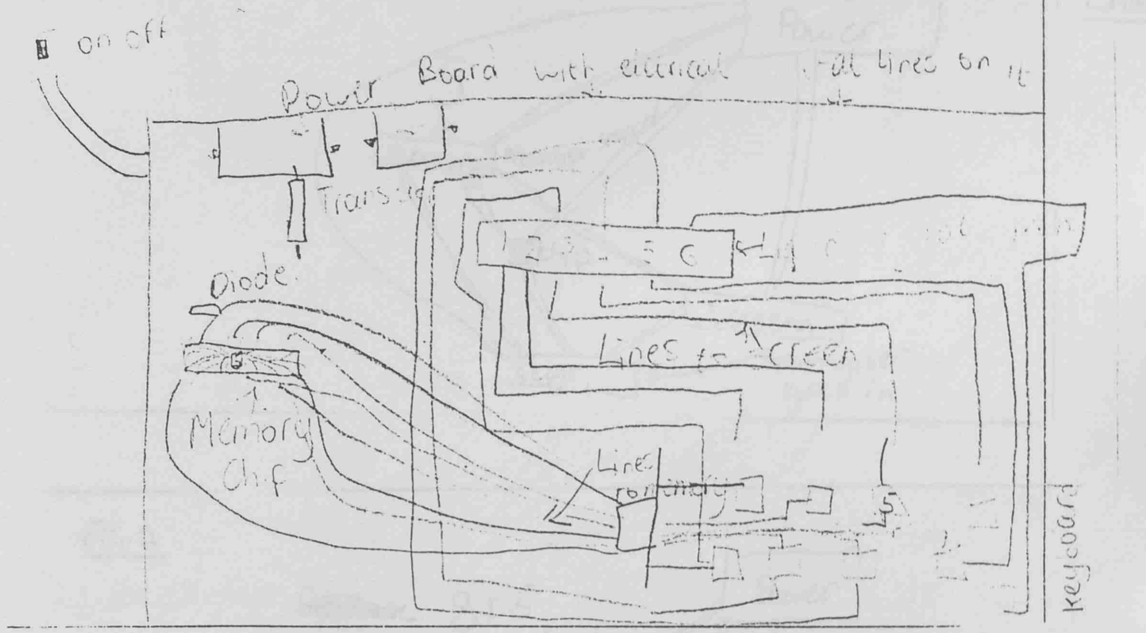

on off
Power Board with
Diode
Memory Chip
Lines to screen
Keyboard

FIG. 2

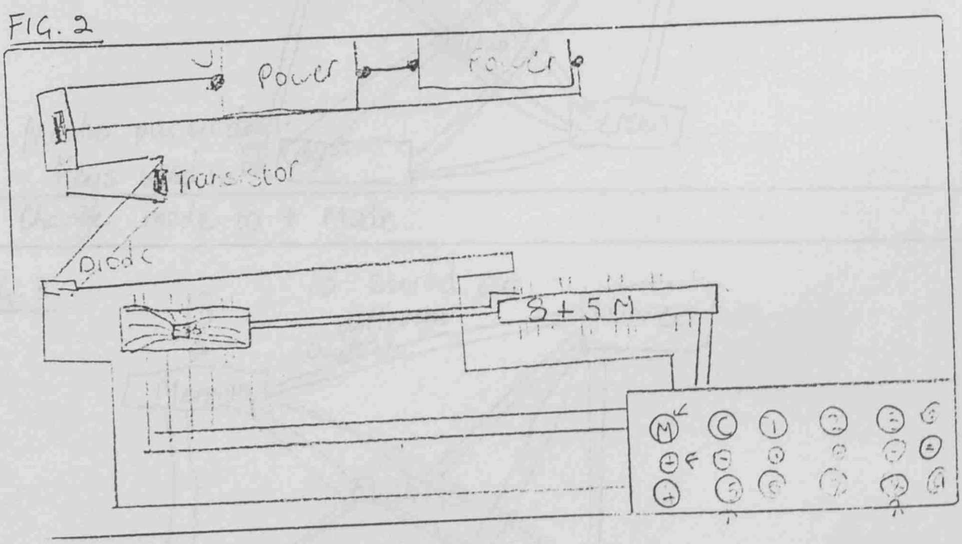

Power
Transistor
Diode
8 + 5 M

FIG. 1

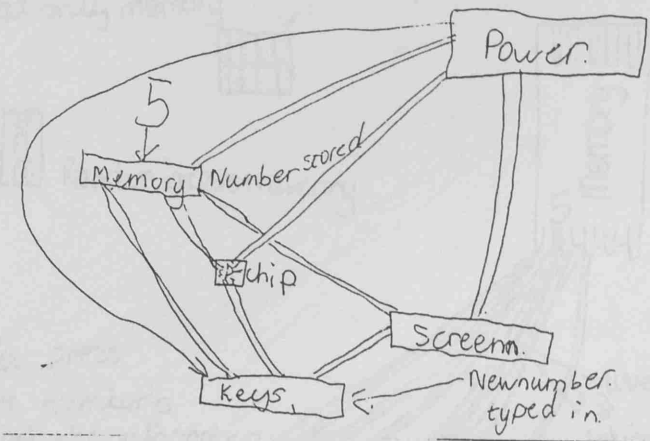
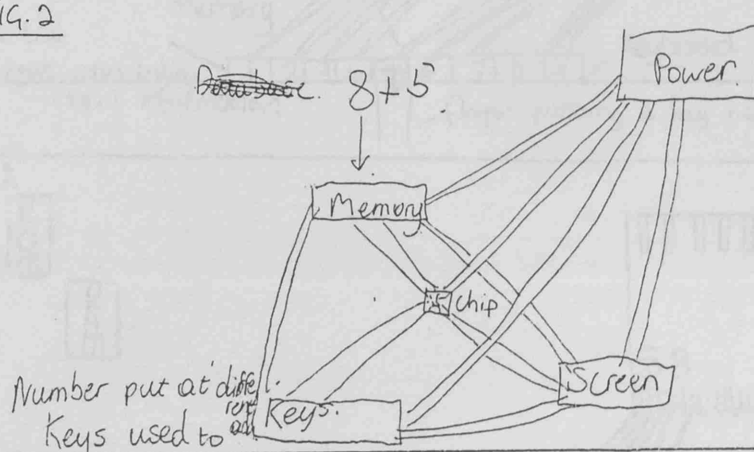
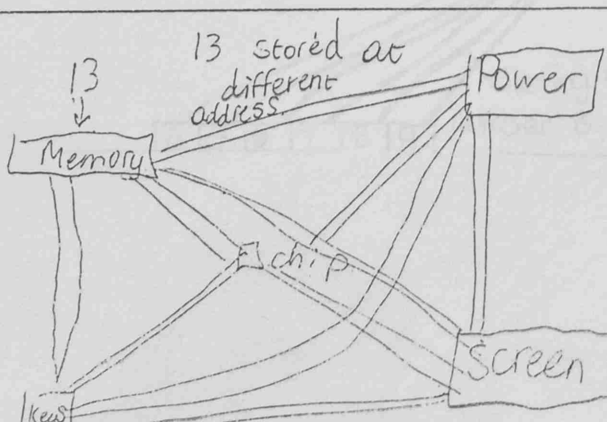


FIG. 2



Change mode to + mode.

FIG. 3



S1A

James

PRETES

FIG. 1

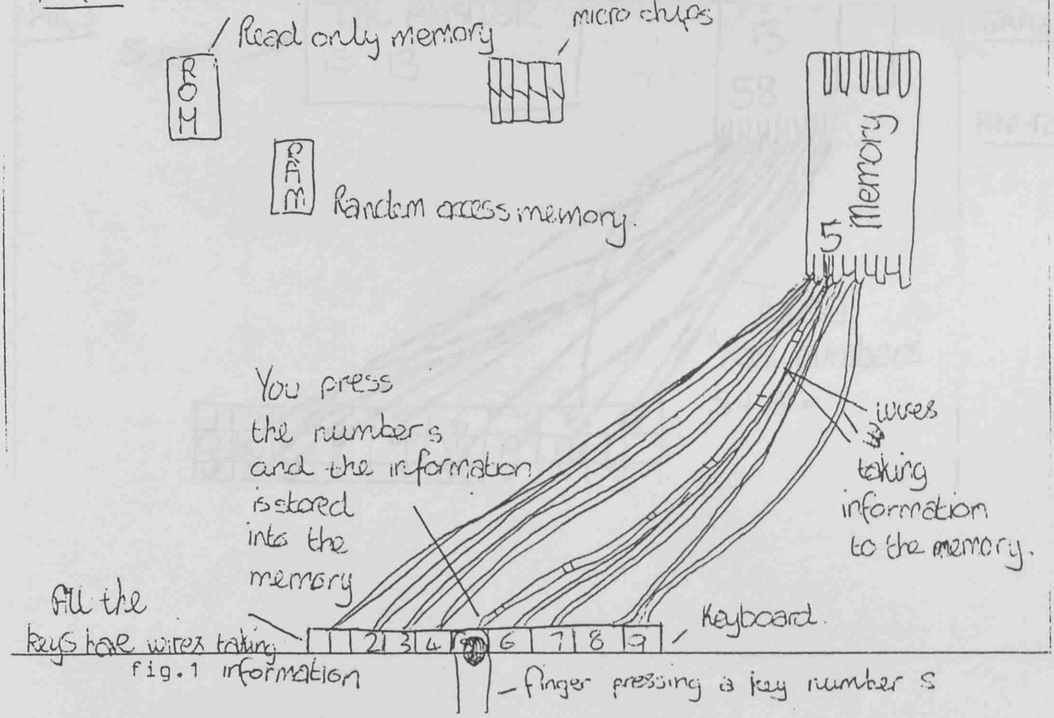
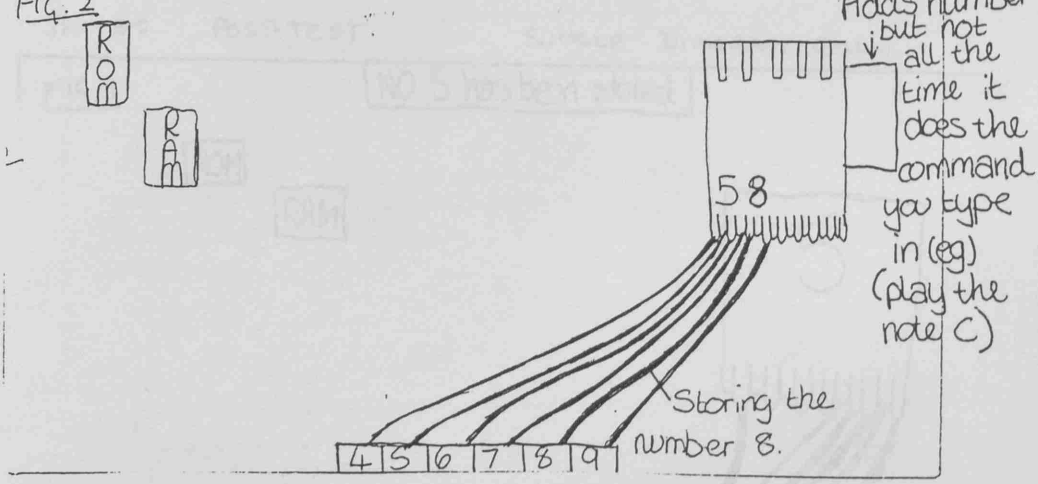
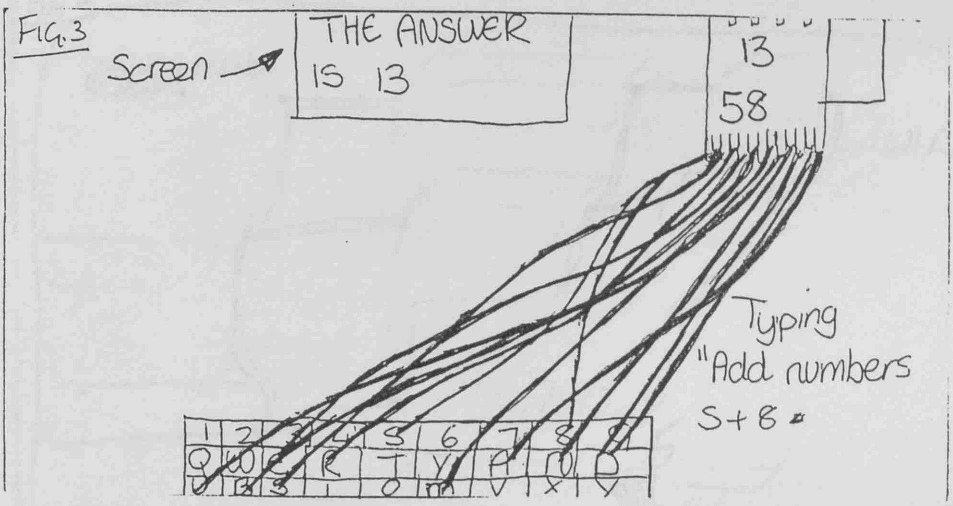


FIG. 2



JAMES

PRE-TEST



JAMES POST-TEST SINGLE DIAGRAM ONLY

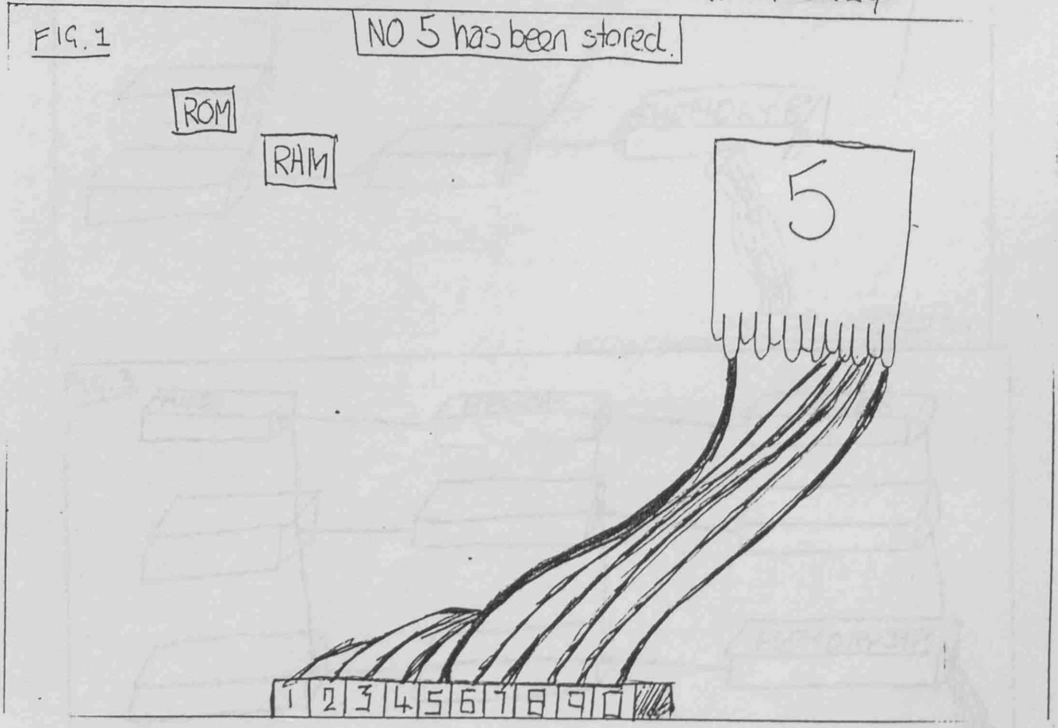


FIG 1

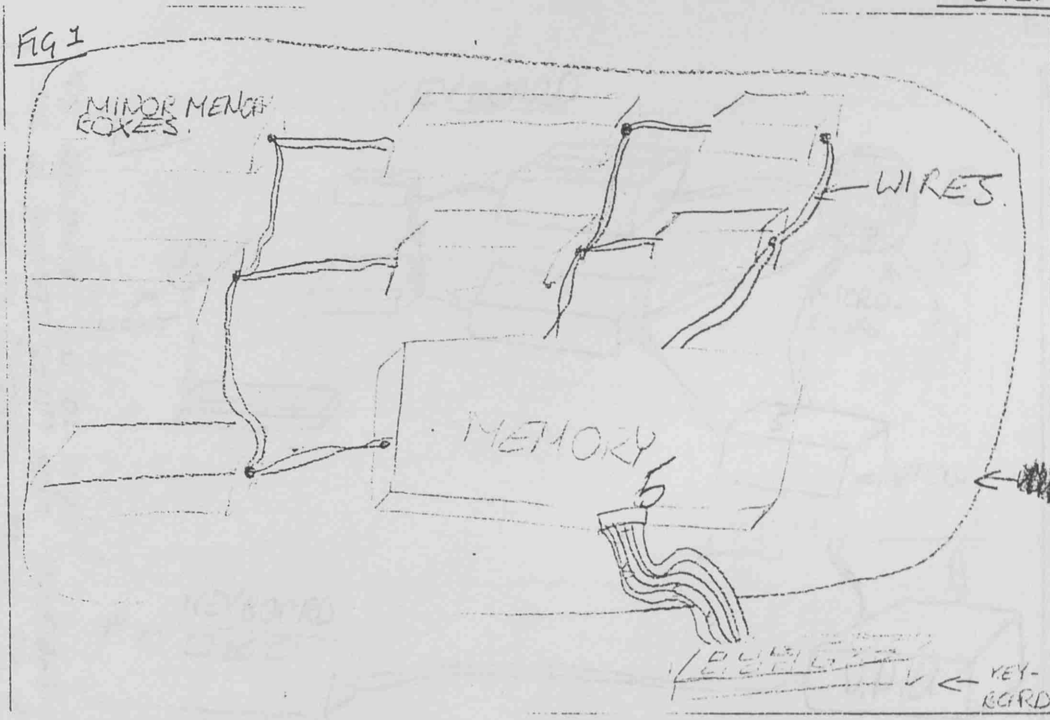


FIG. 2

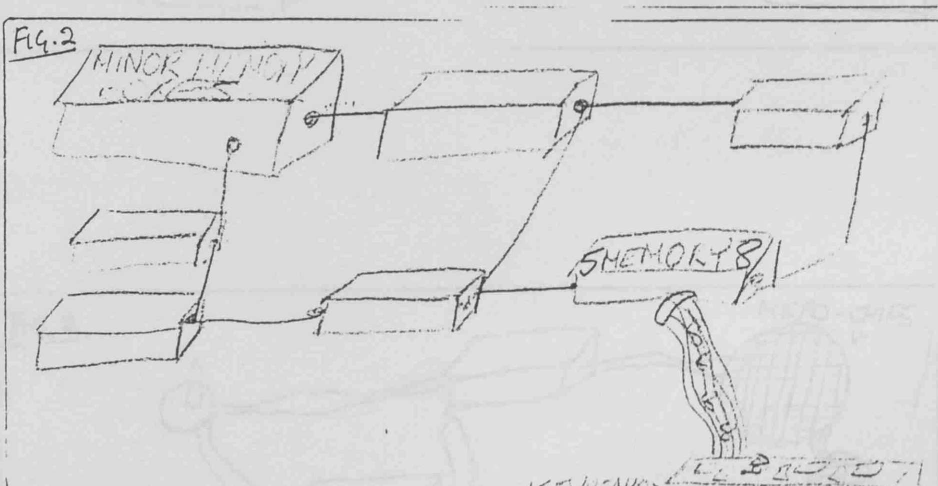
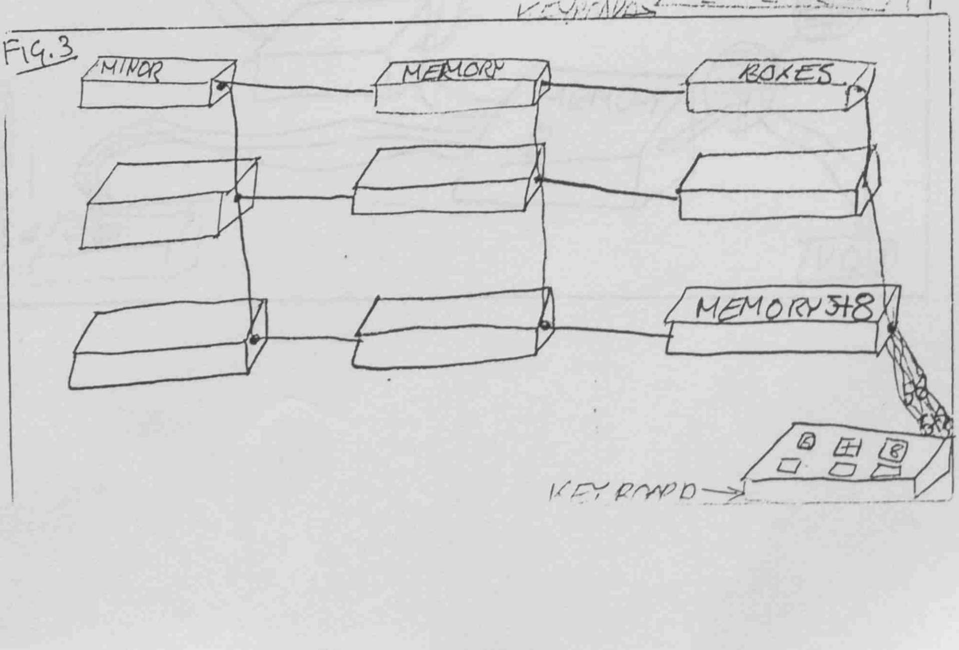
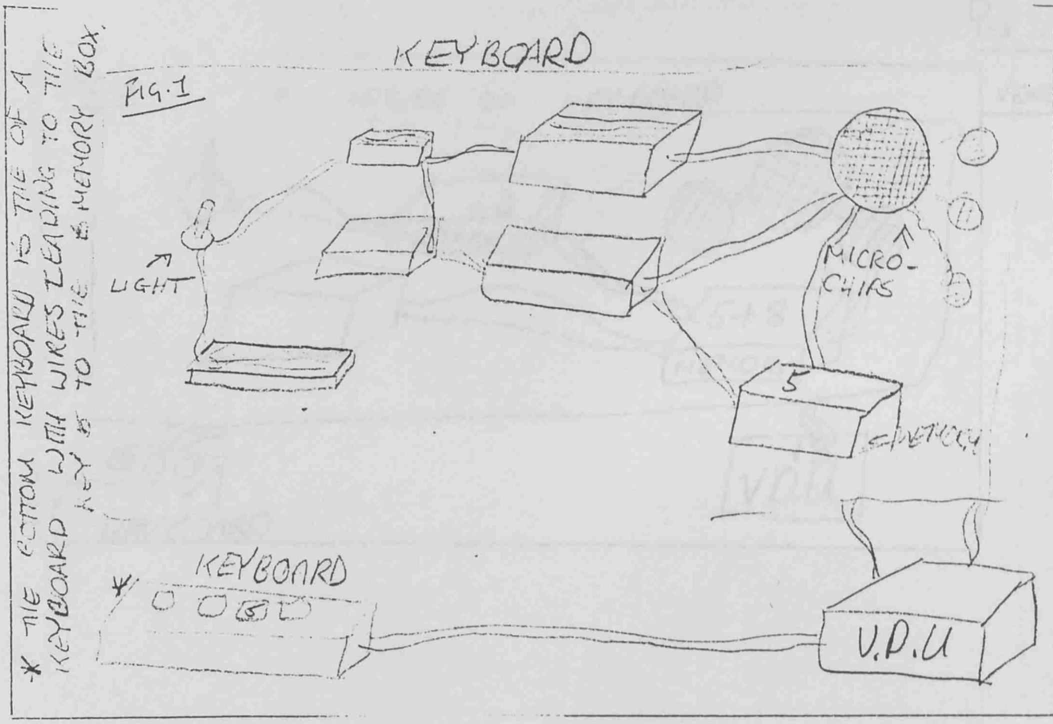


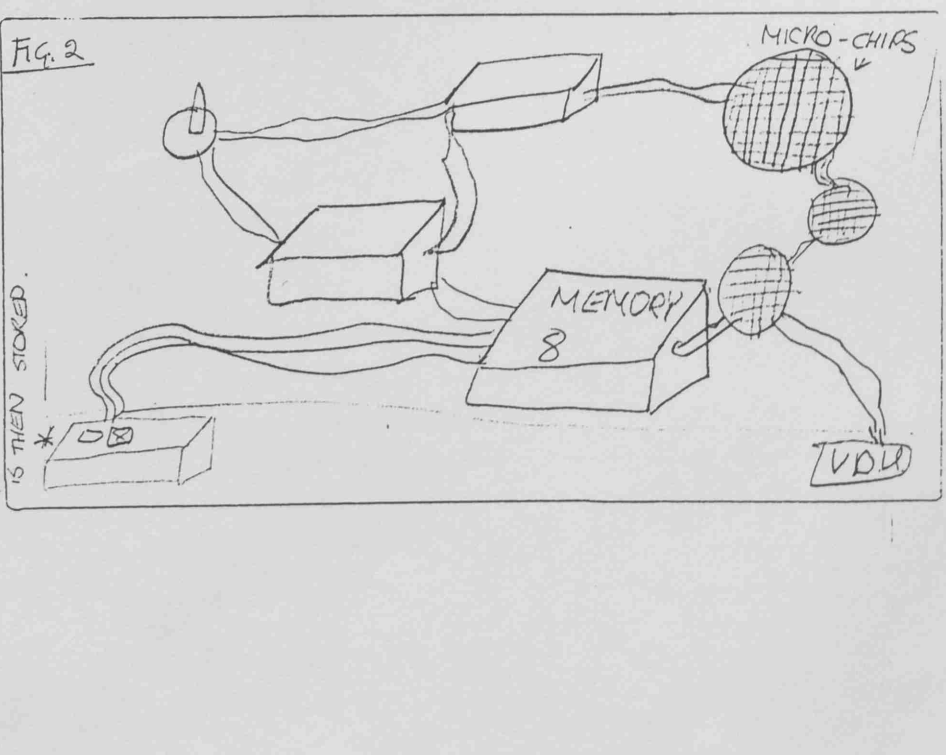
FIG. 3



Post Tes

Veneti



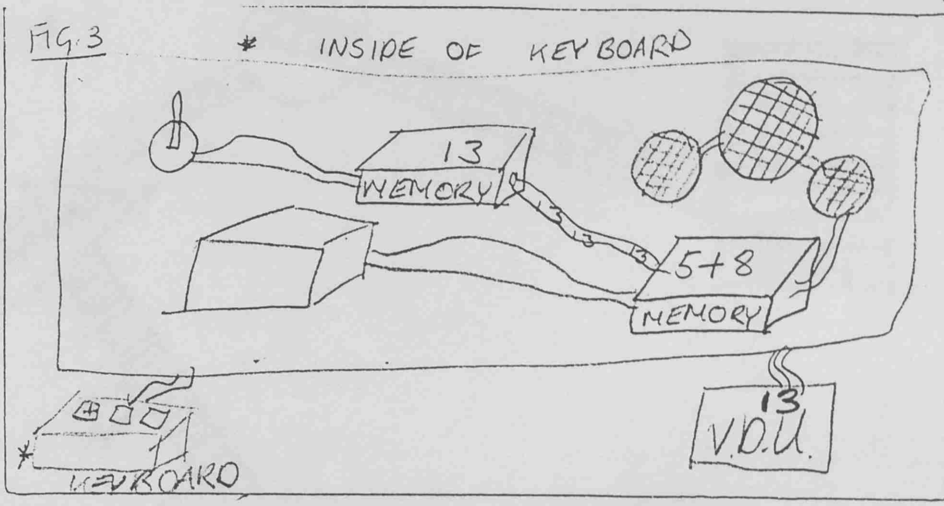


*THE BOTOM KEYBOARD IS CONNECTED BY WIRES 8 IS THEN STORED.

POST TEST

VENETIA

FIG. 3



SIA

PRETEST CLAIRE

FIG. 1

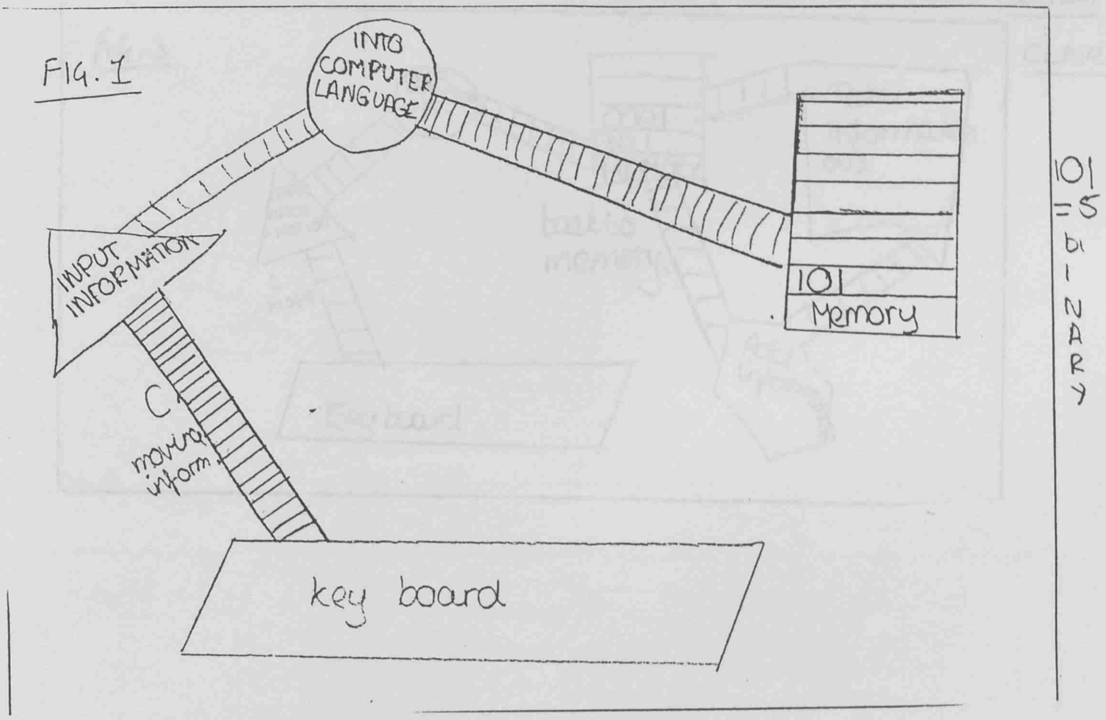
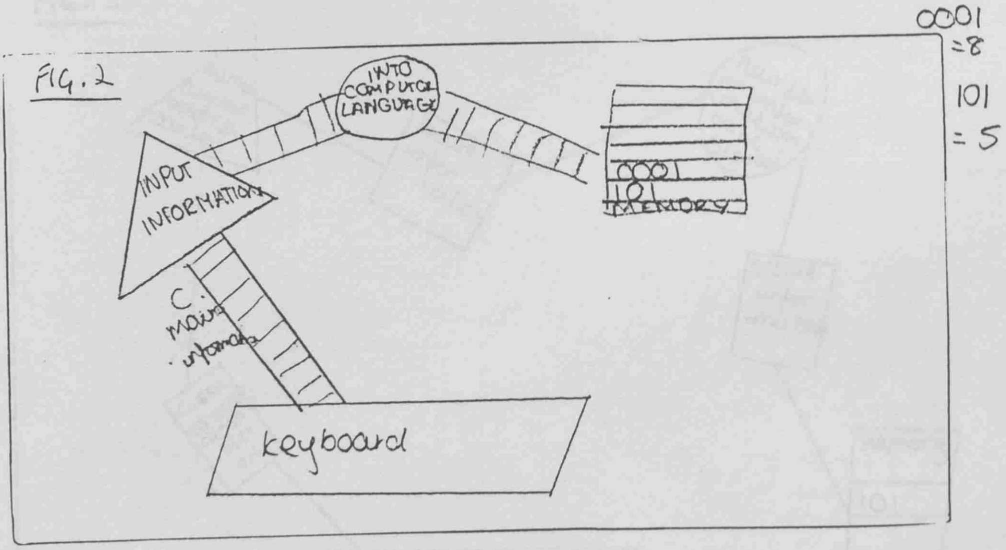
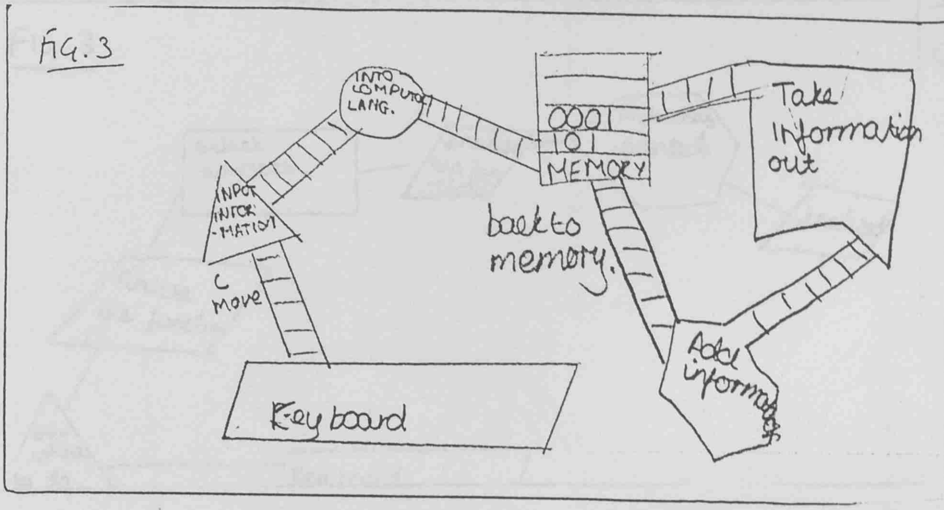


FIG. 2



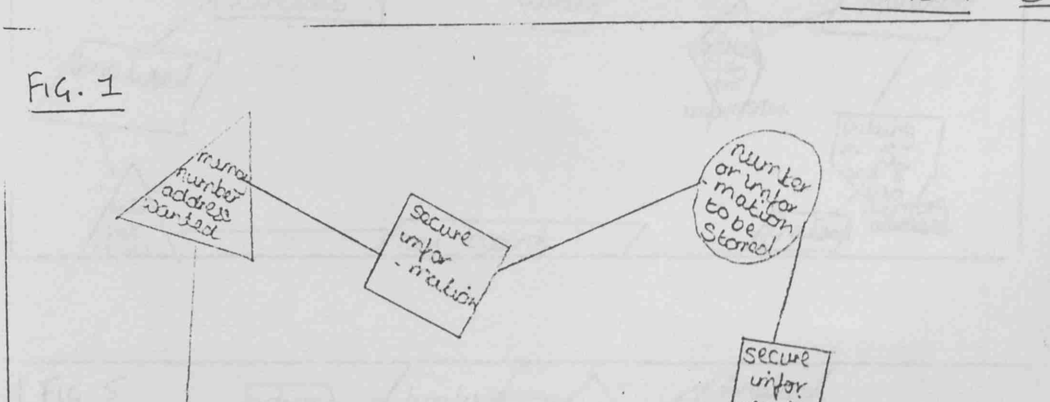
PRETEST CLAIRE

FIG. 3



POST TEST CLAIRE

FIG. 1



POST TEST

CLAIRE

FIG. 3

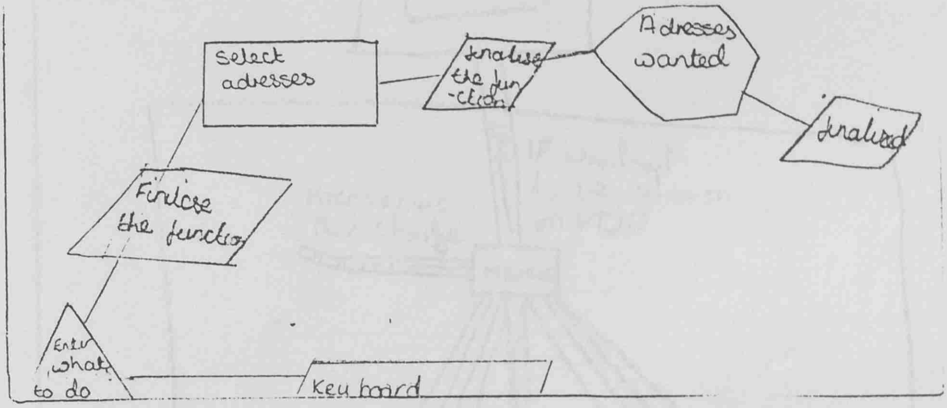


FIG 4

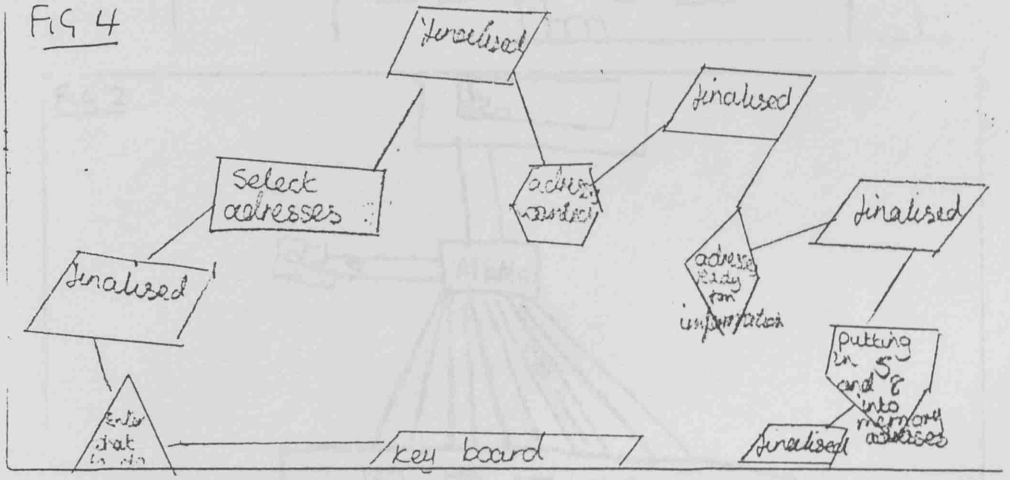
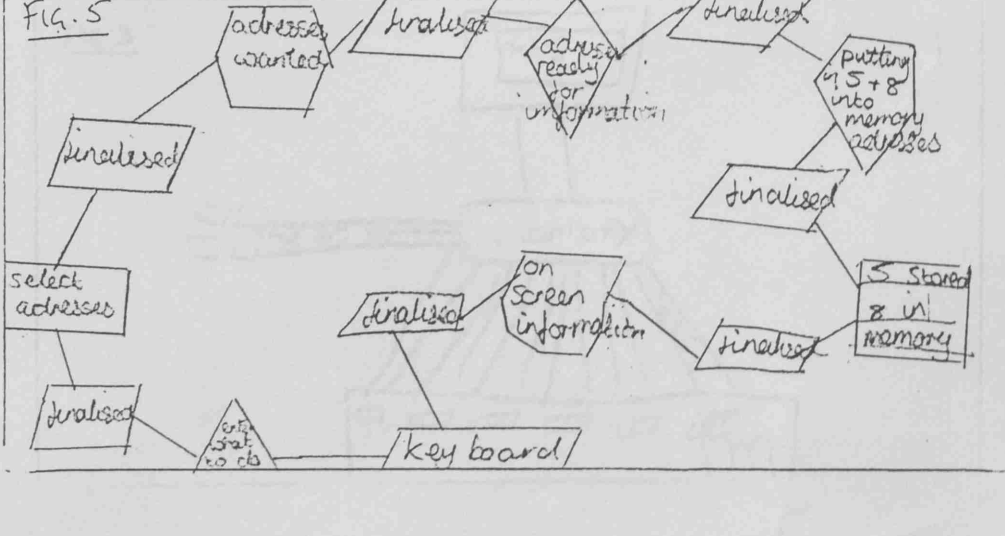


FIG. 5



S1B

PRETEST

WAYNE

FIG 1

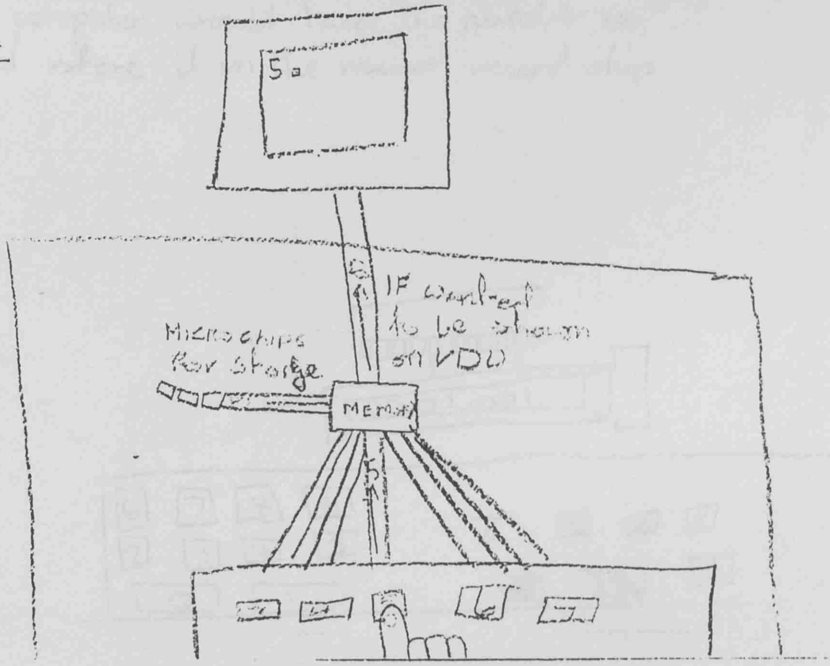


FIG 2

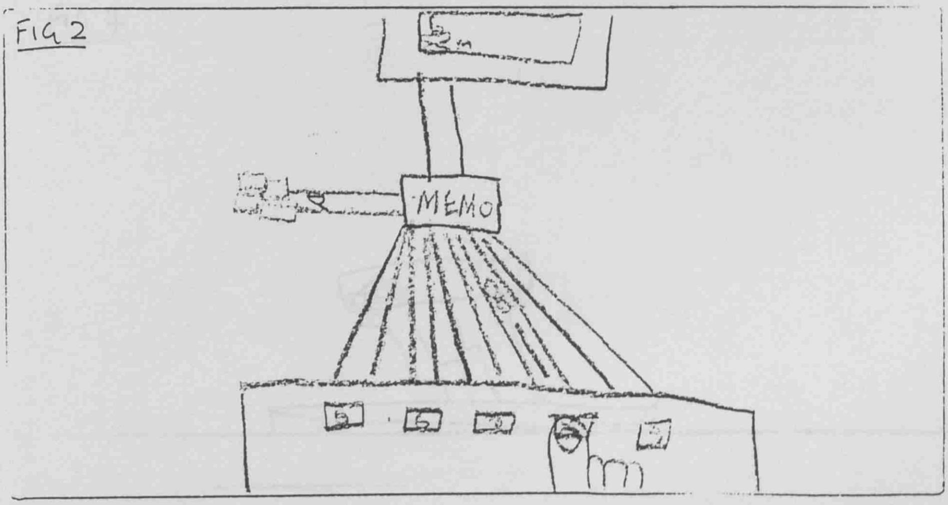
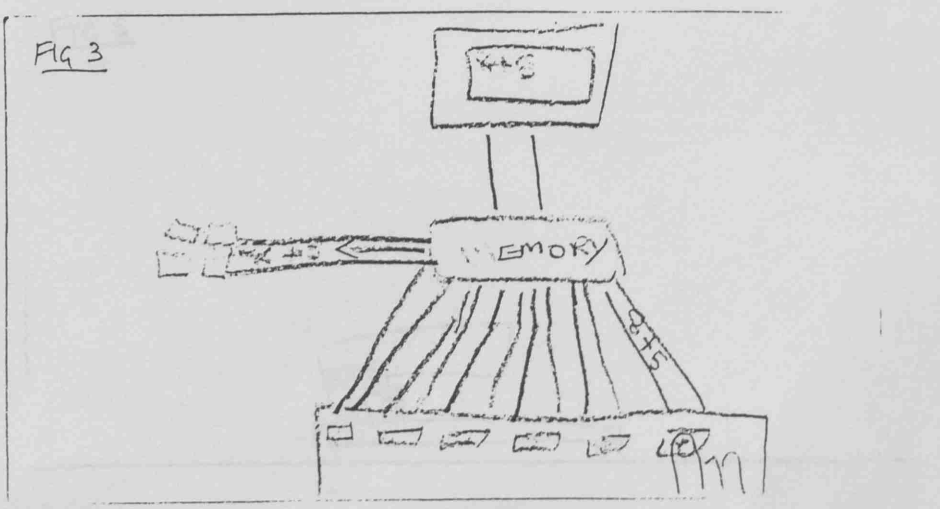


FIG 3



The computer would take the number in and store it in the main micro' chip

FIG 1

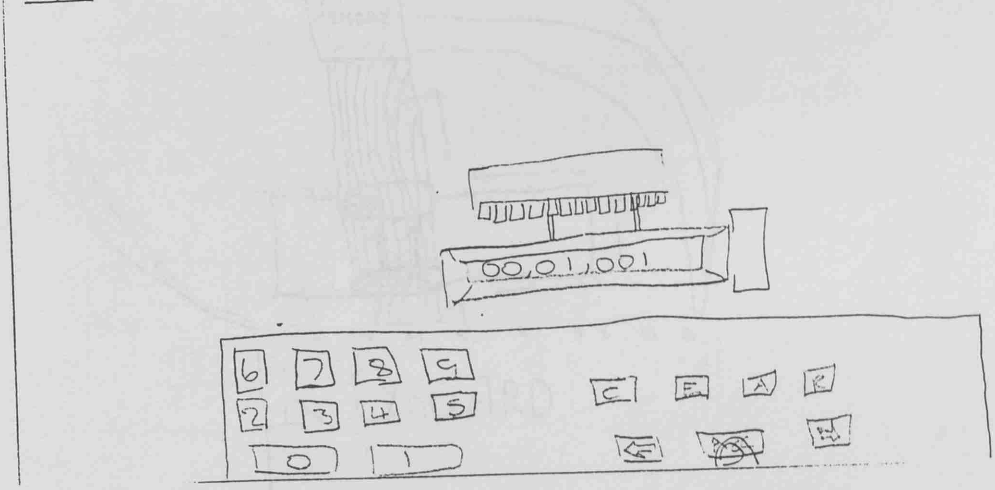


FIG. 4

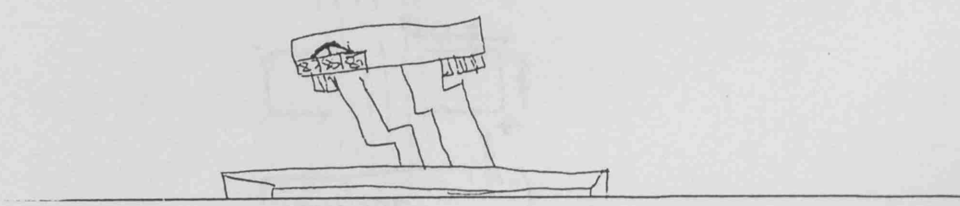
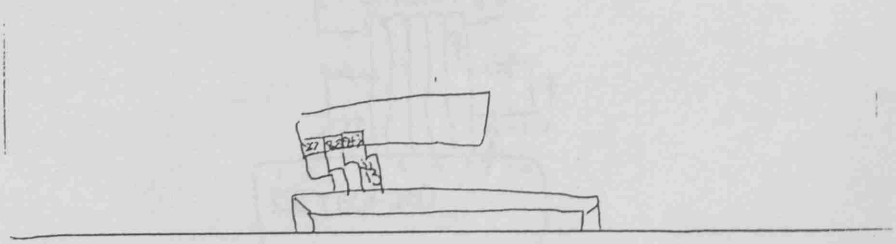


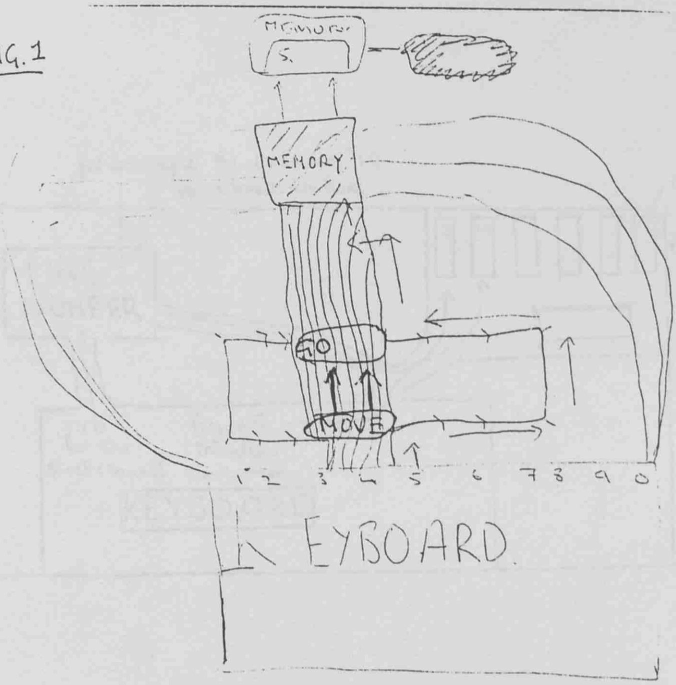
FIG 5



POST
TEST

WAYN

S2c FIG. 1



KIM

PRETO

FIG. 2

2

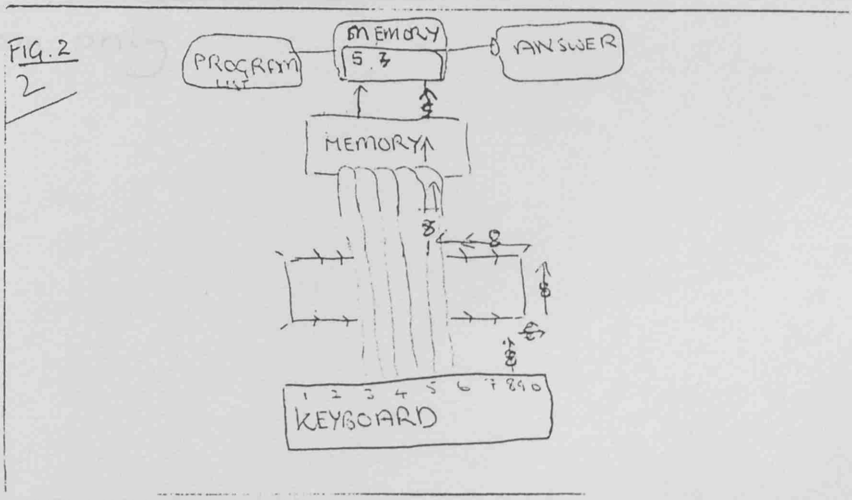


FIG. 3

3

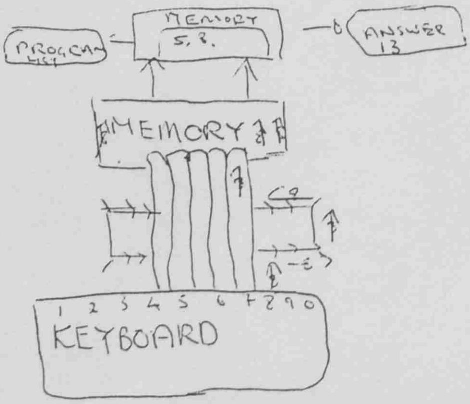
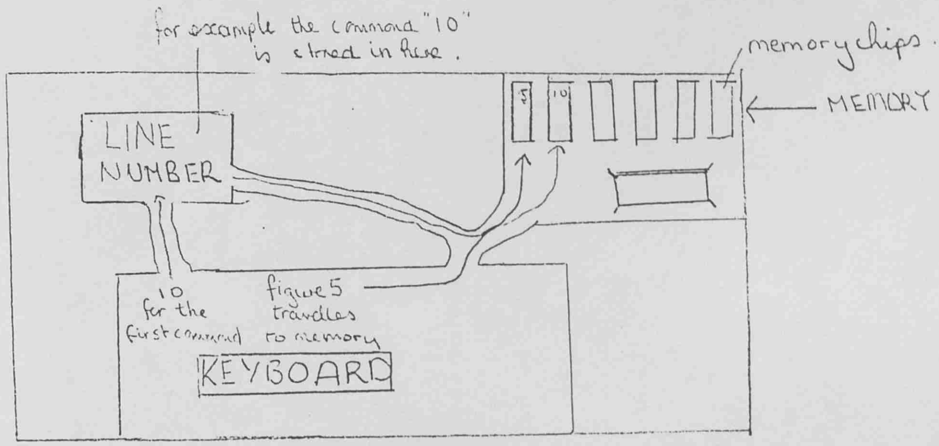


FIG 1

INSIDE OF A IMAGINARY COMPUTER

POST TEST

KIM



Single Diagram only

APPENDIX 1

CHAPTER 3 :MEYER TRANSACTION PERCENTAGE LIST — APPENDIX[1]

[Key: * = correct transaction]

TRANSACTION	PERCENTAGE
INPUT A	
Print ? on the screen *	7
Wait for a number to be entered *	23
Write the number in memory space A *	3
Write A in memory (or data list)	30
Wait for a certain number or letter	13
Print A on the screen	3
READ A	
Write next number from DATA in memory space A *	10
Print value of A in memory	10
Write letter A in memory	3
Wait for number to be entered from keyboard	3
IF A<B GOTO 99	
If the value of A is less than B move to line 99 *	63
If the value of A is more than or equal to B goto line 99	33
Move to 99(without test)	10
Print the number 99 or line 99 on the screen	10
Write A or B or A<B or a number in memory	10
LET A=B+1	
Write the obtained value in space A *	30
Write the equation in memory	43
Write B+1 in memory space A	33
Print A=B+1 or A or the value of A on the screen	23
DATA 80,99,90	
Put numbers in input queue *	27
Put numbers in memory	20
Print numbers on screen	13
Print numbers in memory space A	7

GOTO	
Move to line 60 in program *	67
Continue from there *	37
Find the number 60	10
Move to line 60 if A does not equal a certain number	7
Print the line 60 on the screen	3

PRINT C	
Print number (or 0) on the screen *	40
Print letter C on the screen	33
Write the letter C on memory	7
Print 'error' or nothing on the screen	7

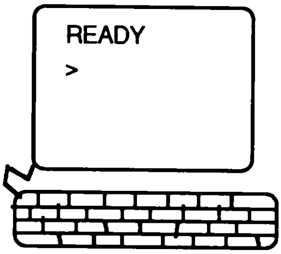
LET D=0	
Write zero in memory space D *	47
Write the equation in memory	47
Write D or 0 in memory	13
Print the equation on the screen	7

PRINT "C"	
Print letter C on the screen *	83
Write letter C in memory	7
Print the value of C on the screen	7
Find the number in memory space C	3

APPENDIX 2

For the purpose of this test we want you to understand four components of the computer.

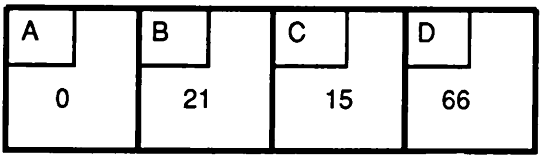
1. The display screen and keyboard.



The display screen may look like the above where READY indicates the computer is ready to accept any command or information you might type and the cursor indicates where you are on the screen.

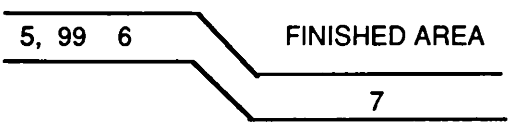
Whatever you type in and whatever the computer replies with gets displayed through the screen. The keyboard is like a typewriter it consists of keys for letters and digits special characters, for example, *+- etc. and special commands such as RETURN and CONTROL.

2. Memory



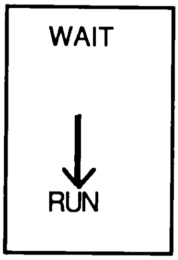
Suppose the memory of the computer consists of boxes labelled ABCD etc. with a number of other information stored in each. In the above diagram BOX labelled A contains 0, the BOX labelled B contains 21, BOX C 15 and BOX D 66.

3. Input System



Numbers wait in line to be processed in the input system. When a number is processed it drops down into the finished area. In the above diagram the numbers 5,99,6 are waiting to be processed whereas number 7 has already been processed. The next number in line to be processed is 6.

4. The Control System

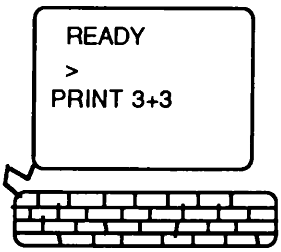


When the computer is waiting for you to type in something the arrow points to WAIT and when the computer is busy carrying out commands the arrow points to RUN.

In the above diagram the computer is RUNNING. Now that you have been introduced to four parts of the computer you are ready to try a problem.

Suppose that you typed in a one line command at the computer keyboard and then pressed RETURN such as PRINT 3+3 RETURN. Before the computer carries out this command i.e. right after you press the RETURN, parts of the computer look like this;

Screen

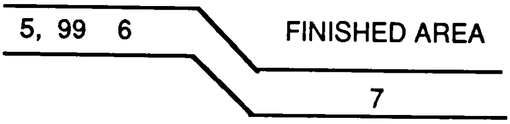


Memory

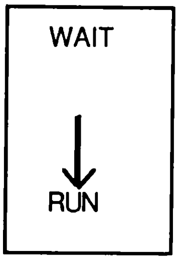
A	B	C	D
0	21	15	66

APPENDIX[2]

Input System



Control System



Can you indicate what the parts of the computer look like after this command has been carried out. Just fill in the diagram on the next page. Note : an identical set of computer 'empty' parts followed for student completion.

APPENDIX 3

LOMOND SCHOOL **PILOT STUDY:RAW SCORES** APPENDIX 3

KEY:																		
Communicat.	1																	
explicit																		
implicit																		
wires																		
transport	2																	
memory	3																	
i/output	4																	
VDU																		
Keyboard																		
graphics																		
w/processor	5																	
games/progs.																		
chip	6																	
dump	7																	
micro	8																	
Data/info.	9																	
centre/repair	10																	
on/off	11																	
subj./comp	1	1	1	1	2	3	4	4	4	4	5	5	6	7	8	9	10	11
S1A: 1	1	1				1							1		1			
2	1	1			1								1					
3	1	1			1	1						1	1	1				
4	1	1				1	1			1			1					
5	1	1			1	1						1		1				
6	1	1		1	1	1	1			1		1	1	1	1			
7	1	1		1	1								1					1
8	1	1		1		1	1					1						
9	1	1			1	1									1	1		
10	1	1			1	1							1				1	
11	1	1			1			1					1		1			
12	1	1		1	1	1							1	1				
13	1	1			1	1	1			1			1		1		1	
14	1	1		1		1	1			1			1			1		
15	1	1		1		1	1		1	1	1			1				
16	1	1			1										1			
17	1	1			1		1				1							
18	1	1		1	1													
19	1	1		1	1		1											
20	1	1			1	1												
sum	20	20	0	8	15	13	8	1	1	5	2	4	11	5	6	2	2	1

Appendix(3) cont.

subj./comp	1	1	1	1	2	3	4	4	4	4	5	5	6	7	8	9	10	11
S1B **1**	1	1			1				1		1				1			
2	1	1			1										1			
3	1	1			1						1							
4	1	1		1	1													
5	1	1			1						1			1				
6	1		1		1	1	1				1					1		
7					1													
8					1													
9	1		1	1	1								1		1	1		
10	1	1			1													
11	1	1			1													
12	1	1		1	1													
13	1	1			1	1												
14	1		1					1			1							
15	1	1			1	1						1	1	1				
16	1	1			1													
17	1	1					1											
18	1	1		1	1	1	1	1										
SUM	16	13	3	4	16	4	3	2	1	1	5	1	2	2	3	2	0	0
S1A+S1B SUM	36	33	3	12	31	17	11	3	2	6	7	5	13	7	9	4	2	1
%AGE	95	87	8	32	82	45	29	8	5	16	18	13	34	18	24	11	5	3

Percentage incidence identified in drawings and before grouping as page 82

APPENDIX 4

Extract from Alice's Adventures in Wonderland by Lewis Carroll.

Taken from Annotated Alice edited by Martine Gardner: Penguin Books,1965 pp30-31

..........she came upon a low curtain she had not noticed before, and behind it was a little door about fifteen inches high: she tried the little golden key in the lock, and to her great delight it fitted!.

Alice opened the door and found that it led into a small passage, not much larger than a rat-hole: she knelt down and looked along the passage into the loveliest garden you ever saw. "Oh, how I wish I could shut up like a telecope! I think I could, if only I knew how to begin". For, you see, so many out-of-the-way things had happened lately that Alice had begun to think that very few things indeed were really impossible.

There seemed to be no use in waiting by the little door, so she went back to the table, half hoping she might find another key on it, or at any rate a book of rules for shutting people up like a telescope: this time she found a little bottle on it("which certainly was not here before," said Alice) and tied round the neck of the bottle was a paper label with the words "DRINK ME' beautifully printed on it in large letters. It was very well to say 'DRINK ME' but the wise little Alice was not going to do that in a hurry; "No, I'll look first," she said, " and see whether it is marked poison or not": she had never forgotton that, if you drink much from a bottle marked 'poison', it is almost certain to disagree with you, sooner or later.

However, this bottle was not marked 'poison' so Alice ventured to taste it, and, finding it very nice (it had, in fact, a sort of mixed flavour of cherry-tart, custard, pineapple, roast turkey, toffy and hot buttered toast), she very soon finished it off. "What a curious feeling!" said Alice", "I must be shutting up like a telescope!" and indeed: she was only TEN INCHES HIGH.

END OF EXTRACT

TIMES HAVE CHANGED A LOT SINCE LEWIS CARROLL WROTE ALICE IN WONDERLAND, FOR EXAMPLE COMPUTERS, ARE PART OF OUR EVERYDAY LIFE.WHAT IF A MODERN DAY ALICE COULD SHUT UP LIKE A TELESCOPE NOT TO THE SIZE OF TEN INCHES BUT SAY 2 CENTIMETRES. SHE WOULD BE ABLE TO CRAWL THROUGH ONE OF THE USER PORTS OF THE BBC MICRO-COMPUTER, STAND UP AND LOOK AROUND

DRAW WHAT YOU THINK SHE WOULD SEE.

GUIDELINES FOR DRAWINGS

Name and form on the back of the drawing

A4 size if choice is available to standardise the paper used

Pupil choice on use of colour/line drawing etc.

Components of the drawing to be labelled or annoted to aid identification.

APPENDIX 5

CLFS component analysis: Staff Recorder PD.

CLFS component analysis						year 1 : 25 subjects (A5)	
subject	Com.link	Trans	Memory	Chip	I/Oput		
1	/			/	/		
2	/						
3	/			/			
4	/		/		/		
5	/				/		
6	/		/	/	/		
7	/		/	/	/		
8	/				/		
9	/						
10	/						
11	/	/					
12	/						
13	/			/			
14	/						
15	/						
16	/		/				
17	/	/	/		/		
18	/		/		/		
19	/	/	/		/		
20	/				/		
21	/						
22	/	/	/	/			
23	/	/					
24	/		/				
25	/						
SUM	25	5	9	6	10	RAW SCORES	

CATEGORY ALLOCATION - RAW SCORES

CLFS component analysis: Staff Recorder PD.

CLFS component analysis						year 2b : 28 subjects A(5)
subject	Com.link	Trans	Memory	Chip	I/Oput	
1	I	I	I		I	
2	I		I		I	
3	I		I		I	
4	I				I	
5	I					
6	I			I		
7	I		I	I	I	
8	I				I	
9	I	I		I	I	
10	I	I		I	I	
11	I		I	I	I	
12	I	I		I		
13	I		I	I		
14	I		I	I		
15	I				I	
16	I			I		
17	I		I		I	
18	I					
19	I		I	I		
20	I		I	I	I	
21	I		I			
22	I		I	I	I	
23	I		I	I		
24	I		I	I	I	
25	I	I			I	
26	I		I	I		
27	I		I		I	
28	I			I	I	
SUM	28	5	16	16	17	

CATEGORY ALLOCATION - RAW SCORES

CLFS component analysis: Staff Recorder 1D

<table>
<tr><th colspan="4">CLFS component analysis</th><th></th><th></th><th colspan="2">year 2R : 29 subjects</th></tr>
<tr><th>subject</th><th>Com.link</th><th>Trans</th><th>Memory</th><th>Chip</th><th>I/Oput</th><th></th><th></th></tr>
<tr><td>1</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td><td></td><td></td></tr>
<tr><td>2</td><td>|</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>|</td><td>|</td><td>|</td><td></td><td>|</td><td></td><td></td></tr>
<tr><td>4</td><td>|</td><td></td><td></td><td>|</td><td>|</td><td></td><td></td></tr>
<tr><td>5</td><td>|</td><td></td><td></td><td>|</td><td>|</td><td></td><td></td></tr>
<tr><td>6</td><td>|</td><td></td><td>|</td><td>|</td><td>|</td><td></td><td></td></tr>
<tr><td>7</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td><td></td><td></td></tr>
<tr><td>8</td><td>|</td><td></td><td></td><td>|</td><td></td><td></td><td></td></tr>
<tr><td>9</td><td>|</td><td></td><td>|</td><td>|</td><td>|</td><td></td><td></td></tr>
<tr><td>10</td><td>|</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>11</td><td>|</td><td>|</td><td></td><td></td><td>|</td><td></td><td></td></tr>
<tr><td>12</td><td>|</td><td></td><td></td><td></td><td>|</td><td></td><td></td></tr>
<tr><td>13</td><td>|</td><td></td><td>|</td><td>|</td><td></td><td></td><td></td></tr>
<tr><td>14</td><td>|</td><td></td><td>|</td><td></td><td>|</td><td></td><td></td></tr>
<tr><td>15</td><td>|</td><td></td><td>|</td><td></td><td>|</td><td></td><td></td></tr>
<tr><td>16</td><td>|</td><td></td><td>|</td><td></td><td></td><td></td><td></td></tr>
<tr><td>17</td><td>|</td><td></td><td>|</td><td>|</td><td>|</td><td></td><td></td></tr>
<tr><td>18</td><td>|</td><td></td><td></td><td>|</td><td>|</td><td></td><td></td></tr>
<tr><td>19</td><td>|</td><td></td><td></td><td>|</td><td></td><td></td><td></td></tr>
<tr><td>20</td><td>|</td><td></td><td>|</td><td></td><td></td><td></td><td></td></tr>
<tr><td>21</td><td>|</td><td></td><td></td><td>|</td><td></td><td></td><td></td></tr>
<tr><td>22</td><td>|</td><td></td><td>|</td><td></td><td></td><td></td><td></td></tr>
<tr><td>23</td><td>|</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>24</td><td>|</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>25</td><td>|</td><td></td><td>|</td><td></td><td></td><td></td><td></td></tr>
<tr><td>26</td><td>|</td><td></td><td></td><td>|</td><td></td><td></td><td></td></tr>
<tr><td>27</td><td>|</td><td></td><td>|</td><td>|</td><td>|</td><td></td><td></td></tr>
<tr><td>28</td><td>|</td><td></td><td>|</td><td></td><td>|</td><td></td><td></td></tr>
<tr><td>29</td><td>|</td><td></td><td>|</td><td>|</td><td>|</td><td></td><td></td></tr>
<tr><td>SUM</td><td>29</td><td>4</td><td>16</td><td>15</td><td>16</td><td></td><td></td></tr>
</table>

CATEGORY ALLOCATION - RAW SCORES

CLFS component analysis: Staff Recorder PBe

CLFS component analysis						year 3B : 23 subjects	
subject	Com.link	Trans	Memory	Chip	I/Oput		
1	/		/	/	/		
2	/		/	/	/		
3	/		/	/	/		
4	/		/	/	/		
5	/			/			
6	/		/	/	/		
7	/		/	/	/		
8	/		/	/			
9	/			/	/		
10	/	/					
11	/		/		/		
12	/						
13	/			/	/		
14	/						
15	/			/	/		
16	/		/	/			
17	/		/	/	/		
18	/	/			/		
19	/		/	/			
20	/		/				
21	/	/		/	/		
22	/	/	/	/	/		
23	/	/	/	/	/		
SUM	23	5	14	17	15		

CATEGORY ALLOCATION - RAW SCORES

CLFS component analysis: Staff Recorder (P)

CLFS component analysis						YEAR 3R : 27 subjects	
subject	Com.link	Trans	Memory	Chip	I/Oput		
1	|		|	|			
2	|			|			
3	|	|	|				
4	|			|	|		
5	|		|	|			
6	|			|	|		
7	|	|					
8	|		~~|~~				
9	|		|				
10	|		|	|			
11	|		|				
12	|	|		|	|		
13		|			|		
14	|	|	|	|	|		
15	|	|	|		|		
16	|						
17	|		|	|	|		
18	|	|	|				
19	|	|	|	|	|		
20	|						
21	|				|		
22	|		|	|	|		
23	|		|				
24	|		|	|	|		
25	|		|	|	|		
26	|			|			
27			|	|			
SUM	25	8	16	15	12		

CATEGORY ALLOCATION - RAW SCORES

CLFS component analysis: Staff Recorder JE

CLFS component analysis — year 1 : 25 subjects

subject	Com.link	Trans	Memory	Chip	I/Oput
1	1	-	-	1	1
2	1	-	-	-	-
3	1	-	-	1	-
4	1	-	1	-	1
5	1	-	-	-	1
6	1	-	1	1	1
7	1	-	1	1	1
8	1	-	-	-	1
9	1	-	-	-	-
10	1	-	-	-	-
11	1	1	-	-	-
12	1	-	-	-	-
13	1	-	-	1	-
14	1	-	-	-	-
15	1	-	-	-	-
16	1	-	1	-	-
17	1	1	1	-	1
18	1	-	1	-	1
19	1	1	1	-	1
20	1	-	-	-	1
21	1	-	-	-	-
22	1	1	1	1	-
23	1	1	-	-	-
24	1	-	1	-	-
25	1	-	-	-	-
SUM	25	5	9	6	10

CATEGORY ALLOCATION - RAW SCORES

CLFS component analysis: Staff Recorder JE

CLFS component analysis | **year 2b : 28 subjects**

subject	Com.link	Trans	Memory	Chip	I/Oput
1	1	1	1	/	1
2	1	/	1	/	1
3	1	/	1	/	1
4	1	/	/	/	1
5	1	/	/	/	/
6	1	/	/	1	/
7	1	/	1	1	1
8	1	/	/		1
9	1	1	/	1	1
10	1	1	/	1	1
11	1	1	1	1	1
12	1	/	/	1	/
13	1	/	1	1	/
14	1	/	1	1	/
15	1	/	/	/	1
16	1	/	/	1	/
17	1	/	1		1
18	1	/	/	/	/
19	1	/	1	1	/
20	1	/	1	1	1
21	1	/	1	/	/
22	1	/	1	1	1
23	1	/	1	1	
24	1	/	1	1	1
25	1	1	/	/	1
26	1	/	1	1	1
27	1	/	1	1	1
28	1	/	/	/	/
SUM	28	5	16	16	17

CATEGORY ALLOCATION - RAW SCORES

CLFS component analysis						year 2R : 29 subjects	
subject	Com.link	Trans	Memory	Chip	I/Oput		
1	1	1	1	1	1		
2	1	-	-	-	-		
3	1	1	1	-	1		
4	1	-	-	1	1		
5	1	-	-	1	1		
6	1	-	1	1	1		
7	1	1	1	1	1		
8	1	-	-	1	-		
9	1	-	1	1	1		
10	1	-	-	-	-		
11	1	1	-	-	1		
12	1	-	-	-	1		
13	1	-	1	1	-		
14	1	-	1		1		
15	1	-	1	-	1		
16	1	-	1	-	-		
17	1	-	1	1	1		
18	1	-	-	1	1		
19	1	-	-	1	-		
20	1	-	1	-	-		
21	1	-	-	1	-		
22	1	-	1	-	-		
23	1	-	-	-	-		
24	1	-	-	-	-		
25	1	-	1	-	-		
26	1	-	-	1	-		
27	1	-	1	1	1		
28	1	-	1	-	1		
29	1	-	1	1	1		
SUM	29	4	16	15	16		

CLFS component analysis: Staff Recorder JE

CLFS component analysis						year 3B : 23 subjects	
subject	Com.link	Trans	Memory	Chip	I/Oput		
1	1	—	1	1	1		
2	1	—	1	1	1		
3	1	—	1	1	1		
4	1	—	1	1	1		
5	1	—	—	1	—		
6	1	—	1	1	1		
7	1	—	1	1	1		
8	1	—	1	1	—		
9	1	—	—	1	1		
10	1	1	—	—	—		
11	1	—	1	—	1		
12	1	—	—				
13	1	—	—	1	1		
14	1	—	—	—	—		
15	1	—	—	1	1		
16	1	—	1	1	—		
17	1	—	1	1	1		
18	1	1	—	—	1		
19	1	—	1	1	—		
20	1	—	1	—	—		
21	1	1	—	1	1		
22	1	1	1	1	1		
23	1	1	1	1	1		
SUM	2~~5~~3	5	14	17	15		

CATEGORY ALLOCATION - RAW SCORES

CLFS component analysis: Staff Recorder JE

CLFS component analysis						YEAR 3R : 27 subjects	
subject	Com.link	Trans	Memory	Chip	I/Oput		
1	1	–	1	1	–		
2	1	–	–	1	–		
3	1	1	1	–	–		
4	1	–	–	1	1		
5	1	–	1	1	–		
6	1	–	–	1	1		
7	1	1	–	–	–		
8	1	–	–	–	–		
9	1	–	1	–	–		
10	1	–	1	1	–		
11	1	–	1	–	–		
12	1	1	–	1	1		
13	–	1	–	–	1		
14	1	1	1	1	1		
15	1	1	1	–	1		
16	1	–	–	–	–		
17	1	–	1	1	1		
18	1	1	1	–	–		
19	1	1	1	1	1		
20	1	–	–	–	–		
21	1	–	–	–	1		
22	1	–	1	1	1		
23	1	–	1	–	–		
24	1	–	1	1	1		
25	1	–	1	1	1		
26	1	–	–	1	–		
27	–	–	1	1	–		
SUM	25	8	16	15	12		

CATEGORY ALLOCATION - RAW SCORES

APPENDIX 6

One way Analysis of Variance: Ho: whether the subject uses 1,2,3,4 or 5 of the selected components in his/her drawing is not affected by the year-band.

Number of selected group of components 1 to 5 used by the subjects.

	X1	$X1^2$	X2	$X2^2$	X3	$X3^2$	N=132 K=3
	55	151	162	520	150	512	
mean =	2.20	-	2.84	-	3	-	

1. $GT = \frac{(55+162+150)^2}{132} = 1020.3712$

2. $SS_{TOTAL} = 151+520+512-1020.3712$

   $=162.6288$

3. $SS_{BETWEEN} = \frac{(55)^2}{25} + \frac{(162)^2}{57} + \frac{(150)^2}{50} - 1020.3712$

   $=11.04985$

4. $SS_{WITHIN} = SS_{TOTAL} - SS_{BETWEEN}$
   $=162.6288-11.04985$
   $=151.57895$

5. DEGREES OF FREEDOM
   d.f. $SS_{TOTAL}=131$
   d.f. $SS_{BETWEEN}=2$
   d.f. $SS_{WITHIN}=129$

6. ESTIMATE VARIANCES
   $VAR_{BETWEEN} = \frac{SS_{BETWEEN}}{d.f._{BETWEEN}}$
   $=5.524925$

7. $VAR_{WITHIN} = 1.1750306$

8. F= 4.7019414

SUMMARY TABLE

Source of Var.	S.Squares	d.f	Variance	F
Between groups	11.051	2	5.5255	4.7024291
Within groups	151.579	129	1.175	
TOTAL	162.63	131		

<u>Scheffe S Test</u>

Means from above: X_1=2.2 X_2=2.842 X_3 =3

VAR $_{WITHIN}$ =1.175

Wg=0.0775439

I=0.7481768 and where the confidence interval I for a gilven level lis smnaller than the difference between two means, then the means are significantly different.

GROUP 1 M1=2.2 m1-m2=0.6421 m1-m3=0.8
GROUP 2 M2=2.8421 m2-m3=0.1579
GROUP 3 M3=3
Therefore a significant differnece between the means of year one and year 3

APPENDIX 7

picture 2 component analysis

APPENDIX[7]

PICTURE2 CLFS RESULTS

Ho= The preferential ranks of component used (from the category 1-5) in years 1 to 3 are significantly different i.e. they are not directed

	%age	observed	
compon.	year 1	year 2	year 3
1	100	100	96
2	20	15.8	26
3	36	56.1	60
4	24	54.4	64
5	40	57.9	54

	rank order		
compon.	year 1	year 2	year 3
1	1	1	1
2	5	5	5
3	3	3	3
4	4	4	2
5	2	2	4

FRIEDMAN 2WAY ANALYSIS OF VARIANCE BY RANKS

compon.	1	2	3	4	5	
YEAR 1	1	5	3	4	2	
YEAR 2	1	5	3	4	2	
YEAR 3	1	5	3	2	4	
N=3 K=5	**3**	**15**	**9**	**10**	**8**	**SUM**
	9	225	81	100	64	X^2
					479	**H**

X^2r =	12*479/15*6 - 54	
=	9.866667	obtained value
=	8.533	critical value at 0.05

RESULT: THE PREFERENTIAL RANKS OF COMPONENT INCIDENCE ACROSS THE AGE RANGE ARE NOT SIGNIFICANTLY DIFFERENT THEREFORE Ho IS REJECTED AND IT IS CONCLUDED THAT GIVEN THE 5 COMPONENTS COMMUNICATION, TRANSPORTATION,MEMORY, CHIP AND I/OUTPUT THE RANKED ORDER OF INCIDENCE IS EXPECTED TO BE AS GIVEN ABOVE REGARDLESS OF THE AGE GROUP AND IT IS DIRECTED AS STATED ABOVE

picture 2 component analysis

						APPENDIX[7]
Drawing2 component analysis			year 1 : 25 subjects			
	1 = yes	0 = no				
subject	communi.	transport	memory	chip	I/o put	number of
	(1)	(2)	(3)	(4)	(5)	components 1-5
1	1	0	0	1	1	3
2	1	0	0	0	0	1
3	1	0	0	1	0	2
4	1	0	1	0	1	3
5	1	0	0	0	1	2
6	1	0	1	1	1	4
7	1	0	1	1	1	4
8	1	0	0	0	1	2
9	1	0	0	0	0	1
10	1	0	0	0	0	1
11	1	1	0	0	0	2
12	1	0	0	0	0	1
13	1	0	0	1	0	2
14	1	0	0	0	0	1
15	1	0	0	0	0	1
16	1	0	1	0	0	2
17	1	1	1	0	1	4
18	1	0	1	0	1	3
19	1	1	1	0	1	4
20	1	0	0	0	1	2
21	1	0	0	0	0	1
22	1	1	1	1	0	4
23	1	1	0	0	0	2
24	1	0	1	0	0	2
25	1	0	0	0	0	1
SUM	25	5	9	6	10	55
%age	100	20	36	24	40	
RANKED	component		Number of subjects using (1-5) components			
			No. used	obser.	%	ranked
	1		5	0	0	5
	5		4	5	20	3
	3		3	3	12	4
	4		2	9	36	1
	2		1	8	32	2

picture 2 component analysis

						APPENDIX[7]
Drawing2 component analysis			year 2b : 28 subjects			*
	1 = yes	0 = no				
subject	communi.	transport	memory	chip	i/o put	number of
	(1)	(2)	(3)	(4)	(5)	components 1-5
1	1	1	1	0	1	4
2	1	0	1	0	1	3
3	1	0	1	0	1	3
4	1	0	0	0	1	2
5	1	0	0	0	0	1
6	1	0	0	1	0	2
7	1	0	1	1	1	4
8	1	0	0	0	1	2
9	1	1	0	1	1	4
10	1	1	0	1	1	4
11	1	0	1	1	1	4
12	1	1	0	1	0	3
13	1	0	1	1	0	3
14	1	0	1	1	0	3
15	1	0	0	0	1	2
16	1	0	0	1	0	2
17	1	0	1	0	1	3
18	1	0	0	0	0	1
19	1	0	1	1	0	3
20	1	0	1	1	1	4
21	1	0	1	0	0	2
22	1	0	1	1	1	4
23	1	0	1	1	0	3
24	1	0	1	1	1	4
25	1	1	0	0	1	3
26	1	0	1	1	0	3
27	1	0	1	0	1	3
28	1	0	0	1	1	3
SUM	28	5	16	16	17	

picture 2 component analysis

						APPENDIX[7]
Drawing2 component analysis			year 2R : 29 subjects			
	1 = yes	0 = no				
subject	communi.	transport	memory	chip	i/o put	number of
	(1)	(2)	(3)	(4)	(5)	components 1-5
1	1	1	1	1	1	5
2	1	0	0	0	0	1
3	1	1	1	0	1	4
4	1	0	0	1	1	3
5	1	0	0	1	1	3
6	1	0	1	1	1	4
7	1	1	1	1	1	5
8	1	0	0	1	0	2
9	1	0	1	1	1	4
10	1	0	0	0	0	1
11	1	1	0	0	1	3
12	1	0	0	0	1	2
13	1	0	1	1	0	3
14	1	0	1	0	1	3
15	1	0	1	0	1	3
16	1	0	1	0	0	2
17	1	0	1	1	1	4
18	1	0	0	1	1	3
19	1	0	0	1	0	2
20	1	0	1	0	0	2
21	1	0	0	1	0	2
22	1	0	1	0	0	2
23	1	0	0	0	0	1
24	1	0	0	0	0	1
25	1	0	1	0	0	2
26	1	0	0	1	0	2
27	1	0	1	1	1	4
28	1	0	1	0	1	3
29	1	0	1	1	1	4
SUM	29	4	16	15	16	
TSUM	57	9	32	31	33	
ALL SECOND YEAR						
RANKED	component		Number of subjects using (1-5) components			
			No. used	obser.	%	ranked
	1		5	2	3.50877	5
	5		4	14	24.5614	3
	3		3	20	35.0877	1
	4		2	15	26.3158	2
	2		1	6	10.5263	4

picture 2 component analysis

						APPENDIX[7]
Drawing2 component analysis			year 3B : 23 subjects			
	1 = yes	0 = no				
subject	communi.	transport	memory	chip	i/o put	number of
	(1)	(2)	(3)	(4)	(5)	components 1-5
1	1	0	1	1	1	4
2	1	0	1	1	1	4
3	1	0	1	1	1	4
4	1	0	1	1	1	4
5	1	0	0	1	0	2
6	1	0	1	1	1	4
7	1	0	1	1	1	4
8	1	0	1	1	0	3
9	1	0	0	1	1	3
10	1	1	0	0	0	2
11	1	0	1	0	1	3
12	1	0	0	0	0	1
13	1	0	0	1	1	3
14	1	0	0	0	0	1
15	1	0	0	1	1	3
16	1	0	1	1	0	3
17	1	0	1	1	1	4
18	1	1	0	0	1	3
19	1	0	1	1	0	3
20	1	0	1	0	0	2
21	1	1	0	1	1	4
22	1	1	1	1	1	5
23	1	1	1	1	1	5
SUM	23	5	14	17	15	

picture 2 component analysis

						APPENDIX[7]
Drawing2 component analysis			YEAR 3R : 27 subjects			
	1 = yes	0 = no				
subject	communl.	transport	memory	chip	i/o put	number of
	(1)	(2)	(3)	(4)	(5)	components 1-5
1	1	0	1	1	0	3
2	1	0	0	1	0	2
3	1	1	1	0	0	3
4	1	0	0	1	1	3
5	1	0	1	1	0	3
6	1	0	0	1	1	3
7	1	1	0	0	0	2
8	1	0	0	0	0	1
9	1	0	1	0	0	2
10	1	0	1	1	0	3
11	1	0	1	0	0	2
12	1	1	0	1	1	4
13	0	1	0	0	1	2
14	1	1	1	1	1	5
15	1	1	1	0	1	4
16	1	0	0	0	0	1
17	1	0	1	1	1	4
18	1	1	1	0	0	3
19	1	1	1	1	1	5
20	1	0	0	0	0	1
21	1	0	0	0	1	2
22	1	0	1	1	1	4
23	1	0	1	0	0	2
24	1	0	1	1	1	4
25	1	0	1	1	1	4
26	1	0	0	1	0	2
27	0	0	1	1	0	2
SUM	25	8	16	15	12	
TSUM	48	13	30	32	27	
RANKED	component	N=50	Number of subjects using (1-5) components			
			No. used	obser.	%	ranked
	1		5	4	8	5
	5		4	14	28	2
	3		3	15	30	1
	2		2	12	24	3
	4		1	5	10	4

APPENDIX 8

CHI SQUARE TEST(1)

YEAR	NIL	DISORG.	ORG.	COMBINED		
1	0	4	6	15	N=25	OBS.
	6.25	6.25	6.25	6.25		Exp.

X^2= 6.25+0.81+0.81+0.01+12.25=19.32 DF=3
CRITICAL AT p<1%=11.34

By removing NIL category
Observed = 8.24
Critical = 5.99 df 2 at p <5%

CHI SQUARE TEST(1)

YEAR	NIL	DISORG.	ORG.	COMBINED		
2	2	1	24	29	N=56	OBS.
	14	14	14	14		Exp

X^2= 10.28+12.07+7.14+16.1=45.59 df=3 p<1%

CHI SQUARE TEST(1)

YEAR	NIL	DISORG.	ORG.	COMBINED		
3	0	0	32	17	n=49	OBS.
	12.25	12.25	12.25	12.25		Exp.

X^2=12.25+12.25+31.84+1.84=58.2 df=3

By removing Nil & Disorg categories
Observed =4.592 df1
critical = 3.84 p<5%

CHI SQUARE TEST(2)

OBSERVED

YEAR	UNORG.	ORG.	COMBINED	
1	4	6	15	**25**
2	1	24	29	**54**
3	0	32	17	**49**
	5	**62**	**61**	**128**

EXPECTED

1	O.977	12.11	11.91
2	2.1	26.16	25.73
3	1.9	23.73	23.35

X2=9.354+0.576+1.9+3+0.1783+2.882+0.8017+0.4156+1.7269
= 20.8345   DF=4   p<1% critical 13.28

APPENDIX 9

Three categories A,B and C

A Showing an intermediate step between keyboard and screen output

B Keyboard to screen with no intermediate function other than transportation of characters(used in the written program)

C Other solution or no solution to the problem. Neither keyboard nor screen representation nor intermediate step.

Group	**Number**	**A**	**B**	**C**
S1	24	3	5	16
2B	23	7	11	5
2R	27	12	3	12
3B	27	22	4	1
3R	21	6	15	0
	122	50	38	0
		41%	**31%**	**28%**

Percentage by year

		A%	**B%**	**C%**
S1	24	12.5	20.8	66.7
S2	50	38.0	28.0	34.0
S3	48	58.3	39.6	2.1

Breakdown of category A: intermediate step shown

	Number	Memory	Sort/order	Other
S1	3	1	0	2
2B	7	6	0	1
2R	12	3	1	8
3B	22	14	5	3
3R	6	4	2	0

Breakdown by year of Category A:Intermediate stage

Yr	Memory%	Sort/order	Other
S1	33.3	0.0	66.7
S2	47.4	5.2	47.4
S3	64.3	25.0	10.7

APPENDIX 10

Chi Square kxn System categories:

Observed

Yr.	Process	No Process	No Solution	
1	3	5	16	**24**
2	19	14	17	**50**
3	28	19	1	**48**
	50	**38**	**34**	**122**

Expected.

9.84	7.48	6.69
20.49	15.57	13.93
19.67	14.95	13.38

X^2=4.755+0.11+3.53+0.822+0.16+1.1+12.96+0.68+11.45=35.57 df=4
p<1% (critical 13.28)

APPENDIX 11

Chi Square kxn System Intermediate process:

Observed

Observed

Yr.	Memory	Sort/order	Other	
1	1	0	2	**3**
2	9	1	9	**19**
3	18	7	3	**28**
	28	**8**	**14**	**50**

Expected.

1.68	0.48	0.84
10.64	3.04	5.32
15.68	4.48	7.84

X^2=0.28+0.253+0.34+0.48+1.37+1.42+1.61+2.55+2.99=11.29 df=4
p<5% (critical 9.49)

APPENDIX 12

Chi Square kxn: Input to Output Model System

Observed

Year Group	No-Solution	Manual	Wire/tube	Transmission	Totals
1	13	1	8	2	**24**
2	16	4	18	12	**50**
3	5	9	17	18	**48**
Totals	**34**	**13**	**43**	**32**	**122**

Expected

6.69	2.56	8.46	6.3
13.93	5.33	17.62	13.11
13.38	5.11	16.92	12.59

X2= 5.95+0.31+0.95+2.53+0.24+0.025+0.008+0.0003+2.93+0.094+2.32
= 20.6 df=6 p<1% (critical 13.28)

APPENDIX 13

DESMOND: acronym for 'Digital Electronic System made of Nifty Devices'.

Intended for teachers of any subject wishing to learn about micro-electronics, Desmond is a self-contained micro-electronics kit developed by the Open University for its P543 and P544 Micros in Schools packs.

Desmond is based around a single chip(Motorola 6805) microcomputer incorporating on-chip interfaces, analogue-to-digital converters, 112 bytes of RAM and 3.8Kb of ROM programmed to provide DESMOND'S micro-electronics kit and simple microprocessor. There is an additional 512 bytes of RAM.

As a micro-computer, Desmond contains 100 locations of user memory. All input and output devices are memory-mapped for easy access. The processor architecture includes an accumulator and Program Counter, plus two Flags.

The Instruction Set		Mode Keys
NOP	DEC	C
construct mode		
NOT	SHR	R
RUN mode		
RET	ADD	A
attach mode		
JSR	JLO	E
erase		
JZ	JMP	
CMP	AND	
STA	LDI	
LDA		

The keyboard is composed of touch sensitive pads and is divided into a number of parts: numbers 0 to 9, keys C, E, A, R the mode keys, key <- and -> which allow backward and forward stepping, and ACC which is the 'accept' key.

Desmond takes its power from 4 Duracell MN 1400 C batteries.

The Desmond Devices:

Sending Devices - data is sent which can be used by the rest of the circuit.

Switches	S11&S12	slider type on/off
	S13	light sensitive
	S14&S15	Tilt
	S16	Push
	S17	Magnetic
	S18	Slot(light sensitive)
Sensors	U11-18	light-level sensor
	T11-18	temperature sensor(thermistor)
	V11-18	Angle sensor(potentiometer)

Receiving Devices- can receive data from a circuit and communicate to people in a useful way.

Lamps	L1	RED
	L2	YELLOW
	L3	GREEN
	L4	RED
	A1-3	Adjustable
Buzzer	B1	
Display	D18	
Motor	M1	

Internally-defined devices

Combinational elements X, Y, Z

Registers R,Q

Clocks Fast, slow

APPENDIX 14

EXAMPLE DESMOND DATA CAPTURE SHEET APPENDIX[4]

TASK2

This time imagine that your computer has stored the number 5 and you now ask it to store another number, say 8.

Finally, you ask it to add the two stored numbers together and to store the answer.

TO RE-CAP: STORE THE NUMBER 5
STORE THE NUMBER 8
ADD THE TWO STORED NUMBERS TOGETHER and
STORE THE ANSWER

Keeping the same components as your first figure 1 describe, by drawing, how your computer would carry out the problem. Make as many stages as you wish - using one frame for each stage.

(actual size 18x12cm)

Frame

Frame

APPENDIX 15

INSTRUCTIONS FOR DESMOND APPENDIX[15]

TASK 1 Subjects were told:

You are going to ask the computer to do a simple task; it has to store the number 5 in its memory. Imagine that you can see into the computer and watch the job being done.

In the space provided, draw a diagram of how you think the computer would look. The components(vocabulary defined) must include:

a). a keyboard, b). memory c). the means of getting from one component to another if required.

You can add any other part or parts as long as it is labelled clearly.

TASK 2 Subjects were told:

The computer has already stored the number 5. Next, it will be instructed to store the number 8.

When 8 has been stored the two numbers must be added together. Finally, the answer to the addition is also stored in memory.

Summary: Store the number 5/store the number 8/add the two stored numbers/store the answer.

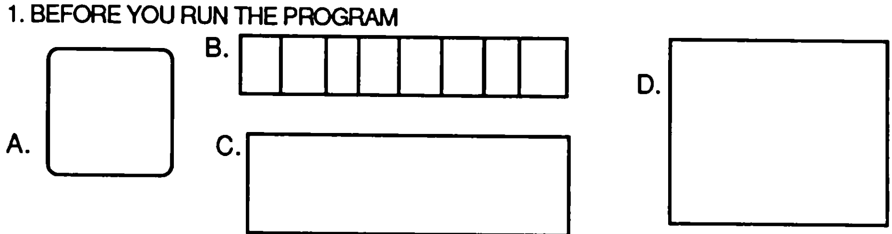
APPENDIX 16

SAMPLE DATA CAPTURE SHEET APPENDIX[16]

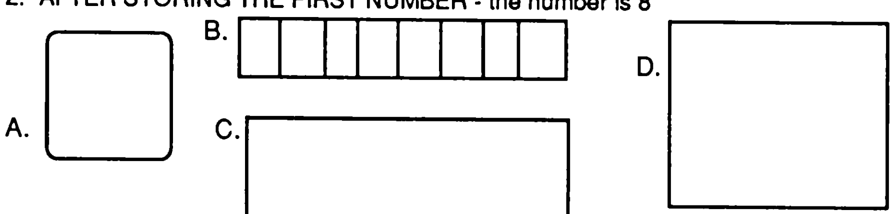
NAME:...................... FORM...................

The four sketches represent the VDU and three components of the micro-processor. For each set, 1 to 5, write onto the sketches the changes you would expect at each step of the program. The step in the program is written above each set of sketches to remind you of the program steps

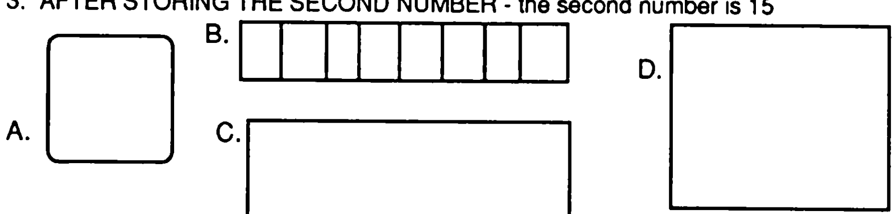
1. BEFORE YOU RUN THE PROGRAM



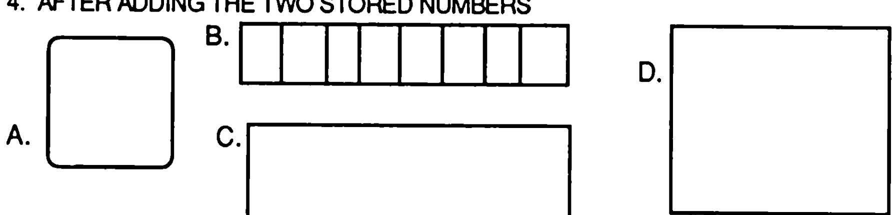
2. AFTER STORING THE FIRST NUMBER - the number is 8



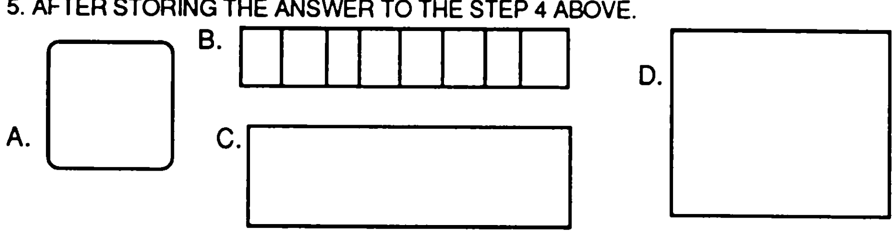
3. AFTER STORING THE SECOND NUMBER - the second number is 15



4. AFTER ADDING THE TWO STORED NUMBERS



5. AFTER STORING THE ANSWER TO THE STEP 4 ABOVE.

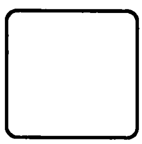


APPENDIX 17

PRECHECK INFORMATION SHEET APPENDIX[17]

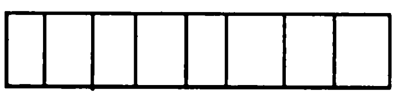
In this exercise you are asked to think about just three of the micro-computer components - ACCUMULATOR, ARITHMETIC UNIT and MEMORY SPACE and the MONITOR, which you know well. The sketch of the monitor will look a little like your real monitor or VDU but the others are just representations or pictures of what they could look like for this exercise. We are more interested in the job they do than what they really look like and below you will find the sketch and a brief description of the job that each does.

A. VDU



A. this is where you see what you type and where the results of your program appear.

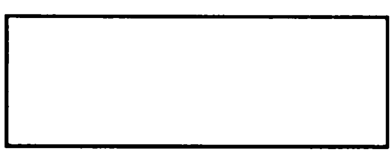
B. ACCUMULATOR



B. small temporary memory with space for 8 bits. 1Bit=1 or 0

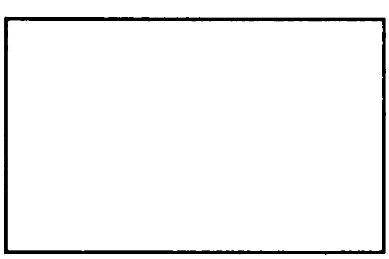
C. ARITHMETIC UNIT

C. where numbers from memory can be used in addition, subtraction, division and multiplication, +-x /



D. MEMORY SPACE

D.this represents the main memory space and you should arrange or label you data in the way you think best within this space.



APPENDIX 18

Table(1)PRE-TEST RAW SCORES:Memory

Year	stage1	stage2	stage3	stage4	stage5
1	21	8	7	11	1
2	11	3	3	0	0
3	40	20	13	18	8

Table(2)POST-TEST RAW SCORES:Memory

Year	stage1	stage2	stage3	stage4	stage5
1	25	10	6	8	5
2	11	1	0	0	1
3	36	19	13	17	9

Table(3) RANKED PRE-TEST SCORES:Memory

Year	stage1	stage2	stage3	stage4	stage5
1	5	3	2	4	1
2	5	3.5	3.5	1.5	1.5
3	5	4	2	3	1

Table(4) RANKED POST-TEST SCORES:Memory

Year	stage1	stage2	stage3	stage4	stage5
1	5	4	2	3	1
2	5	3.5	1.5	1.5	3.5
3	5	4	2	3	1

Memory Raw Scores at Pre and Post test

APPENDIX 18

YEAR 1 PRE TEST (27 subjects)

Subject/stage 1-5 PRE	1	2	3	4	5	total
1	1					1
2	1	1	1	1		4
3	1					1
4	1					1
5	1					1
6	1			1		2
7	1	1	1	1		4
8	1					1
9	1					1
10	1	1	1	1		4
11	1	1		1		3
12	1	1	1	1		4
13	1					1
14	1			1		2
15	1				1	2
16	1	1	1	1		4
17	1	1	1	1		4
18	1					1
19	1	1	1	1		4
20	0					0
21	0					0
22	0					0
23	0					0
24	0					0
25	1					1
26	1			1		2
27	0					0
Tot. Pre	21	8	7	11	1	48

Mean Number blanks per subject: 1.78

YEAR 1 POST TEST

stage 1-5 Post	1	2	3	4	5	total
1	1					1
2	1	1	1	1		4
3	1					1
4	1					1
5	1	1	1			3
6	1					1
7	1	1		1		3
8	1					1
9	1					1
10	1		1	1		3
11	1					1
12	1	1	1	1		4
13	1					1
14	1	1				2
15	1				1	2
16	1			1		2
17	1				1	2
18	1	1	1	1	1	5
19	1	1	1	1	1	5
20	1					1
21	1	1				2
22	1					1
23	1	1		1	1	4
24	1					1
25	0					0
26	1	1				2
27	0					0
Tot. Post	25	10	6	8	5	54

Mean Number blanks per subject:2.

RANK ORDER numerical magnitude

Pre	stage
1st	5
2nd	3
3rd	2
4th	4
5th	1

RANK ORDER numerical magnitude

Post	stage
1st	5
2nd	3
3rd	4
4th	2
5th	1

Memory Raw Scores at Pre and Post test

APPENDIX 18

	YEAR 2 PRE TEST (11 subjects)						YEAR 2 POST TEST					
Subject/stage 1-5							stage 1-5			Post		
PRE	1	2	3	4	5	total	1	2	3	4	5	total
1	1					1	1					1
2	1		1			2	1				1	2
3	1	1	1			3	1					1
4	1		1			2	1					1
5	1	1				2	1					1
6	1	1				2	1					1
7	1					1	1					1
8	1					1	1					1
9	1					1	1					1
10	1					1	1	1				2
11	1					1	1					1
Totals	11	3	3	0	0	17	11	1	0	0	1	13
Pre							Post					

Mean Number blanks per subject: 1.55

Mean Number blanks per subject: 1

RANK ORDER numerical magnitude		RANK ORDER numerical magnitude	
Pre	stage	Post	stage
1st	4/5	1st	3/4
2nd		2nd	
3rd	2/3	3rd	2/5
4th		4th	
5th	1	5th	1

Memory Raw Scores at Pre and Post test

													APPENDI
YEAR 3 PRE TEST				(43 subjects)				**YEAR 3 POST TEST**					
Subject/stage 1-5								**stage 1-5**			**Post**		
PRE	1	2	3	4	5	total		1	2	3	4	5	total
1	1	1	1	1		4		1		1	1	1	4
2	1			1		2		1				1	2
3	1					1		1					1
4	1	1			1	3		1			1		2
5	1			1		2		1	1	1	1	1	5
6	1	1		1	1	4		1	1	1	1		4
7	1	1	1	1		4		1	1	1	1		4
8	1	1	1	1		4		1	1	1	1		4
9	1					1		1					1
10	1					1		0					0
11	1					1		1					1
12	1	1	1	1		4		1					1
13	1				1	2		1	1	1	1		4
14	1	1	1	1		4		1			1		2
15	1					1		1			1		2
16	1					1		1	1	1	1		4
17	1					1		1	1	1	1		4
18	1	1	1	1		4		1	1	1	1		4
19	1	1		1		3		0					0
20	0					0		0					0
21	1	1	1	1		4		1	1	1	1		4
22	1	1	1	1		4		1	1		1	1	4
23	1	1	1	1	1	5		1	1	1	1	1	5
24	1	1	1	1	1	5		1				1	2
25	0					0		0					0
26	1	1				2		1	1				2
27	1	1		1		3		1	1				2
28	1					1		1	1				2
29	0					0		1					1
30	1					1		1	1	1	1		4
31	1	1				2		1	1				2
32	1	1		1	1	4		1	1		1	1	4
33	1					1		1					1
34	1					1		0	1	1		1	3
35	1	1	1		1	4		1	1			1	3
36	1				1	2		0					0
37	1			1		2		1					1
38	1	1	1			3		1					1
39	1					1		1					1
40	1					1		1					1
41	1					1		0					0
42	1					1		1					1
43	1	1	1	1		4		1					1
	40	20	13	18	8	99		36	19	13	17	9	94
Mean Number blanks per subject: 2.3								Mean Number blanks per subject: 2					

18

APPENDIX 19

Kendall's Coeficient of Concordance 'w'
Memory content Blank or non-Blank over 5 stages

PRE-TEST rank order(Appendix[ ])

Year	stage 1	2	3	4	5
1	5	3	2	4	1
2	5	3.5	3.5	1.5	1.5
3	5	4	2	3	1
R_j	15	10.5	7.5	8.5	3.5

ER_j=45 M=9
K=3 N=5
Adj. for row 2 K_2=(2^3-2)+(2^3-2)/12=1 therefore adj=1

Result
(1) Sum ranks 45 (2) Mean =9 (3)S= 36+2.25+2.25+0.25+30.25
therefore **S =71**

(4)W=S/ 0.0833*k^2(N^3-N)-K(adj) therefore **W=71/87=0.816092**

N<7 therefore significance of 'w' found through S. S=71 and is significant for p< 5% (critical 64)

POST-TEST rank order

Year	stage 1	2	3	4	5
1	5	4	3	2	1
2	5	3.5	1.5	1.5	3.5
3	5	4	2	3	1
R_j	15	11.5	5.5	7.5	5.5

ER_j=45 M=9
K=3 N=5
Adj. for row 2 K_2=(2^3-2)+(2^3-2)/12=1 therefore adj=1

Result
(1) Sum ranks 45 (2) Mean =9 (3)S= 36+6.25+12.25+2.25+12.25
therefore **S =69**

(4)W=S/ [1/12*k^2(N^3-N)]-K(adj) therefore **W=69/87=0.7931034**

N<7 therefore significance of 'w' found through S. S=69 and is significant for p< 5% (critical 64)

APPENDIX 20

CHI SQUARE X^2
POST-TEST ADDRESSED OR NON/ADDRESSED

<u>Observed</u>

Year	Addressed	Non-addressed	Total
1	10	17	27
2	9	2	11
3	9	34	43
Totals	28	53	81

<u>Expected</u>

	9.3	17.67
	3.8	7.2
	14.9	28.14

0.05+7.12+2.3+0.03+3.76+1.2=14.46

<u>Chi Square X^2=14.46</u>

<u>df=2 p<1% (critical 9.21)</u>

APPENDIX 21

STAGE 5 MEMORY BOX CONTENTS APPENDIX[2(]

YEAR 1

PUPIL	CONTENTS STAGE 5 PRE-TEST	POST-TEST
1	8+15=23	?115=23
2	23	23
3	8 15 23	8 15 23
4	8+15=23	[129] 8 [13])15 [131]8+15=23(addressed)
5	23 15 8	8+15=23
6	8 15 23	?141 23
7	[1](23) (addressed)	[115]8 [116] 15 [117] 23(addressed)
8	8+15=23	08+15=23
9	8+15=23	[1]8 [2] 15 [3]8+15=23(addressed)
10	8+15=23	15+8=23
11	8+15=23	[250]8+[251]15=[252]23(addressed)
12	23	118=116+117=5
13	8+15=23	[1]23 [5])15 [55] 8(addressed)
14	8+15=23	[1] 15 [2]+[3]8 [9]= [5]23
15	8 15 23	BLANK
16	Program	23(in one cell of a grid)
17	8+15=23	BLANK
18	8 15 23	BLANK
19	BLANK	BLANK
20	8,15,23	?202=10101 [200]8 [210]15 [203]23(A)
21	23+program	8+15
22	8+15=23	[120] 8 [121]15 [122] 23
23	8+15=23	BLANK
24	23	8+15=23
25	23	33
26	8+15=23	23
27	Program	23

YEAR 1 PRE-TEST ANSWER TYPES REDUCED TO 7 different ITEMS

1	BLANK
2	8 15 23
3	8+15=23
4	23
5	[1](23) (addressed)
6	Program
7	23 and Program

CONTENTS STAGE 5 11 items

POST-TEST

1. BLANK
2. 8+15
3. 8+15=23
4. 8 15 23
5. 23
6. 118=116+117=5
7. [1] 15 [2]+[3]8 [9]= [5]23
8. 23(in one cell of a grid)
9. ?115=23
10. ?202=10101 [200]8 [210]15 [203]23(A)
11. [1]8 [2] 15 [3]8+15=23(addressed)
12. [120] 8 [121]15 [122] 23

STAGE 5 MEMORY BOX CONTENTS
YEAR 2

PUPIL	CONTENTS STAGE 5 PRE-TEST	POST-TEST
1	1011_2	[6]15 [4]8 [7]23
2	23	BLANK
3	23	[1] 8[5] 15
4	8+15=23	[2]200 [4] 600
5	23	[1]8 [2]15 [3]23
6	23	2 3
7	8+15=23	[128]8 [5]15 [6]23
8	23	[1]8 [5]15 [7]23
9	8+15=23	[1]8 [2]15 [3]23
10	23	[1]8-[2]15=7
11	23	[1]8 [2]15 [3]23

PRE-TEST 3 DIFFERENT ANSWER TYPES :

1. 1011_2
2. 23
3. 8+15=23

POST-TEST 6 DIFFERENT ANSWER TYPES:

BLANK
2 3
[1] 8 [5] 15
[2]200 [4] 600
[1]8-[2]15=7
[6]15 [4]8 [7]23

STAGE 5 MEMORY BOX CONTENTS
YEAR 3

PUPIL	CONTENTS STAGE 5 PRE-TEST	POST-TEST
1	23	BLANK
2	23	BLANK
3	8+15=23	{[10]=8 [11]=15 }23
4	BLANK	23
5	23	BLANK
6	Answer	23
7	Answer is 23	Print ?8=15
8	Answer is 23	8+15=23
9	23	23
10	1111 1000 10111	[200]8 [210]15 [220]23
11	23	23
12	23	8,15
13	8+15=23	[201] 8 [202]15 [203]?201+?202 23
14	BLANK	23
15	23	23
16	23	23
17	23	23
18	23	23
19	23	23
20	10110 (binary 22)	10110 (binary 22) assume mean 23
21	23	23
22	23	23
23	23	BLANK
24	BLANK	BLANK
25	BLANK	BLANK
26	BLANK	[1] 8 [2] 15 [3]23
27	23	[100] 8 [101] 15 [102] 23
28	23	23
29	Program, 23	8 15
30	8,15,23	23
31	23	23
32	23	?100=15?116=8?120=23
33	BLANK	BLANK
34	23	23
35	23	15
36	BLANK	BLANK
37	BLANK	[380]23
38	23	23
39	23	8 15 23
40	23	23
41	8 15 23	25 10 15
42	23	180(an address)
43	23	8+15=23

YEAR 3 PRE-TEST 7 DIFFERENT ANSWER TYPES

	CONTENTS STAGE 5 PRE-TEST
1	23
2	8+15=23
3	BLANK
4	Answer
5	8+15=23
6	8,15,23
7	Program, 23

YEAR 3 POST-TEST 13 DIFFERENT ANSWER TYPES
CONTENTS STAGE 5 POST-TEST

1. BLANK
2. 15
3. Print ?8=15
4. 8,15
5. 8+15=23
6. 23
7. 8 15 23
8. 180(an address)
9. [380]23
10. ?100=15?116=8?120=23
11. [201] 8 [202]15 [203]?201+?202 23
12. {[10]=8 [11]=15 }23
13. [200]8 [210]15 [220]23

APPENDICES 22&23

Two way Analysis of Variance: Memory:Summation to F

Source of Var.	df	SSquares	MSquares	F	p
Yearband	2	3.953	1.977	5.850	0.0043
Pre/post test	1	4.440	4.440	19.261	0.0000
Interaction	2	1.731	0.866	3.755	0.0277

AB Simple effects one level

Effect	MSn	DFn	DFe	MSe	F	p
A at B1	0.229	2	78	0.109	2.089	0.130
A at B2	2.613	2	78	0.459	5.690	0.005
B at A1	0.667	1	78	0.231	2.892	0.093
B at A2	2.909	1	78	0.231	12.618	0.001
b at A3	0.942	1	78	0.231	4.085	0.047

Tukey(hsd)Pairwise comparison(For A)

	A	B	C
A. A 3	X	-	s
B. A 1	-	X	-
C. A 2	s	-	X

Tukey(hsd)Pairwise comparison(For B)

A.B 1	X	s
B.B 2	s	X

APPENDIX[23]

Two way Analysis of Variance: VDU:Summation to F

Source of Var.	df	SSquares	MSquares	F	p
Year band	2	3.222	1.611	0.836	0.4372
Pre/Post test	1	4.440	4.440	4.812	0.0312
Interaction	2	11.295	5.648	6.120	0.0034

AB Simple effects one level

Effect	MSn	DFn	DFe	MSe	F	p
A at B1	4.049	2	78	1.863	2.173	0.121
A at B2	3.210	2	78	0.986	3.254	0.044
B at A1	0.667	1	78	0.923	0.722	0.398
B at A2	0.409	1	78	0.923	0.443	0.508
B at A3	31.442	1	78	0.923	34.070	0.000

Tukey(hsd)Pairwise comparison(For A)

	A	B	C
A. A 1	X	-	-
B. A 3	-	X	-
C. A 2	-	-	X

Tukey(hsd)Pairwise comparison(For B)

A.B2	X	s
B. B1	-	X

APPENDIX 24

Memory storage of Equation or no-Equation

PRE-TEST:**observed**

Year	Equation	No-Equation	Total
1	12	15	**27**
2	3	8	**11**
3	2	41	**43**
	17	**64**	**81**

expected

	5.67	21.33
	2.3	8.69
	9.02	33.98

Result

X^2=7.07+0.21+5.46+1.88+0.055+1.45 = 16.125

X^2=16 p<1% (critical 9.21)

POST-TEST:**observed**

Year	Equation	No-Equation	Total
1	8	19	**27**
2	0	11	**11**
3	2	41	**43**
	10	**71**	**81**

expected

	3.3	23.67
	1.36	9.64
	5.31	37.69

Result

X^2=6.69 +1.36+2.06+0.92+0.19+0.29

X^2=11.5 p<1% (critical 9.21)

APPENDIX 25

VDU component: at each of the 5 stages the was examined to see whether it was BLANK or NOT BLANK each subject has a disposable score of 5:

PRE-TEST VDU(1)

OBSERVED

YEAR	BLANK	NON BLANK	TOTALS
1	24	111	**135**
2	10	45	**55**
3	74	141	**215**
TOTALS	**108**	**297**	**405**

EXPECTED

YEAR	BLANK	NON BLANK
1	36	99
2	14.7	40.3
3	57.3	157.7

X^2=6+1.4545+1.5+0.55+5+1.77=16.2745

Critical value p<0.01=9.2
Observed value 16.2745

Ho rejected, conclude that there is a significant relationship between the belief held about VDU state BLANK or NOT BLANK and the year-band of the subject at PRE-TEST.

POST-TEST VDU(1)

OBSERVED

YEAR	BLANK	NON BLANK	TOTALS
1	16	119	**135**
2	13	42	**55**
3	27	188	**215**
TOTALS	**56**	**349**	**405**

EXPECTED

YEAR	BLANK	NON BLANK
1	18.7	116.3
2	7.6	47.4
3	298.7	185.3

X^2=0.39+3.84+0.25+0.06+0.62+0.039=5.199

Critical value p<0.01=9.2
Observed value 5.199

Therefore Ho cannot be rejected at POST-TEST.

APPENDIX 26

PRE-TEST RAW SCORES:VDU

Year	stage1	stage2	stage3	stage4	stage5
1	8	4	2	7	3
2	0	3	1	4	2
3	0	19	18	20	17

POST-TEST RAW SCORES:VDU

Year	stage1	stage2	stage3	stage4	stage5
1	8	3	3	2	0
2	9	1	1	1	1
3	8	3	5	5	6

RANKED PRE-TEST SCORES:VDU

Year	stage1	stage2	stage3	stage4	stage5
1	5	3	1	4	2
2	1	4	2	5	3
3	1	4	3	5	2

RANKED POST-TEST SCORES:VDU

Year	stage1	stage2	stage3	stage4	stage5
1	5	3.5	3.5	2	1
2	5	2.5	2.5	2.5	2.5
3	5	1	2.5	2.5	4

APPENDIX 27

STUDY 2A CATEGORY PERCENTAGE

							APPENDIX[27]
COMMAND 1							
CONTROL 2							
IN/OUTPUT 3							
SUNDRY 4							
WORD/YEAR	IST	2ND	3RD	L4	L5	U5	
INPUT	3	3	3	3	3	3	
	1/2/4	1/2/4	1/2/4	1/2/4	1/2/4	1	
						2/4	
PRINT	1	1	1	1	1	1	
	2	2	3	2	3	3	
	3/4	3/4	2	3	2	2/4	
			4	4	4		
PROGRAM	2	2	4	2	2	4	
	1	4	2	4	4	2	
	4	1	1	1	1	3	
	3	3	3	3	3	1	
LINE NO	2	4	4	4	4	4	
	3	2	3	2	2/3	2	
	4	1	1	3		1/3	
	1	3	2	1	1		
RUN	1	1	1	1	1	1	
	2	2	2	2	2	3	
	3	4	3/4	3	3	2/4	
	4	3		4	4		
LIST	1	1	1	1	1	1	
	2/3	2	2	2	3	2	
		3/4	3	3	2/4	3	
	4		4	4		4	
CHAIN	3	2	1	1	1/3	1	
	1	1	2	2		2/4	
	2/4	3/4	3	4	2/4		
			4	3		3	
MOVE	1	1	1	1	1	1	
	4	2	2	2	2	2	
	2/3	3	4	3	3	3	
		4	3	4	4	4	
							APPENDIX[ ]

STUDY 2A CATEGORY PERCENTAGE

							APP(27)
WORD/YEAR	IST	2ND	3RD	L4	L5	U5	
DRAW	1	1	1	1	1	1	
	2	2	2	2	3	2	
	4	4	4	3	2	3	
	3	3	3	4	4	4	
CLS	1	1	1	1	1	1	
	2/3	2/4	2	2	2	2	
			3	3	3	4	
	4	3	4	4	4	3	
COLOUR	2	2	2	2	4	2	
	4	4	4	4	2	4	
	1	1	1/3	1	3	1	
	3	3		3	1	3	
OLD	2	4	2	4	2	2	
	3	2	3	3	1	4	
	1	3	1	2	3	1	
	4	1	4	1	4	3	
MODE	3	2	2	2	2	2	
	1	1	4	4	3/4	4	
	2	4	3	1		1	
	4	3	1	3	1	3	
REM	3	4	1	3	2/4	4	
	1	3	2	2		2	
	2	2	3	1	3	1	
	4	1	4	4	1	3	
DISK	4	4	4	4	4	4	
	2	2/3	2	3	3	3	
	3		3	2	2	2	
	1	1	1	1	1	1	
							APPENDIX[ ]
MEMORY%	1ST	2ND	3RD	L4	L5	U5	
INPUT	25	40	15	56.9	37.5	20	
PRINT	28.6	37.8	28.3	34.5	25	26.7	

STUDY 2A CATEGORY PERCENTAGE

App(27)

PROGRAM	53.6	64.4	55	69	37.5	66.7	
LINE NO.	28.6	46.7	30	50	37.5	40	
RUN	57	28.9	48.3	44.8	29.2	26.7	
LIST	64.3	37.8	55	43.1	45.8	53.3	
CHAIN	53.6	51.1	36.7	44.8	58.3	40	
MOVE	17.9	24.4	16.7	19	8.3	6.7	
DRAW	17.9	26.7	21.7	24.1	12.5	13.3	
CLS	42.9	42.2	50	41.4	29.2	6.7	
COLOUR	14.3	33.3	16.7	25.9	12.5	0	
OLD	60.7	55.6	65	56.9	70.8	66.7	
MODE	35.7	37.8	26.7	46.6	33.3	6.7	
REM	53.6	51.1	45	62.1	20.8	46.7	
DISK	60.7	64.4	36.7	50	54.2	60	

APPENDIX 28

	LANGUAGE PARTA				**APPENDIX**	(28)
	CHI X^2					
	YEAR 1 SUBJECTS 28					
	CONTROL	COMMAND	IN/OUTPUT	OTHER	WORD	**OBS/EXP**
	0	0	28	0	INPUT	observed
	0	0	28	0		expected
	1	26	0	0	PRINT	observed
	0	0	28	0		expected
	11	10	2	6	PROGRAM	observed
	0	0	0	28		expected
	15	2	6	5	LINE NO	observed
	28	0	0	0		expected
	3	23	2	0	RUN	observed
	0	28	0	0		expected
	2	23	2	1	LIST	observed
	0	0	28	0		expected
	6	7	9	6	CHAIN	observed
	0	28	0	0		expected
	2	19	2	5	MOVE	observed
	0	28	0	0		expected
	5	19	2	3	DRAW	observed
	0	0	28	0		expected
	5	17	4	1	CLS	observed
	0	28	0	0		expected
	12	5	4	7	COLOUR	observed
	0	0	28	0		expected
	12	5	7	4	OLD	observed
	0	28	0	0		expected
	5	8	11	4	MODE	observed
	0	28	0	0		expected
	7	8	10	3	REM	observed
	0	0	0	28		expected
	10	1	6	11	DISK	observed
	0	0	28	0		expected
	96	173	95	56		
	28	168	168	56		
(o-e)^2/e	165.1429	0.148809	31.7202381	0	**CHI X^2 =**	**197.011905**
					DF=3	

	LANGUAGE PARTA					**APPENDIX** (28)
	CHI X^2					
	YEAR 2 SUBJECTS 45					
	CONTROL	COMMAND	IN/OUTPUT	OTHER	WORD	**OBS/EXP**
	0	0	45	0	INPUT	observed
	0	0	45	0		expected
	2	41	1	1	PRINT	observed
	0	0	45	0		expected
	22	4	3	16	PROGRAM	observed
	0	0	0	45		expected
	11	7	3	25	LINE NO	observed
	45	0	0	0		expected
	6	38	0	2	RUN	observed
	0	45	0	0		expected
	12	28	2	2	LIST	observed
	0	0	45	0		expected
	16	13	8	8	CHAIN	observed
	0	45	0	0		expected
	12	29	3	1	MOVE	observed
	0	45	0	0		expected
	10	28	3	4	DRAW	observed
	0	0	45	0		expected
	8	21	7	8	CLS	observed
	0	45	0	0		expected
	20	9	5	11	COLOUR	observed
	0	0	45	0		expected
	2	6	15	22	OLD	observed
	0	45	0	0		expected
	17	15	5	8	MODE	observed
	0	45	0	0		expected
	9	8	12	16	REM	observed
	0	0	0	45		expected
	8	1	9	26	DISK	observed
	0	0	45	0		expected
	155	248	121	150		
	45	270	270	90		
	268.8889	1.792593	82.2259259	40	**CHI X^2 =**	**392.907407**
						DF=3

	LANGUAGE PARTA				**APPENDIX**	(28)
	CHI X^2					
	YEAR 3 SUBJECTS 60					
	CONTROL	COMMAND	IN/OUTPUT	OTHER	WORD	**OBS/EXP**
	0	0	60	0	INPUT	observed
	0	0	60	0		expected
	2	47	10	1	PRINT	observed
	0	0	60	0		expected
	20	10	4	27	PROGRAM	observed
	0	0	0	60		expected
	16	5	7	32	LINE NO	observed
	60	0	0	0		expected
	4	56	0	0	RUN	observed
	0	60	0	0		expected
	6	47	4	3	LIST	observed
	0	0	60	0		expected
	26	21	8	6	CHAIN	observed
	0	60	0	0		expected
	15	41	1	2	MOVE	observed
	0	60	0	0		expected
	8	47	1	4	DRAW	observed
	0	0	60	0		expected
	15	35	6	3	CLS	observed
	0	60	0	0		expected
	23	11	11	13	COLOUR	observed
	0	0	60	0		expected
	21	13	16	10	OLD	observed
	0	60	0	0		expected
	25	6	13	16	MODE	observed
	0	60	0	0		expected
	12	22	18	14	REM	observed
	0	0	0	60		expected
	7	0	11	37	DISK	observed
	0	0	60	0		expected
	200	361	170	168		
	60	360	360	120		
	326.6667	0.002777	100.277778	19.2	**CHI X^2 =**	**446.147222**
					DF=3	

	LANGUAGE PARTA				**APPENDIX**	(28)
	CHI X^2					
	YEAR L4 SUBJECTS 58					
	CONTROL	COMMAND	IN/OUTPUT	OTHER	WORD	**OBS/EXP**
	0	0	58	0	INPUT	observed
	0	0	58	0		expected
	3	53	2	0	PRINT	observed
	0	0	58	0		expected
	23	13	6	16	PROGRAM	observed
	0	0	0	58		expected
	19	2	4	32	LINE NO	observed
	58	0	0	0		expected
	7	46	3	2	RUN	observed
	0	58	0	0		expected
	6	47	4	1	LIST	observed
	0	0	58	0		expected
	19	20	8	11	CHAIN	observed
	0	58	0	0		expected
	10	46	2	1	MOVE	observed
	0	58	0	0		expected
	6	49	3	0	DRAW	observed
	0	0	58	0		expected
	16	25	9	8	CLS	observed
	0	58	0	0		expected
	23	12	8	14	COLOUR	observed
	0	0	58	0		expected
	11	8	12	27	OLD	observed
	0	58	0	0		expected
	24	12	9	14	MODE	observed
	0	58	0	0		expected
	15	14	16	13	REM	observed
	0	0	0	58		expected
	6	3	16	33	DISK	observed
	0	0	58	0		expected
	188	350	160	172		
	58	348	348	116		
	291.3793	0.011494	101.563218	27.03448	**CHI X^2 =**	**419.988506**
					DF=3	

	LANGUAGE PARTA				**APPENDIX**	(28)
	CHI X^2					
	YEAR L5 SUBJECTS 24					
	CONTROL	COMMAND	IN/OUTPUT	OTHER	WORD	**OBS/EXP**
	0	0	24	0	INPUT	observed
	0	0	24	0		expected
	1	20	3	0	PRINT	observed
	0	0	24	0		expected
	16	1	0	7	PROGRAM	observed
	0	0	0	24		expected
	4	0	4	17	LINE NO	observed
	0	24	0	0		expected
	2	21	1	0	RUN	observed
	24	0	0	0		expected
	0	22	2	0	LIST	observed
	0	0	24	0		expected
	3	9	9	3	CHAIN	observed
	0	24	0	0		expected
	3	19	2	0	MOVE	observed
	0	24	0	0		expected
	1	18	4	0	DRAW	observed
	0	0	24	0		expected
	6	13	3	1	CLS	observed
	0	24	0	0		expected
	8	1	6	9	COLOUR	observed
	0	0	24	0		expected
	8	7	6	3	OLD	observed
	0	24	0	0		expected
	7	5	6	6	MODE	observed
	0	24	0	0		expected
	10	0	4	10	REM	observed
	0	0	0	24		expected
	5	1	6	12	DISK	observed
	0	0	24	0		expected
	74	137	80	68		
	24	144	144	48		
	104.1667	0.340277	28.4444444	8.333333	**CHI X^2 =**	**141.284722**
					DF=3	

	LANGUAGE PARTA				**APPENDIX**	(28)
	CHI X^2					
	YEAR U5 SUBJECTS 14					
	CONTROL	COMMAND	IN/OUTPUT	OTHER	WORD	**OBS/EXP**
	1	2	11	0	INPUT	observed
	0	0	14	0		expected
	0	12	3	0	PRINT	observed
	0	0	14	0		expected
	4	2	3	6	PROGRAM	observed
	0	0	0	14		expected
	5	0	0	12	LINE NO	observed
	0	14	0	0		expected
	0	13	2	0	RUN	observed
	14	0	0	0		expected
	3	10	2	0	LIST	observed
	0	0	14	0		expected
	3	8	1	3	CHAIN	observed
	0	14	0	0		expected
	3	10	2	0	MOVE	observed
	0	14	0	0		expected
	4	8	3	0	DRAW	observed
	0	0	14	0		expected
	4	8	0	3	CLS	observed
	0	14	0	0		expected
	7	2	1	5	COLOUR	observed
	0	0	14	0		expected
	6	3	2	4	OLD	observed
	0	14	0	0		expected
	9	2	1	3	MODE	observed
	0	14	0	0		expected
	3	2	1	8	REM	observed
	0	0	0	14		expected
	2	0	6	7	DISK	observed
	0	0	14	0		expected
	54	82	38	51		
	14	84	84	28		
	114.2857	0.047619	25.1904762	18.89286	**CHI X^2 =**	**158.416667**
					DF=3	

APPENDIX 29

	LANGUAGE PARTA				**APPENDIX** (29)	
	Friedman Two-way Analysis of Variance				**RANK ORDER**	
	YEAR	**CATEGORY**				
		CONTROL	COMMAND	I/OUTPUT	OTHER	
	1	1	2	4	3	
	2	1	3	4	2	
	3	1	3	4	2	
	L4	1	3	4	2	
	L5	1	3	4	2	
	U5	1	3	4	2	
	Rj	6	17	24	13	
	Rj^2	36	289	576	169	
	N=6	K=4				
	H=	1070				
	X(r)^2=	**12xH/NK(K+1)-3N(K+1)**				
	X(r)^2=	17				
	CRITICAL VALUE FOR p< 0.01 = 10.50					
	OBSERVED VALUE = 17					

	LANGUAGE PARTA				**APPENDIX** (29)	
	Friedman Two-way Analysis of Variance				neg/pos table	
	year 1					
	Control	Command	I/out	Other		
	0	0	0	0		
	1	26	-28	0		
	11	10	2	-22		
	-13	2	6	5		
	3	-5	2	0		
	2	23	-26	1		
	6	-21	9	6		
	2	-9	2	5		
	5	19	-26	3		
	5	-11	4	1	**YEAR 1**	**RANKED**
	12	5	-24	7	1	CONTROL
	12	-23	7	4	2	COMMAND
	5	-20	11	4	3	OTHER
	7	8	10	-25	4	IN/OUTPUT
	10	1	-22	11		
	68	**5**	**-73**	**0**		
	LANGUAGE PARTA					
	Friedman Two-way Analysis of Variance				neg/pos table	
	year 2					
	Control	Command	I/out	Other		
	0	0	0	0		
	2	41	-44	1		
	22	4	3	-29		
	-34	7	3	25		
	6	-7	0	2		
	12	28	-43	2		
	16	-32	8	8		
	12	-16	3	1		
	10	28	-42	4		
	8	-24	7	8	**YEAR 2**	**RANKED**
	20	9	-40	11	1	CONTROL
	2	-39	15	22	2	OTHER
	17	-30	5	8	3	COMMAND
	9	8	12	-29	4	IN/OUTPUT
	8	1	-36	26		
	110	**-22**	**-149**	**60**		

	LANGUAGE PARTA					**APPENDIX(29)**
	Friedman Two-way Analysis of Variance				neg/pos table	
	year 3					
	Control	Command	I/out	Other		
	0	0	0	0		
	2	47	-50	0		
	20	10	4	-33		
	-44	5	7	32		
	4	-4	0	0		
	6	47	-56	3		
	26	-39	8	6		
	15	-19	1	2		
	8	47	-59	4		
	15	-25	6	3	**YEAR 3**	**RANKED**
	23	11	-49	13	1	CONTROL
	21	-47	16	10	2	OTHER
	25	-54	13	16	3	COMMAND
	12	22	18	-26	4	IN/OUTPUT
	7	0	-49	37		
	140	1	-190	67		
	LANGUAGE PARTA					
	Friedman Two-way Analysis of Variance				neg/pos table	
	year L4					
	Control	Command	I/out	Other		
	0	0	0	0		
	3	53	-56	0		
	23	13	6	-42		
	-39	2	4	32		
	7	-12	3	2		
	6	47	-54	1		
	19	-38	8	11		
	10	-12	2	1		
	6	49	-55	0		
	16	-33	9	8	**YEAR L4**	**RANKED**
	23	12	-50	14	1	CONTROL
	11	-50	12	27	2	OTHER
	24	-46	9	14	3	COMMAND
	15	14	16	-45	4	IN/OUTPUT
	6	3	-42	33		
	130	**2**	**-188**	**56**		

	LANGUAGE PARTA				**APPENDIX**	(29)
	Friedman Two-way Analysis of Variance				neg/pos table	
	year L5					
	Control	Command	I/out	Other		
	0	0	0	0		
	1	20	-21	0		
	16	1	0	-17		
	4	-24	4	17		
	-22	21	1	0		
	0	22	-22	0		
	3	-15	9	3		
	3	-5	2	0		
	1	18	-20	0		
	6	-11	3	1	**YEAR L5**	**RANKED**
	8	1	-18	9	1	CONTROL
	8	-17	6	3	2	OTHER
	7	-19	6	6	3	COMMAND
	10	0	4	-14	4	IN/OUTPUT
	5	1	-18	12		
	50	**-7**	**-64**	**20**		
	LANGUAGE PARTA					
	Friedman Two-way Analysis of Variance				neg/pos table	
	year U5					
	Control	Command	I/out	Other		
	1	2	-3	0		
	0	12	-11	0		
	4	2	3	-8		
	5	-14	0	12		
	-14	13	2	0		
	3	10	-12	0		
	3	-6	1	3		
	3	-4	2	0		
	4	8	-11	0		
	4	-6	0	3	**YEAR U5**	**RANKED**
	7	2	-13	5	1	CONTROL
	6	-11	2	4	2	OTHER
	9	-12	1	3	3	COMMAND
	3	2	1	-6	4	IN/OUTPUT
	2	0	-8	7		
	40	**-2**	**-46**	**23**		

APPENDIX 30

MEMORY VARIANCE :

APPENDIX[30]

2A MEMORY VARIANCE	X(1)	X(1)^2	X(2)	X(2)^2	X(3)	X(3)^2	X(4)	X(4)^2	X(5)	X(5)^2	X(6)	X(6)^2
INPUT	7	49	18	324	9	81	33	1089	9	81	3	9
PRINT	8	64	17	289	17	289	20	400	6	36	4	16
PROGRAM	15	225	29	841	33	1089	40	1600	9	81	10	100
LINE NO	8	64	21	441	18	324	29	841	9	81	6	36
RUN	16	256	13	169	29	841	26	676	7	49	4	16
LIST	18	324	17	289	33	1089	25	625	11	121	8	64
CHAIN	15	225	23	529	22	484	26	676	14	196	6	36
MOVE	5	25	11	121	10	100	11	121	2	4	1	1
DRAW	5	25	12	144	13	169	14	196	3	9	2	4
CLS	12	144	19	361	30	900	24	576	7	49	1	1
COLOUR	4	16	15	225	10	100	15	225	3	9	0	0
OLD	17	289	25	625	39	1521	33	1089	17	289	10	100
MODE	10	100	17	289	16	256	27	729	8	64	1	1
REM	15	225	23	529	27	729	36	1296	5	25	7	49
DISK	17	289	29	841	22	484	29	841	13	169	9	81
SUM	172	2320	289	6017	328	8456	388	10980	123	1263	72	514
MEAN	11.467	154.6666667	19.267	401.13333	21.867	563.7	25.867	732	8.2	84.2	4.8	34.27

MEMORY VARIANCE :

2B MEMORY VARIANCE							APPENDIX[30]	
MEMORY	X(1)	X(1)^2	X(2)	X(2)^2	X(3)	X(3)^2	X(4)	X(4)^2
	YR 1	YR 1	YR 2	YR 2	YR 3	YR 3	YR L4	YR 4
NEW	15	225	14	196	20	400	30	900
10	3	9	13	169	15	225	5	25
REM	13	169	14	196	32	1024	22	484
DRAWING	1	1	2	4	3	9	2	4
MODE	4	16	4	16	8	64	6	36
2	1	1	6	36	13	169	5	25
GCOL	2	4	6	36	8	64	6	36
0 1	3	9	7	49	7	49	5	25
COMMA	0	0	4	16	0	0	3	9
PRINT	3	9	0	0	3	9	2	4
INV COMM	2	4	6	36	2	4	2	4
FRED	1	1	4	16	7	49	14	196
MOVE	0	0	0	0	2	4	1	1
200 200	3	9	4	16	3	9	5	25
DRAW	1	1	1	1	1	1	3	9
RUN	2	4	5	25	4	16	6	36
	54	462	90	812	128	2096	117	1819
N=64							DFSStot	63
K=4							DFSSbe	3
GT =(54+90+128+117)^2/N			GT=	2364.39			DFSSwi	60
SStotal=(462+812+2096+1819)-GT								
			SStota	2824.61				
SSbetween=(54^2/16+90^2/16+128^2/16+117^2/16)-GT								
			SSbetw	203.672				
SSWITH	SStotal-SSbetween							
	2621		VARbetween =		67.891			
			vARwithin =		43.682	F=	1.554	

MEMORY VARIANCE :

									APPENDIX[30]
N=90						DFSStotal		89	
K=6						DFSSbetween		5	
			GT=	20915.4		DFSSwithin		84	
SStotal=(2320+6017+8456+10980+1263+514)-GT									
			SStotal=	8634.62					
SSbetween=(172^2/16+289^2/16+328^2/16+388^2/16+123^2/16+72^2/16)-GT									
		SSbetw=		5187.69					
SSWITHIN	SStotal-SSbetween								
	3447		SStotal	8634.6222					
			SSbetw	5187.6889					
VARbetween=		1037.537778							
VARwithin=		41.03492063							
****F=	25.28	*****							
SCHEFFE S TEST		I=S(SUARE ROOT)(VARwithin)(Wg)							
Wg=1/15+ 1/15+ 1/15 =			0.2	AT 5% S=	3.3912	: AT 1% S=		4	
VARwithin=		41.03492063		I =	9.7150	FOR 5%			
				I =	11.459	FOR 1%			

STUDY -TWO(A) MEMORY/CONTROL %

							APPENDIX[3d]
	MEMORY %						
WORD	NUMBER	1ST	2ND	3RD	L4	L5	U5
INPUT	1	25	40	15	56.9	37.5	20
PRINT	2	28.6	37.8	28.3	34.5	25	26.7
PROGRAM	3	53.6	64.4	55	69	37.5	66.7
LINE NO	4	28.6	46.7	30	50	37.5	40
FOR	5	-	37.8	13.3	20.7	33.3	0
NEXT	6	-	28.9	26.7	32.8	25	0
RUN	7	57	28.9	48.3	44.8	29.2	26.7
*CAT	8	-	62.2	60	48.3	41.7	46.7
LIST	9	64.3	37.8	55	43.1	45.8	53.3
SAVE	10	-	77.8	86.7	84.5	70.8	73.3
NEW	11	-	73.3	60	60.3	58.3	26.7
CHAIN	12	53.6	51.1	36.7	44.8	58.3	40
MOVE	13	17.9	24.4	16.7	19	8.3	6.7
DRAW	14	17.9	26.7	21.7	24.1	12.5	13.3
CLS	15	42.9	42.2	50	41.4	29.2	6.7
COLOUR	16	14.3	33.3	16.7	25.9	12.5	0
OLD	17	60.7	55.6	65	56.9	70.8	66.7
MODE	18	35.7	37.8	26.7	46.6	33.3	6.7
REM	19	53.6	51.1	45	62.1	20.8	46.7
DISC	20	60.7	64.4	36.7	50	54.2	60
	CONTROL %						
WORD	NUMBER	1ST	2ND	3RD	L4	L5	U5
INPUT	1	0	0	0	0	0	6.7
PRINT	2	3.6	4.4	3.3	5.2	4.2	0
PROGRAM	3	39.3	48.9	33.3	39.7	66.7	26.7
LINE NO.	4	53.6	22.2	6.7	32.8	16.7	33.3
FOR	5	-	31.1	8.3	19	25	40
NEXT	6	-	40	10	29.3	41.7	40
RUN	7	10.7	13.3	11.7	12.1	8.3	0
*CAT	8	-	33.3	13.3	27.6	29.2	26.7
LIST	9	7.1	26.7	15	10.3	0	20
SAVE	10	-	20	16.7	22.4	4.2	13.3
NEW	11	-	35.6	18.3	31	25	26.7
CHAIN	12	21.4	35.6	20	32.8	12.5	20
MOVE	13	7.1	26.7	21.7	17.2	12.5	20
DRAW	14	17.9	22.2	23.3	10.3	4.2	26.7
CLS	15	17.9	17.8	25	27.6	25	26.7
COLOUR	16	42.9	44.4	26.7	39.7	33.3	46.7
OLD	17	42.9	4.4	28.3	19	33.3	40
MODE	18	17.9	37.8	30	41.4	29.2	60
REM	19	25	20	31.7	25.9	41.7	20
DISC	20	35.7	20	33.3	10.3	20.8	13.3

STUDY-TWO(A) COMMAND/IN-OUTPUT %

APPENDIX[30]

WORD	COMMAND NUMBER	% 1ST	2ND	3RD	L4	L5	U5
INPUT	1	0	0	0	0	0	13.3
PRINT	2	92.9	91.1	78.3	91.4	83.3	80
PROGRAM	3	32.1	8.9	16.7	22.4	4.2	13.3
LINE NO	4	7.1	15.6	8.3	3.4	0	0
FOR	5	-	35.6	23.3	37.9	37.5	26.7
NEXT	6	-	33.3	28.3	44.8	45.8	33.3
RUN	7	82	82.2	93.3	79.3	87.5	86.7
*CAT	8	-	22.2	13.3	36.2	29.2	33.3
LIST	9	82	64.4	78.3	81	91.7	66.7
SAVE	10	-	35.6	70	63.8	91.7	80
NEW	11	-	17.8	23.3	29.3	41.7	26.7
CHAIN	12	25	28.9	35	34.5	37.5	53.3
MOVE	13	64.3	64.4	68.3	79.3	79.2	66.7
DRAW	14	67.9	62.2	78.3	84.5	75	53.3
CLS	15	60.7	48.9	58.3	43.1	54.2	53.3
COLOUR	16	17.9	20	18.3	20.7	4.2	13.3
OLD	17	17.9	13.3	21.7	13.8	29.2	20
MODE	18	28.6	33.3	10	20.7	20.8	13.3
REM	19	28.6	17.8	36.7	24.1	0	13.3
DISC	20	3.6	2.2	0	5.2	4.2	0

WORD	IN/OUTPUT% NUMBER	1ST	2ND	3RD	L4	L5	U5
INPUT	1	100	100	100	100	100	73.3
PRINT	2	0	2.2	16.7	3.4	12.5	20
PROGRAM	3	7.1	6.7	6.7	10.3	0	20
LINE NO	4	21.4	6.7	11.7	6.9	16.7	0
FOR	5	-	11.1	20	12.1	8.3	0
NEXT	6	-	13.3	21.7	10.3	4.2	6.7
RUN	7	7.1	0	0	5.2	4.2	13.3
*CAT	8	-	13.3	23.3	17.2	20.8	26.7
LIST	9	7.1	4.4	6.7	6.9	8.3	13.3
SAVE	10	-	33.3	13.3	12.1	4.2	6.7
NEW	11	-	22.2	23.3	25.9	16.7	0
CHAIN	12	32	17.8	13.3	13.8	37.5	6.7
MOVE	13	7.1	6.7	1.7	3.4	8.3	13.3
DRAW	14	7.1	6.7	1.7	5.2	16.7	20
CLS	15	17.9	15.6	10	15.5	12.5	0
COLOUR	16	14.3	11.1	18.3	13.8	25	6.7
OLD	17	25	33.3	26.7	20.7	25	13.3
MODE	18	39.3	11.1	21.7	15.5	25	6.7
REM	19	35.7	26.7	30	27.6	16.7	6.7
DISC	20	21.4	20	18.3	27.6	25	40

STUDY-TWO(A) SUNDRY %

						APPENDIX[30]	
	SUNDRY%						
WORD	NUMBER	1ST	2ND	3RD	L4	L5	U5
INPUT	1	0	0	0	0	0	0
PRINT	2	0	2.2	0	0	0	0
PROGRAM	3	21.4	35.6	45	27.6	29.2	40
LINE NO	4	17.9	55.6	53.3	55.2	70.8	80
FOR	5	-	22.2	25	31	29.2	33.3
NEXT	6	-	13.3	18.3	15.5	4.2	20
RUN	7	0	4.4	0	3.4	0	0
*CAT	8	-	31.1	23.3	19	25	13.3
LIST	9	3.6	4.4	5	1.7	0	0
SAVE	10	-	11	1.7	1.7	4.2	0
NEW	11	-	24.4	21.7	13.8	16.7	46.7
CHAIN	12	21.4	17.8	10	19	12.5	20
MOVE	13	17.9	2.2	3.3	1.7	0	0
DRAW	14	10.7	8.9	6.7	0	0	0
CLS	15	3.6	17.8	5	13.8	4.2	20
COLOUR	16	25	24.4	21.7	24.1	37.5	33.3
OLD	17	14.3	48.9	16.7	46.6	12.5	26.7
MODE	18	14.3	17.8	26.7	24.1	25	20
REM	19	10.7	35.6	23.3	22.4	41.7	53.3
DISC	20	39.3	57.8	61.7	56.9	50	46.7

STUDY 2A WORD CATEGORIES

							APPENDIX[3d]
Table[ ]		CONTROL	*	*	*	*	*
WORD	**NUMBER**	**U5**	**L5**	**L4**	**3rd**	**2nd**	**1st**
INPUT	1	1	0	0	0	0	0
PRINT	2	0	1	3	2	2	1
PROGRAM	3	4	16	23	20	22	11
LINE NO.	4	5	4	19	16	11	15
FOR	5	6	6	11	18	14	n/a
NEXT	6	6	10	17	20	18	n/a
RUN	7	0	2	7	4	6	3
*CAT	8	4	7	16	25	15	n/a
LIST	9	3	0	6	6	12	2
SAVE	10	2	1	13	10	9	n/a
NEW	11	4	6	18	19	16	n/a
CHAIN	12	3	3	19	26	16	6
MOVE	13	3	3	10	15	12	2
DRAW	14	4	1	6	8	10	5
CLS	15	4	6	16	15	8	5
COLOUR	16	7	8	23	23	20	12
OLD	17	6	8	11	21	2	12
MODE	18	9	7	24	25	17	5
REM	19	3	10	15	12	9	7
DISC	20	2	5	6	7	8	10
	210	76	104	263	292	227	96
COMMAND	*	*	*	*	*		
U5	**L5**	**L4**	**3rd**	**2nd**	**1st**		
2	0	0	0	0	0		
12	20	53	47	41	26		
2	1	13	10	4	10		
0	0	2	5	7	2		
4	9	22	14	15	n/a		
5	11	26	17	15	n/a		
13	21	46	56	38	23		
5	7	21	8	10	n/a		
10	22	47	47	28	23		
12	22	37	42	16	n/a		
4	10	17	14	8	n/a		
8	9	20	21	13	7		
10	19	46	41	29	19		
8	18	49	47	28	19		
8	13	25	35	21	17		
2	1	12	11	9	5		
3	7	8	13	6	5		
2	5	12	6	15	8		
2	0	14	22	8	8		
0	1	3	0	1	1		
112	196	473	456	312	173		

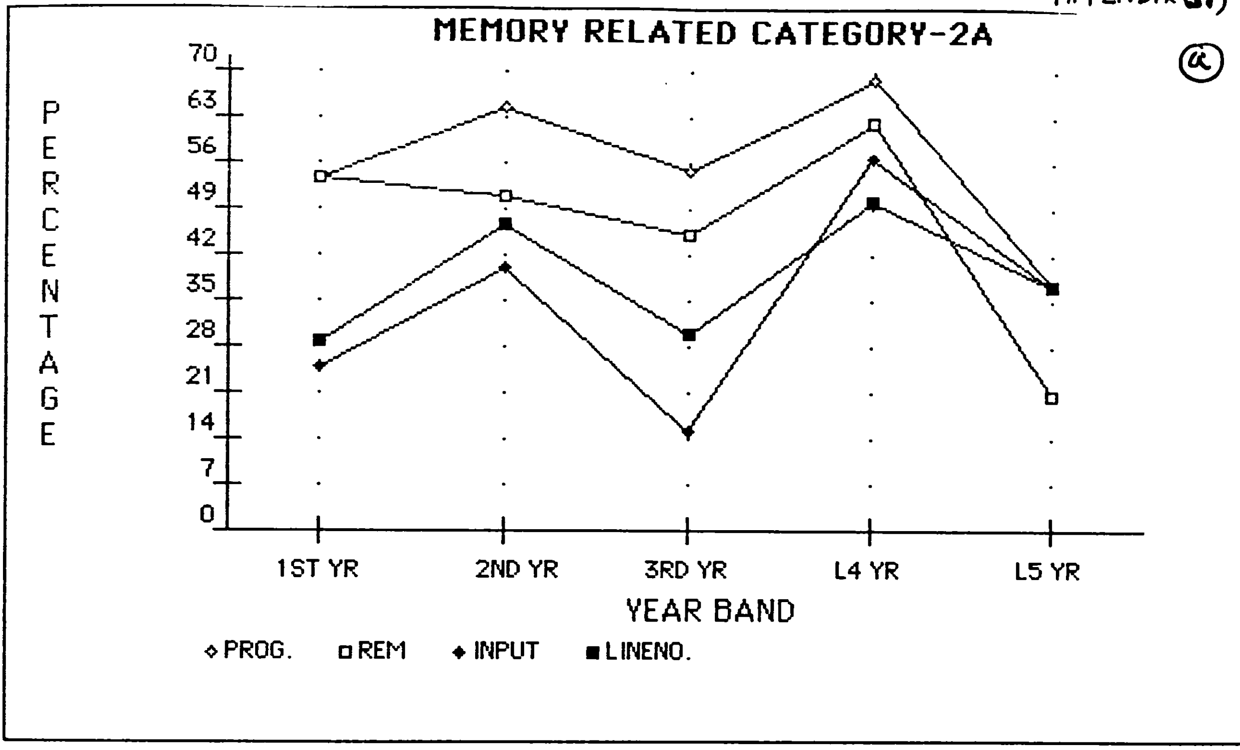
STUDY 2A WORD CATEGORIES

						APPENDIX[3a]
Table[ ]	*	I/O PUT	*	*	*	
U5	L5	L4	3rd	2nd	1st	
11	24	58	60	45	28	
3	3	2	10	1	0	
3	0	6	4	3	2	
0	4	4	7	3	6	
0	2	7	12	5	n/a	
1	1	6	13	6	n/a	
2	1	3	0	0	2	
4	5	10	14	6	n/a	
2	2	4	4	2	2	
1	1	7	8	15	n/a	
0	4	15	14	10	n/a	
1	9	8	8	8	9	
2	2	2	1	3	2	
3	4	3	1	3	2	
0	3	9	6	7	4	
1	6	8	11	5	4	
2	6	12	16	15	7	
1	6	9	13	5	11	
1	4	16	18	12	10	
6	6	16	11	9	6	
44	93	205	231	163	95	
OTHER	*	*	*	*	*	
U5	L5	L4	3rd	2nd	1st	
0	0	0	0	0	0	
0	0	0	0	1	0	
6	7	16	27	16	6	
12	17	32	32	25	5	
5	7	18	15	10	n/a	
3	1	9	11	6	n/a	
0	0	2	0	2	0	
2	6	11	14	14	n/a	
0	0	1	3	2	1	
0	1	1	1	5	n/a	
7	4	8	13	11	n/a	
3	3	11	6	8	6	
0	0	1	2	1	5	
0	0	0	4	4	3	
3	1	8	3	8	1	
5	9	14	13	11	7	
4	3	27	10	22	4	
3	6	14	16	8	4	
8	10	13	14	16	3	
7	12	33	37	26	11	
68	87	219	221	196	56	

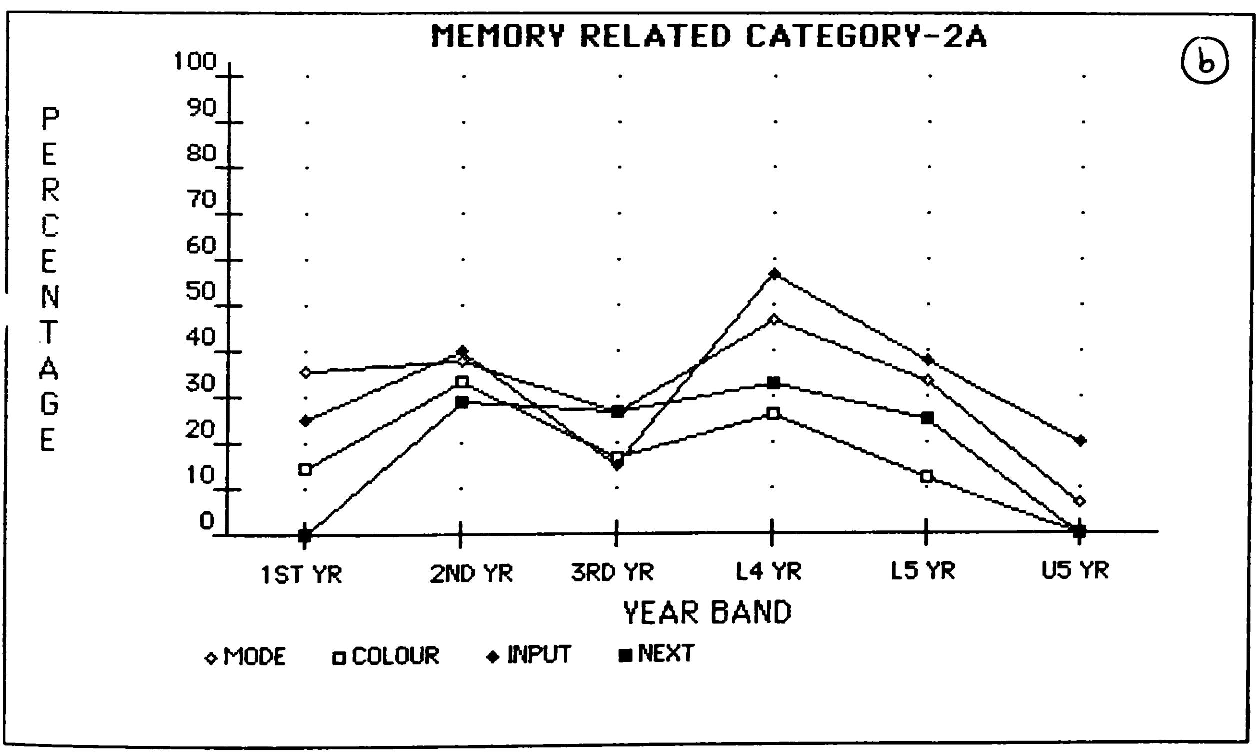
STUDY 2A WORD CATEGORIES

					APPENDIX[30]
Table[ ]	**MEMORY**	**RELATED**	*	*	*
U5	**L5**	**L4**	**3rd**	**2nd**	**1st**
3	9	33	9	18	7
4	6	20	17	17	8
10	9	40	33	29	15
6	9	29	18	21	8
0	8	12	8	17	n/a
0	6	19	16	13	n/a
4	7	26	29	13	16
7	10	28	36	28	n/a
8	11	25	33	17	18
11	17	49	52	35	n/a
4	14	35	36	33	n/a
6	14	26	22	23	15
1	2	11	10	11	5
2	3	14	13	12	5
1	7	24	30	19	12
0	3	15	10	15	4
10	17	33	39	25	17
1	8	27	16	17	10
7	5	36	27	23	15
9	13	29	22	29	17
94	178	531	476	415	172

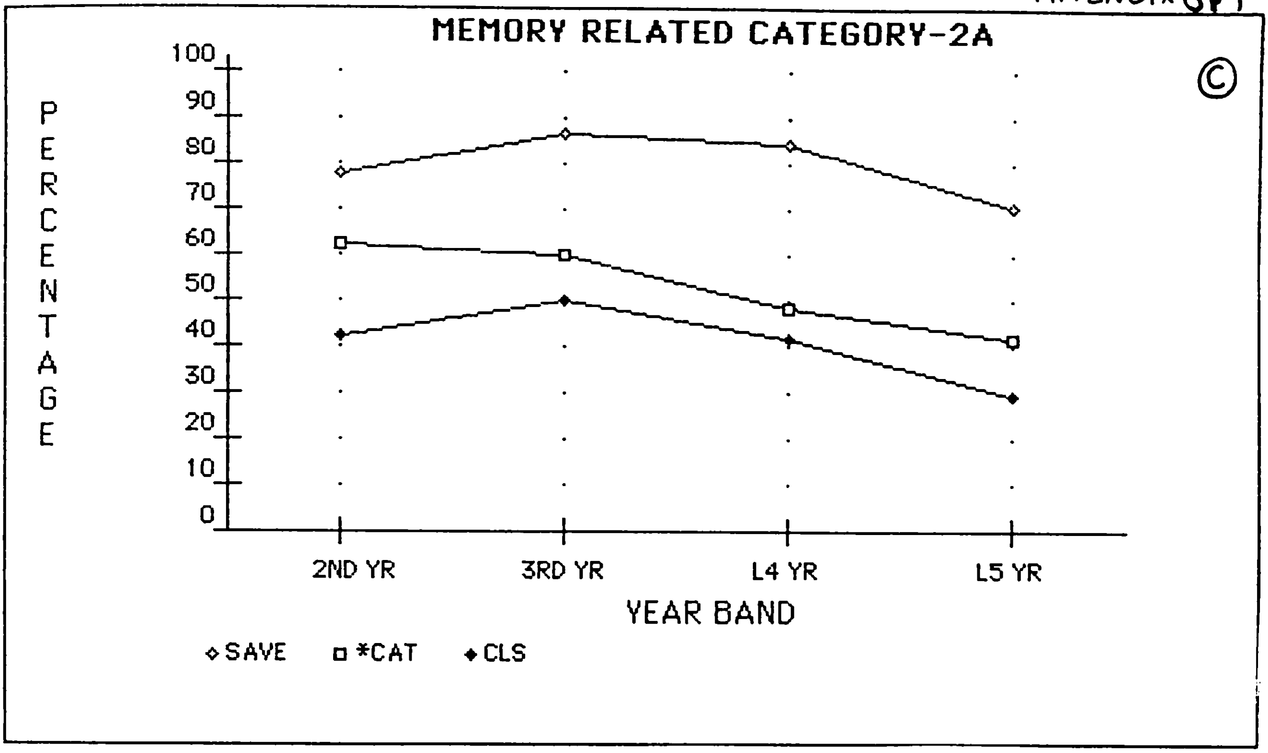
APPENDIX 31

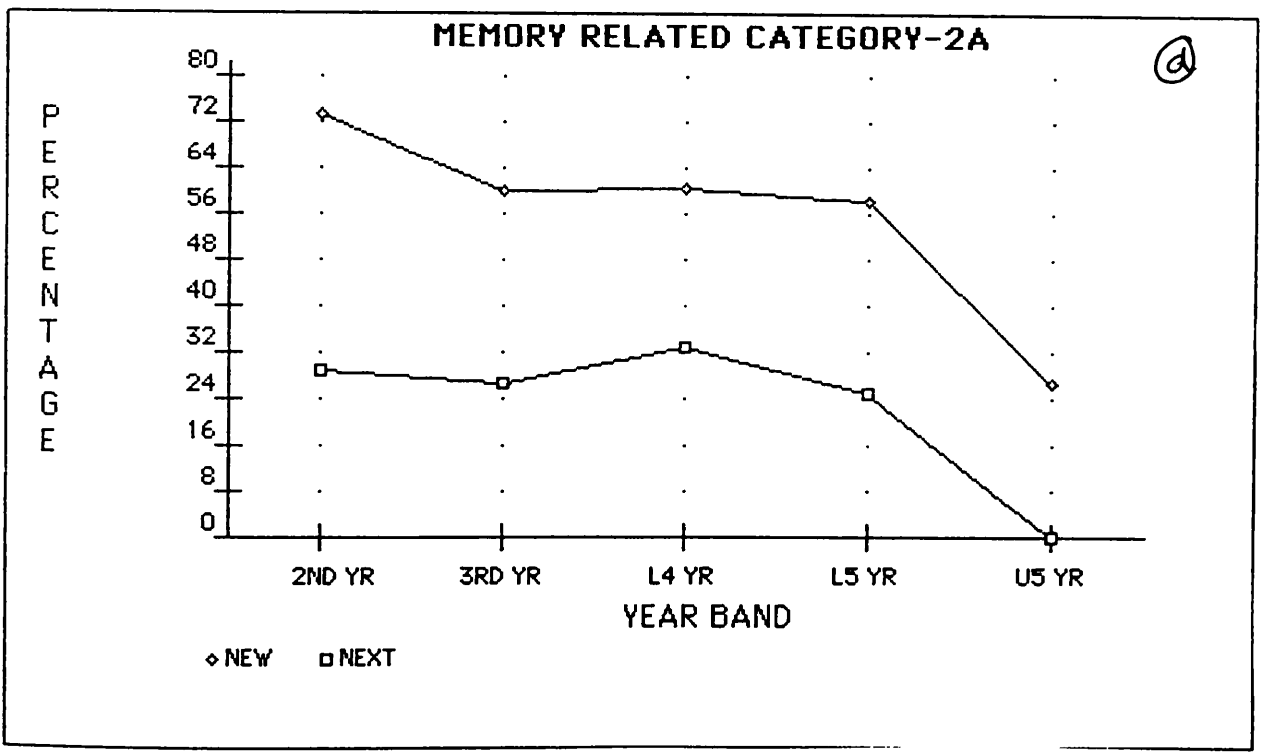


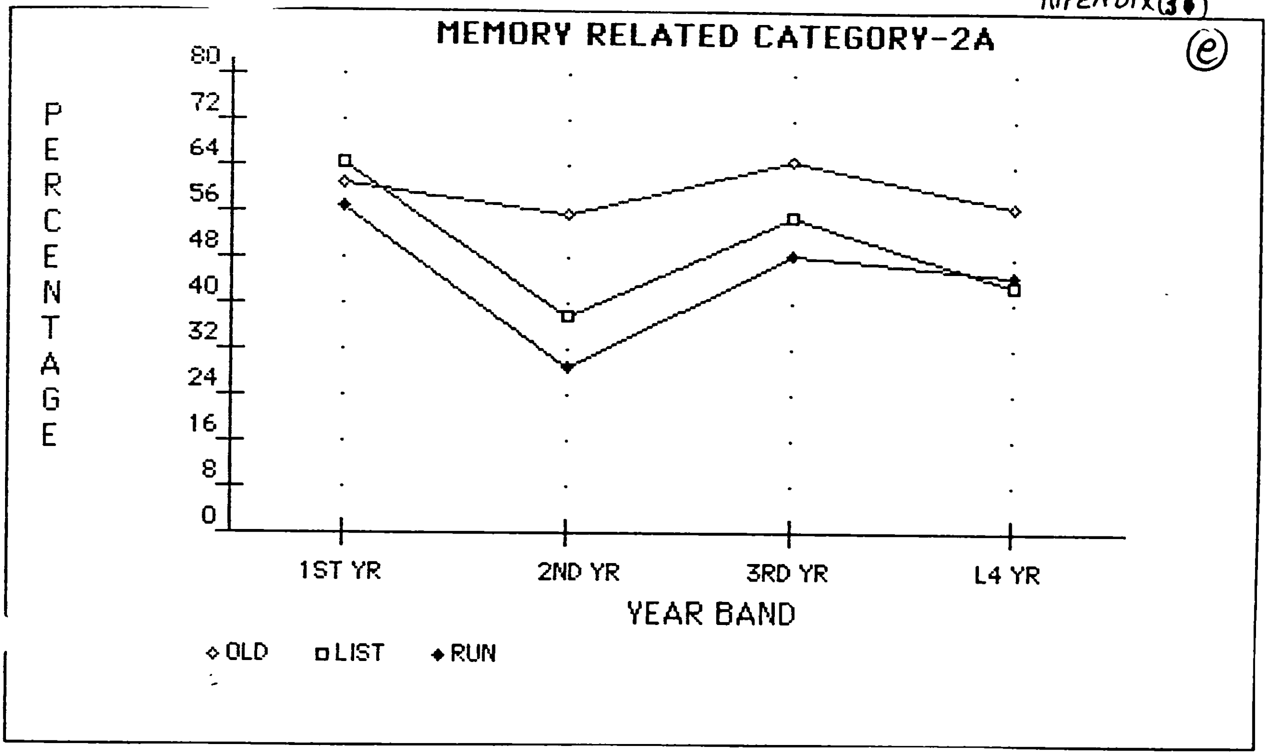
MEMORY RELATED CATEGORY-2A
(a)
PERCENTAGE
70
63
56
49
42
35
28
21
14
7
0
1ST YR
2ND YR
3RD YR
L4 YR
L5 YR
YEAR BAND
PROG.
REM
INPUT
LINENO.

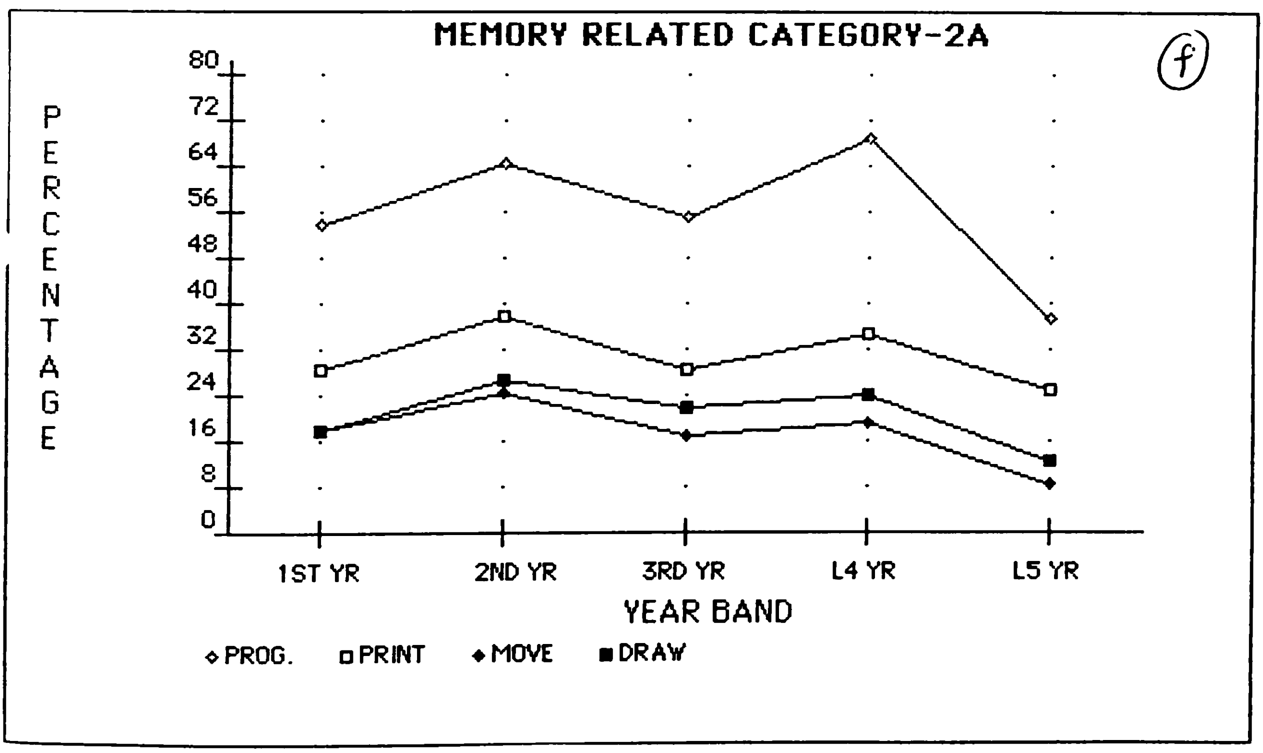


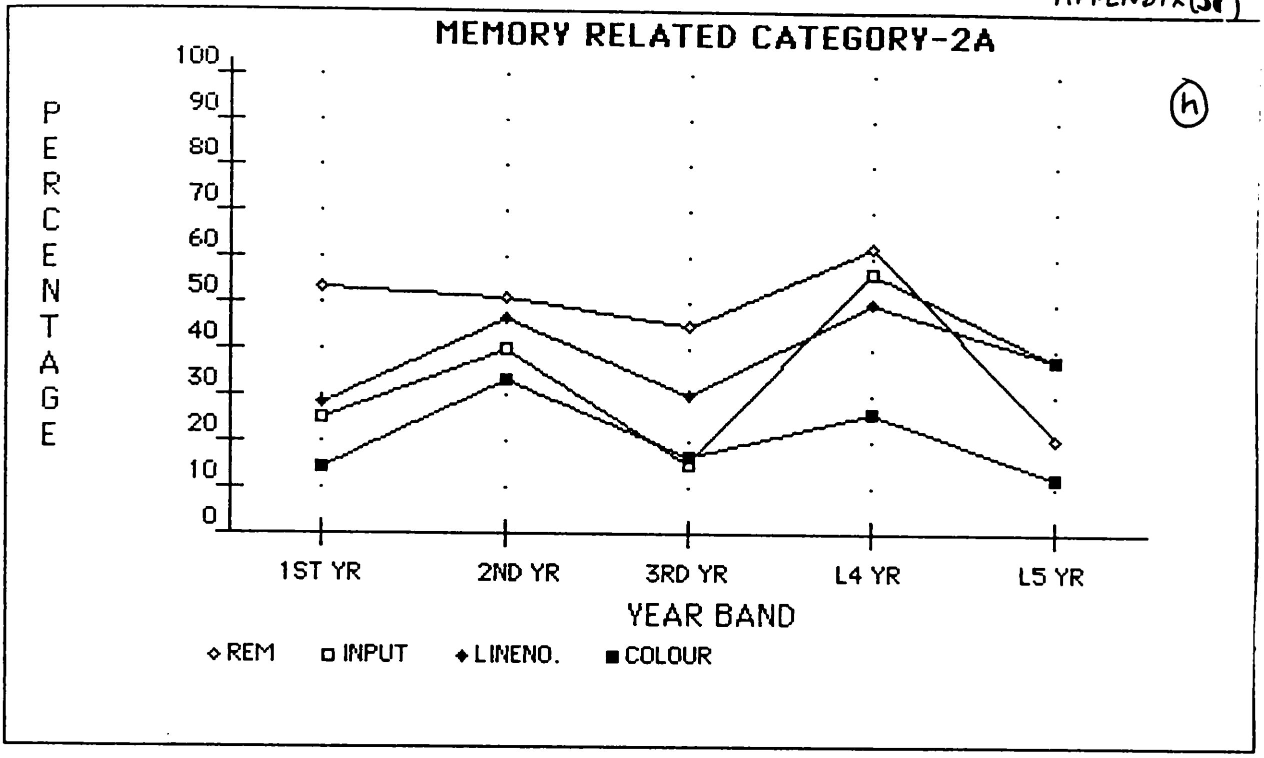
MEMORY RELATED CATEGORY-2A
(b)
PERCENTAGE
100
90
80
70
60
50
40
30
20
10
0
1ST YR
2ND YR
3RD YR
L4 YR
L5 YR
U5 YR
YEAR BAND
MODE
COLOUR
INPUT
NEXT

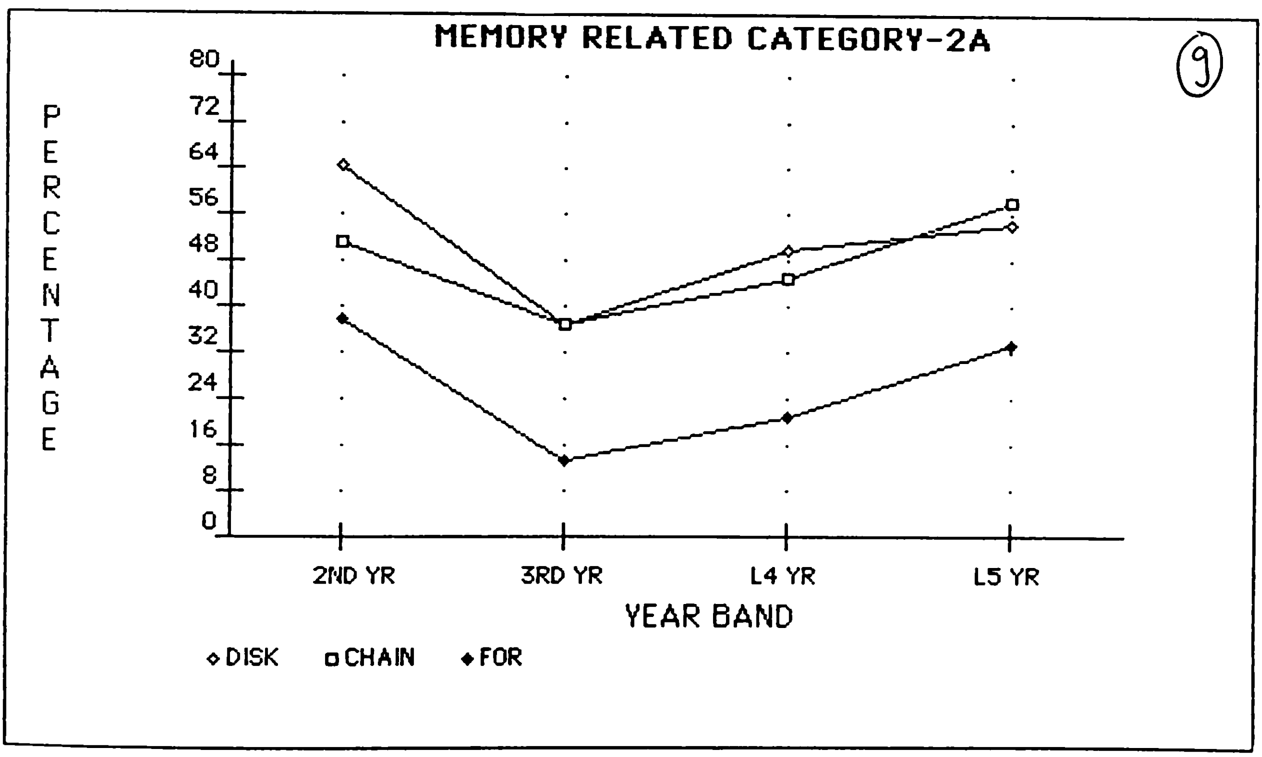

MEMORY RELATED CATEGORY-2A
©
PERCENTAGE
100
90
80
70
60
50
40
30
20
10
0
2ND YR
3RD YR
L4 YR
L5 YR
YEAR BAND
SAVE
*CAT
CLS


MEMORY RELATED CATEGORY-2A
ⓓ
PERCENTAGE
80
72
64
56
48
40
32
24
16
8
0
2ND YR
3RD YR
L4 YR
L5 YR
U5 YR
YEAR BAND
NEW
NEXT


MEMORY RELATED CATEGORY-2A
e
PERCENTAGE
80
72
64
56
48
40
32
24
16
8
0
1ST YR
2ND YR
3RD YR
L4 YR
YEAR BAND
OLD
LIST
RUN


MEMORY RELATED CATEGORY-2A
f
PERCENTAGE
80
72
64
56
48
40
32
24
16
8
0
1ST YR
2ND YR
3RD YR
L4 YR
L5 YR
YEAR BAND
PROG.
PRINT
MOVE
DRAW


MEMORY RELATED CATEGORY-2A
(h)
PERCENTAGE
100
90
80
70
60
50
40
30
20
10
0
1ST YR
2ND YR
3RD YR
L4 YR
L5 YR
YEAR BAND
REM
INPUT
LINENO.
COLOUR


MEMORY RELATED CATEGORY-2A
(g)
PERCENTAGE
80
72
64
56
48
40
32
24
16
8
0
2ND YR
3RD YR
L4 YR
L5 YR
YEAR BAND
DISK
CHAIN
FOR

APPENDIX 32

SAMPLE DATA COLLECTION SHEET LANGUAGE PART B

APPENDIX[32]

NAME.. FORM..................

1. LOOK AT THE FIRST PROGRAM BELOW:

```
NEW
10 REM DRAWING PROGRAM
20 MODE 2
30 GCOL 0,1
40 PRINT "FRED"
50 MOVE 200,200
60 DRAW 200,800
RUN
```

DO YOU KNOW WHAT IT DOES? YES / NO (ring one)

2. THE SAME PROGRAM HAS BEEN SPLIT UP SO THAT EACH NEW SECTION IS ON A NEW LINE. PUT A RING AROUND THE LABEL WHICH BEST DESCRIBES THE PROGRAM SECTION.

NEW...............	MEMORY/CONTROL/COMMAND/IN-OUT/DATA/OTHER	☐
10	MEMORY/CONTROL/COMMAND/IN-OUT/DATA/OTHER	☐
REM..................	MEMORY/CONTROL/COMMAND/IN-OUT/DATA/OTHER	☐
DRAWING..........	MEMORY/CONTROL/COMMAND/IN-OUT/DATA/OTHER	☐
MODE................	MEMORY/CONTROL/COMMAND/IN-OUT/DATA/OTHER	☐
2	MEMORY/CONTROL/COMMAND/IN-OUT/DATA/OTHER	☐
GCOL	MEMORY/CONTROL/COMMAND/IN-OUT/DATA/OTHER	☐
0 1	MEMORY/CONTROL/COMMAND/IN-OUT/DATA/OTHER	☐
, ...	MEMORY/CONTROL/COMMAND/IN-OUT/DATA/OTHER	☐
PRINT	MEMORY/CONTROL/COMMAND/IN-OUT/DATA/OTHER	☐
" "	MEMORY/CONTROL/COMMAND/IN-OUT/DATA/OTHER	☐
FRED	MEMORY/CONTROL/COMMAND/IN-OUT/DATA/OTHER	☐
MOVE	MEMORY/CONTROL/COMMAND/IN-OUT/DATA/OTHER	☐
200 200	MEMORY/CONTROL/COMMAND/IN-OUT/DATA/OTHER	☐
DRAW	MEMORY/CONTROL/COMMAND/IN-OUT/DATA/OTHER	☐
RUN	MEMORY/CONTROL/COMMAND/IN-OUT/DATA/OTHER	☐

DO NOT WRITE BELOW THIS LINE:

MEMORY																
CONTROL																
COMMAND																
INPUT/OUTPUT																
DAT																
OTHE																
Y/N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

APPENDIX 33

2B SENTENCE BREAKDOWN

						APPENDIX[33]	
SENTENCE	B/Dn						
YEAR	1	2	3	4	5	6	7
1ST	56	11	48	3.7	14.8	3.7	7.4
2ND	31.8	29.5	31.8	4.5	9.1	13.6	13.6
3RD	33.3	25	53.3	5	13.3	21.7	13.3
L4	54.5	9.1	40	3.6	10.9	9.1	10.9
YEAR	1ST	2ND	3RD	L4			
WORD							
NEW	56	31.8	33.3	54.5			
10(line no.)	11	29.5	25	9.1			
REM	48	31.8	53.3	40			
DRAWING	3.7	4.5	5	3.6			
MODE	14.8	9.1	13.3	10.9			
2(mode)	3.7	13.6	21.7	9.1			
GCOL	7.4	13.6	13.3	10.9			
0,1(colour)	11	16	12	9.1			
comma	0	9.1	0	5.5			
PRINT	11	0	5	3.6			
speech marks	7.4	14	3.3	3.6			
FRED	3.7	9.1	11.7	25.5			
MOVE	0	0	3.3	1.8			
200 200	11	9.1	5	9.1			
DRAW	3.7	2.3	1.7	5.5			
RUN	7.4	11.4	6.7	10.9			
RUN 2A	57	28.9	48.3	44.8			
PRINT 2A	28.6	37.8	28.3	34.5			
NEW 2A		73.3	60	60.3			
line no. 2A	28.6	46.7	30	50			
REM 2A	53.6	51.1	45	62.1			
MODE 2A	35.7	37.8	26.7	46.6			
MOVE 2A	17.9	24.4	16.7	19			
DRAW 2A	17.9	26.7	21.7	24.1			
COLOUR 2A	14.3	33.3	16.7	25.9			

STUDY B						Appendix[33]	
****YEAR 1****							
CATEGORY	MEMORY	CONTROL	COMMAND	I/O-PUT	DATA	OTHER	TOTAL
SENTENCE SEGMENT							
NEW	15	0	7	5	0	0	27
10(line number)	3	3	5	5	2	9	27
REM	13	2	8	1	2	1	27
DRAWING	1	7	10	1	2	6	27
MODE	4	12	6	2	1	2	27
2(mode)	1	4	6	4	1	11	27
GCOL	2	5	12	2	1	5	27
0 1 (colour)	3	3	7	6	1	7	27
,	0	8	1	5	3	10	27
PRINT	3	3	17	0	2	2	27
" "	2	5	6	1	3	10	27
FRED	1	4	6	2	5	9	27
MOVE	0	14	10	1	1	1	27
200 200	3	10	4	3	2	5	27
DRAW	1	7	17	0	1	1	27
RUN	2	3	17	3	1	1	27
****YEAR 2****							
CATEGORY	MEMORY	CONTROL	COMMAND	I/O-PUT	DATA	OTHER	TOTAL
SENTENCE SEGMENT							
NEW	14	4	10	5	9	2	44
10(line number)	13	11	6	2	6	6	44
REM	14	4	3	6	3	14	44
DRAWING	2	16	17	0	6	3	44
MODE	4	8	17	3	8	4	44
2(mode)	6	5	6	7	8	12	44
GCOL	6	13	12	4	4	5	44
0 1 (colour)	7	8	9	5	9	6	44
,	4	5	8	5	4	18	44
PRINT	0	2	37	2	0	3	44
" "	6	10	3	6	7	12	44
FRED	4	4	7	4	12	13	44
MOVE	0	16	26	1	0	1	44
200 200	4	19	8	5	5	3	44
DRAW	1	11	23	3	2	4	44
RUN	5	9	25	2	1	2	44

2B SENTENCE BREAKDOWN

STUDY B						Appendix[33]	
****YEAR 3****							
CATEGORY	MEMORY	CONTROL	COMMAND	I/O-PUT	DATA	OTHER	TOTAL
SENTENCE SEGMENT							
NEW	20	4	22	4	6	4	60
10(line number)	15	5	6	3	17	14	60
REM	32	9	5	3	7	4	60
DRAWING	3	18	26	1	5	7	60
MODE	8	13	15	5	11	8	60
2(mode)	13	6	13	5	10	13	60
GCOL	8	16	16	8	8	4	60
0 1 (colour)	7	12	10	8	13	10	60
comma	0	12	5	8	11	24	60
PRINT	3	6	45	2	2	2	60
" "	2	7	11	9	16	15	60
FRED	7	3	4	11	18	17	60
MOVE	2	18	37	0	2	1	60
200 200	3	18	12	11	11	5	60
DRAW	1	9	50	0	0	0	60
RUN	4	10	41	2	2	1	60
****YEAR L4****							
CATEGORY	MEMORY	CONTROL	COMMAND	I/O-PUT	DAT/	OTHER	TOTAL
SENTENCE SEGMENT							
NEW	30	6	10	2	6	1	55
10(line number)	5	9	8	13	13	7	55
REM	22	4	8	2	4	15	55
DRAWING	2	15	16	5	6	11	55
MODE	6	20	15	4	9	1	55
2(mode)	5	8	6	5	15	16	55
GCOL	6	14	23	5	3	4	55
0 1 (colour)	5	9	12	5	15	9	55
comma	3	13	0	4	11	24	55
PRINT	2	4	45	1	0	3	55
" "	2	17	3	6	14	13	55
FRED	14	1	4	5	21	10	55
MOVE	1	14	39	0	0	1	55
200 200	5	12	7	9	15	7	55
DRAW	3	6	42	3	1	0	55
RUN	6	7	39	1	1	1	55

APPENDIX 34

2B SENTENCE BREAKDOWN

STUDY B							A(34)
****YEAR 1***	(27 subjects)						
CATEGORY	MEMORY	CONTROL	COMMAND	I/O-PUT	DATA	OTHER	
SENTENCE SEGMENT							
NEW	15	0	7	5	0	0	observed
	27	0	0	0	0	0	expectd
10(line number)	3	3	5	5	2	9	observed
	0	27	0	0	0	0	expectd
REM	13	2	8	1	2	1	observed
	0	0	0	0	0	27	expectd
DRAWING	1	7	10	1	2	6	observed
	0	0	0	27	0	0	expectd
MODE	4	12	6	2	1	2	observed
	0	0	27	0	0	0	expectd
2(mode)	1	4	6	4	1	11	observed
	0	0	0	0	27	0	expectd
GCOL	2	5	12	2	1	5	observed
	0	0	0	27	0	0	expectd
0 1 (colour)	3	3	7	6	1	7	observed
	0	0	0	0	27	0	expectd
,	0	8	1	5	3	10	observed
	0	27	0	0	0	0	expectd
PRINT	3	3	17	0	2	2	observed
	0	0	0	27	0	0	expectd
" "	2	5	6	1	3	10	observed
	0	27	0	0	0	0	expectd
FRED	1	4	6	2	5	9	observed
	0	0	0	0	27	0	expectd
MOVE	0	14	10	1	1	1	observed
	0	0	0	0	0	27	expectd
200 200	3	10	4	3	2	5	observed
	0	0	0	0	27	0	expectd
DRAW	1	7	17	0	1	1	observed
	0	0	0	27	0	0	expectd
RUN	2	3	17	3	1	1	observed
	0	0	27	0	0	0	expectd
total observed	54	90	139	41	28	80	DF=5
total expected	85	139	112	166	139	85	χ^2
	11.3059	17.2734	6.508929	94.127	88.6	0.294	218.15

****YEAR 2***	44 subjects						
CATEGORY	MEMORY	CONTROL	COMMAND	I/O-PUT	DATA	OTHER	
SENTENCE SEGMENT							
NEW	14	4	10	5	9	2	observed
	44	0	0	0	0	0	expectd
10(line number)	13	11	6	2	6	6	observed
	0	44	0	0	0	0	expectd
REM	14	4	3	6	3	14	observed
	0	0	0	0	0	44	expectd
DRAWING	2	16	17	0	6	3	observed
	0	0	0	44	0	0	expectd
MODE	4	8	17	3	8	4	observed
	0	0	44	0	0	0	expectd
2(mode)	6	5	6	7	8	12	observed
	0	0	0	0	44	0	expectd
GCOL	6	13	12	4	4	5	observed
	0	0	0	44	0	0	expectd
0 1 (colour)	7	8	9	5	9	6	observed
	0	0	0	0	44	0	expectd
,	4	5	8	5	4	18	observed
	0	44	0	0	0	0	expectd
PRINT	0	2	37	2	0	3	observed
	0	0	0	44	0	0	expectd
" "	6	10	3	6	7	12	observed
	0	44	0	0	0	0	expectd
FRED	4	4	7	4	12	13	observed
	0	0	0	0	44	0	expectd
MOVE	0	16	26	1	0	1	observed
	0	0	0	0	0	44	expectd
200 200	4	19	8	5	5	3	observed
	0	0	0	0	44	0	expectd
DRAW	1	11	23	3	2	4	observed
	0	0	0	44	0	0	expectd
RUN	5	9	25	2	1	2	observed
	0	0	44	0	0	0	expectd
total observed	90	145	217	60	84	108	DF=5
total expected	102	190	146	234	190	102	
	1.41176	10.6579	34.52740	129.38	59.1	0.352	235.47

STUDY B						Appendix(34)	
****YEAR 3***	60 subjects						
CATEGORY	MEMORY	CONTROL	COMMAND	I/O-PUT	DATA	OTHER	
SENTENCE SEGMENT							
NEW	20	4	22	4	6	4	observed
	60	0	0	0	0	0	expected
10(line number)	15	5	6	3	17	14	observed
	0	60	0	0	0	0	expected
REM	32	9	5	3	7	4	observed
	0	0	0	0	0	60	expected
DRAWING	3	18	26	1	5	7	observed
	0	0	0	60	0	0	expected
MODE	8	13	15	5	11	8	observed
	0	0	60	0	0	0	expected
2(mode)	13	6	13	5	10	13	observed
	0	0	0	0	60	0	expected
GCOL	8	16	16	8	8	4	observed
	0	0	0	60	0	0	expected
0 1 (colour)	7	12	10	8	13	10	observed
	0	0	0	0	60	0	expected
comma	0	12	5	8	11	24	observed
	0	60	0	0	0	0	expected
PRINT	3	6	45	2	2	2	observed
	0	0	0	60	0	0	expected
" "	2	7	11	9	16	15	observed
	0	60	0	0	0	0	expected
FRED	7	3	4	11	18	17	observed
	0	0	0	0	60	0	expected
MOVE	2	18	37	0	2	1	observed
	0	0	0	0	0	60	expected
200 200	3	18	12	11	11	5	observed
	0	0	0	0	60	0	expected
DRAW	1	9	50	0	0	0	observed
	0	0	0	60	0	0	expected
RUN	4	10	41	2	2	1	observed
	0	0	60	0	0	0	expected
total observed	128	166	318	80	139	129	DF=5
total expected	118	238	178	298	238	118	X^2
	0.84745	21.7815	110.1124	159.48	41.2	1.025	334.42

							A(34)
****YEAR L4**	55 pupils						
CATEGORY	MEMORY	CONTROL	COMMAND	I/O-PUT	DAT/	OTHER	
SENTENCE SEGMENT							
NEW	30	6	10	2	6	1	observed
	55	0	0	0	0	0	expected
10(line number)	5	9	8	13	13	7	observed
	0	55	0	0	0	0	expected
REM	22	4	8	2	4	15	observed
	0	0	0	0	0	55	expected
DRAWING	2	15	16	5	6	11	observed
	0	0	0	55	0	0	expected
MODE	6	20	15	4	9	1	observed
	0	0	55	0	0	0	expected
2(mode)	5	8	6	5	15	16	observed
	0	0	0	0	55	0	expected
GCOL	6	14	23	5	3	4	observed
	0	0	0	55	0	0	expected
0 1 (colour)	5	9	12	5	15	9	observed
	0	0	0	0	55	0	expected
comma	3	13	0	4	11	24	observed
	0	55	0	0	0	0	expected
PRINT	2	4	45	1	0	3	observed
	0	0	0	55	0	0	expected
" "	2	17	3	6	14	13	observed
	0	55	0	0	0	0	expected
FRED	14	1	4	5	21	10	observed
	0	0	0	0	55	0	expected
MOVE	1	14	39	0	0	1	observed
	0	0	0	0	0	55	expected
200 200	5	12	7	9	15	7	observed
	0	0	0	0	55	0	expected
DRAW	3	6	42	3	1	0	observed
	0	0	0	55	0	0	expected
RUN	6	7	39	1	1	1	observed
	0	0	55	0	0	0	expected
total observed	117	159	277	70	134	123	DF=5
total expected	113	223	168	278	223	113	
	0.14159	18.3677	70.72024	155.63	35.5	0.884	281.26

APPENDIX 35

LANGUAGE PART B				APPENDIX(35)		
Friedman Two-way Analysis of Variance					RANKED	
YEAR	**CATEGORY**					
	MEMORY	CONTROL	COMMAND	I/OUT	DATA	OTHER
1	2	4	1	5	6	3
2	2	4	1	6	5	3
3	2	4	1	6	5	3
L4	2	4	1	6	5	3
Rj	8	16	4	23	21	12
Rj^2	64	256	16	529	441	144
H=	**1450**					
N=4	**K=6**					
X(r)^2=	**12xH/NK(K+1)-3N(K+1)**					
X(r)^2=	**19.571**					
CRITICAL VALUE FOR p<0.01 = 12.82						
OBSERVED VALUE = 19.57						

A(35)

RANKED ORDER		RANKED ORDER	
YEAR 1		YEAR 2	
1	COMMAND	1	COMMAND
2	MEMORY	2	MEMORY
3	OTHER	3	OTHER
4	CONTROL	4	CONTROL
5	IN/OUTPUT	5	DATA
6	DATA	6	IN/OUTPUT

RANKED ORDER		RANKED ORDER	
YEAR 3		YEAR L4	
1	COMMAND	1	COMMAND
2	MEMORY	2	MEMORY
3	OTHER	3	OTHER
4	CONTROL	4	CONTROL
5	DATA	5	DATA
6	IN/OUTPUT	6	IN/OUTPUT

LANGUAGE PART B — **APPENDIX(35)**

Friedman Two-way Analysis of Variance — neg/pos table

YEAR 1

Memory	Control	Command	In/output	Data	Other
-12	0	7	5	0	0
3	-24	5	5	2	9
13	2	8	1	2	-26
1	7	10	-26	2	6
4	12	-21	2	1	2
1	4	6	4	-26	11
2	5	12	-25	1	5
3	3	7	6	-26	7
0	-19	1	5	3	10
3	3	17	-27	2	2
2	-22	6	1	3	10
1	4	6	2	-22	9
0	14	10	1	1	-26
3	10	4	3	-25	5
1	7	17	-27	1	1
2	3	-10	3	1	1
27	**9**	**85**	**-67**	**-80**	**26**

YEAR 2

Memory	Control	Command	In/output	Data	Other
-30	4	10	5	9	2
13	-33	6	2	6	6
14	4	3	6	3	-30
2	16	17	-44	6	3
4	8	-27	3	8	4
6	5	6	7	-36	12
6	13	12	-40	4	5
7	8	9	5	-35	6
4	-39	8	5	4	18
0	2	37	-42	0	3
6	-34	3	6	7	12
4	4	7	4	-32	13
0	16	26	1	0	-43
4	19	8	5	-39	3
1	11	23	-41	2	4
5	9	-19	2	1	2
46	**13**	**129**	**-116**	**-92**	**20**

LANGUAGE PART B **APPENDIX(35)**

Friedman Two-way Analysis of Variance neg/pos table

YEAR 3

Memory	Control	Command	In/outp	Data	Other
-40	4	22	4	6	4
15	-55	6	3	17	14
32	9	5	3	7	-56
3	18	26	-59	5	7
8	13	-45	5	11	8
13	6	13	5	-50	13
8	16	16	-52	8	4
7	12	10	8	-47	10
0	-48	5	8	11	24
3	6	45	-58	2	2
2	-53	11	9	16	15
7	3	4	11	-42	17
2	18	37	0	2	-59
3	18	12	11	-49	5
1	9	50	-60	0	0
4	10	-19	2	2	1
68	**-14**	**198**	**-160**	**-101**	**9**

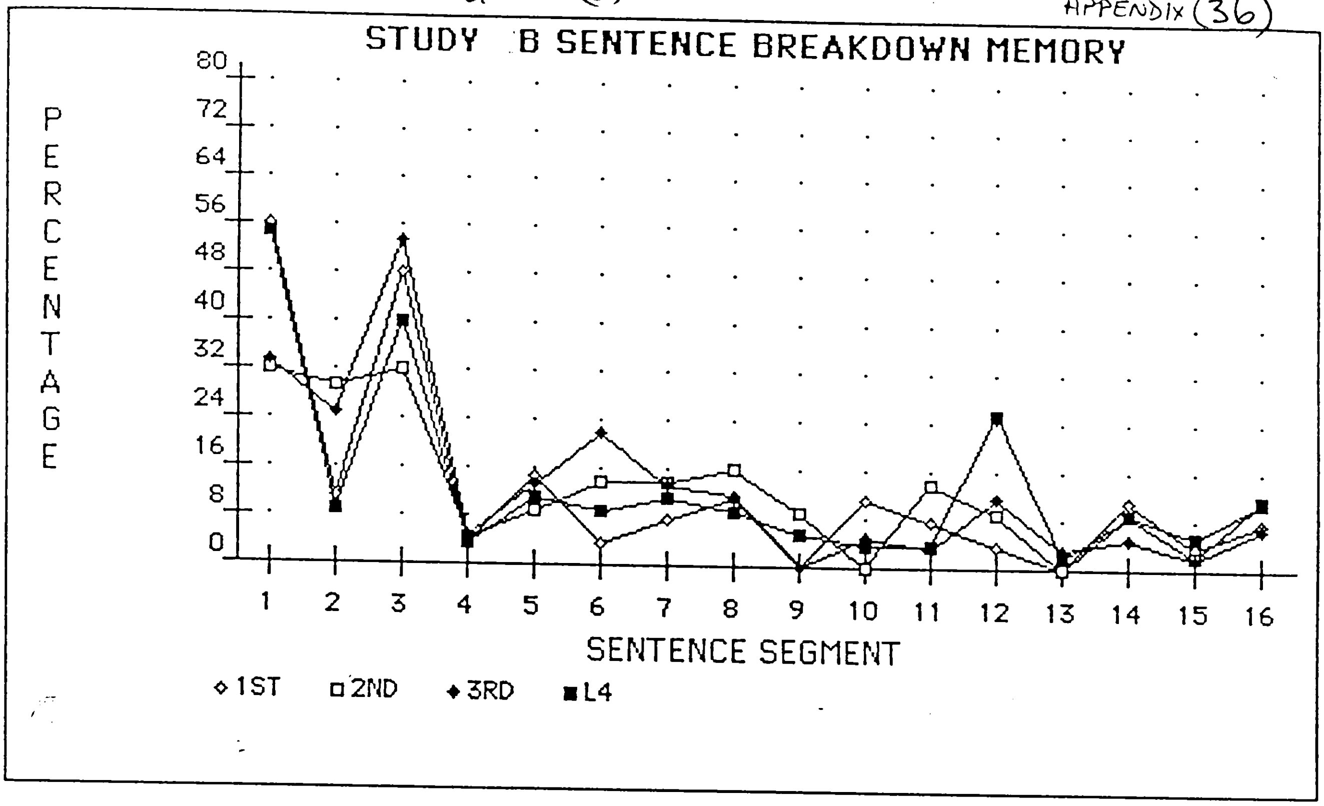
Friedman Two-way Analysis of Variance neg/pos table

YEAR L4

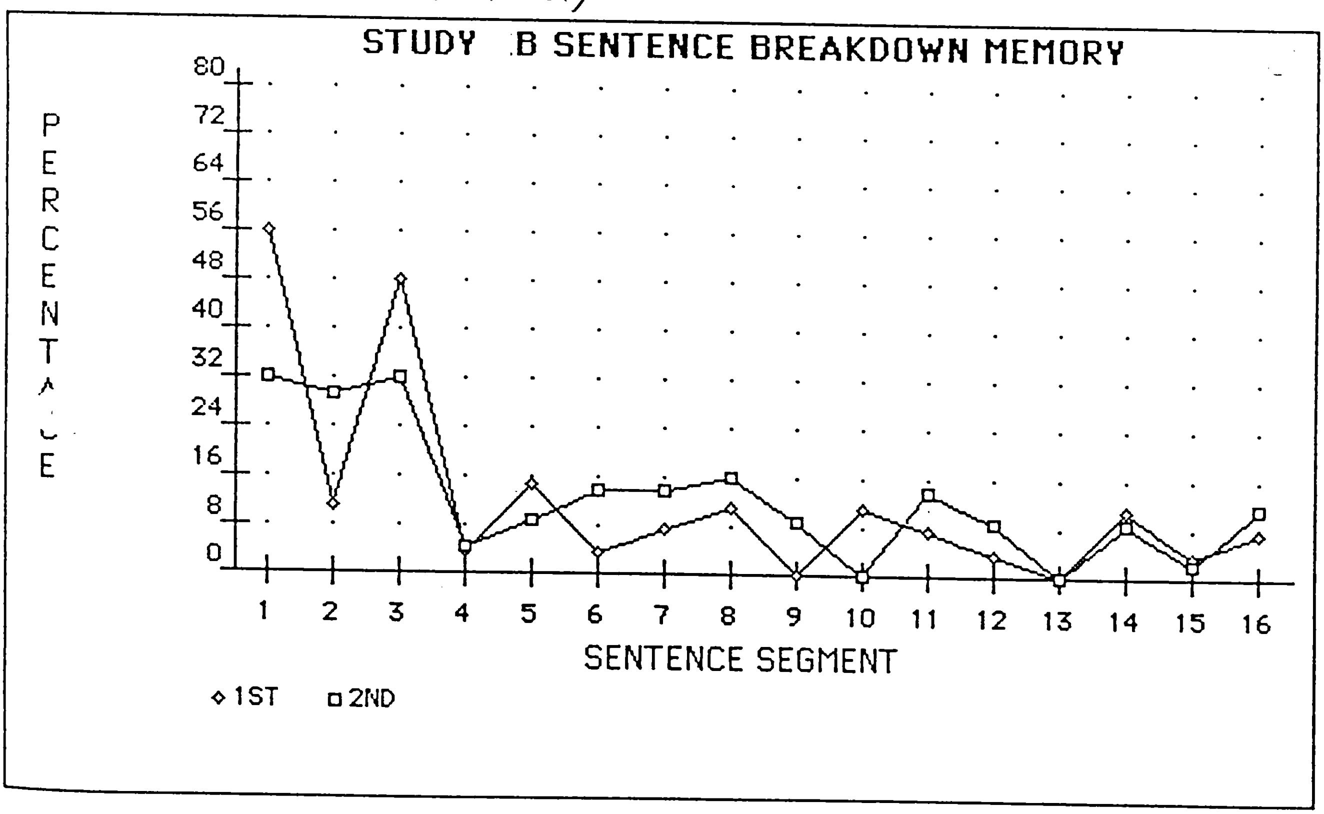
Memory	Control	Command	In/outp	Data	Other
-25	6	10	2	6	1
5	-46	8	13	13	7
22	4	8	2	4	-40
2	15	16	-50	6	11
6	20	-40	4	9	1
5	8	6	5	-40	16
6	14	23	-50	3	4
5	9	12	5	-40	9
3	-42	0	4	11	24
2	4	45	-54	0	3
2	-38	3	6	14	13
14	1	4	5	-34	10
1	14	39	0	0	-54
5	12	7		-40	7
3	6	42		1	0
6	7	-16		1	1
62	**-6**	**167**	**-108**	**-86**	**13**

APPENDIX 36

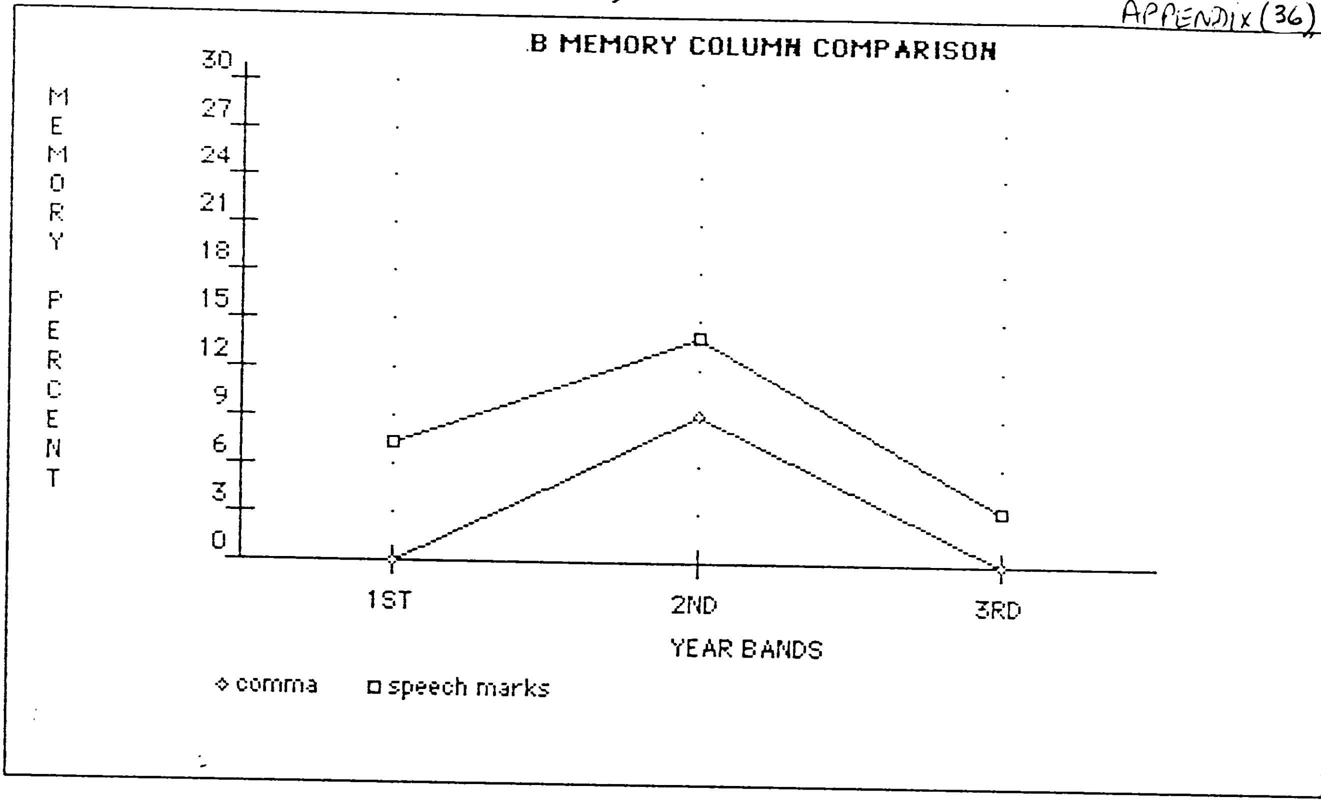
GRAPH (3)

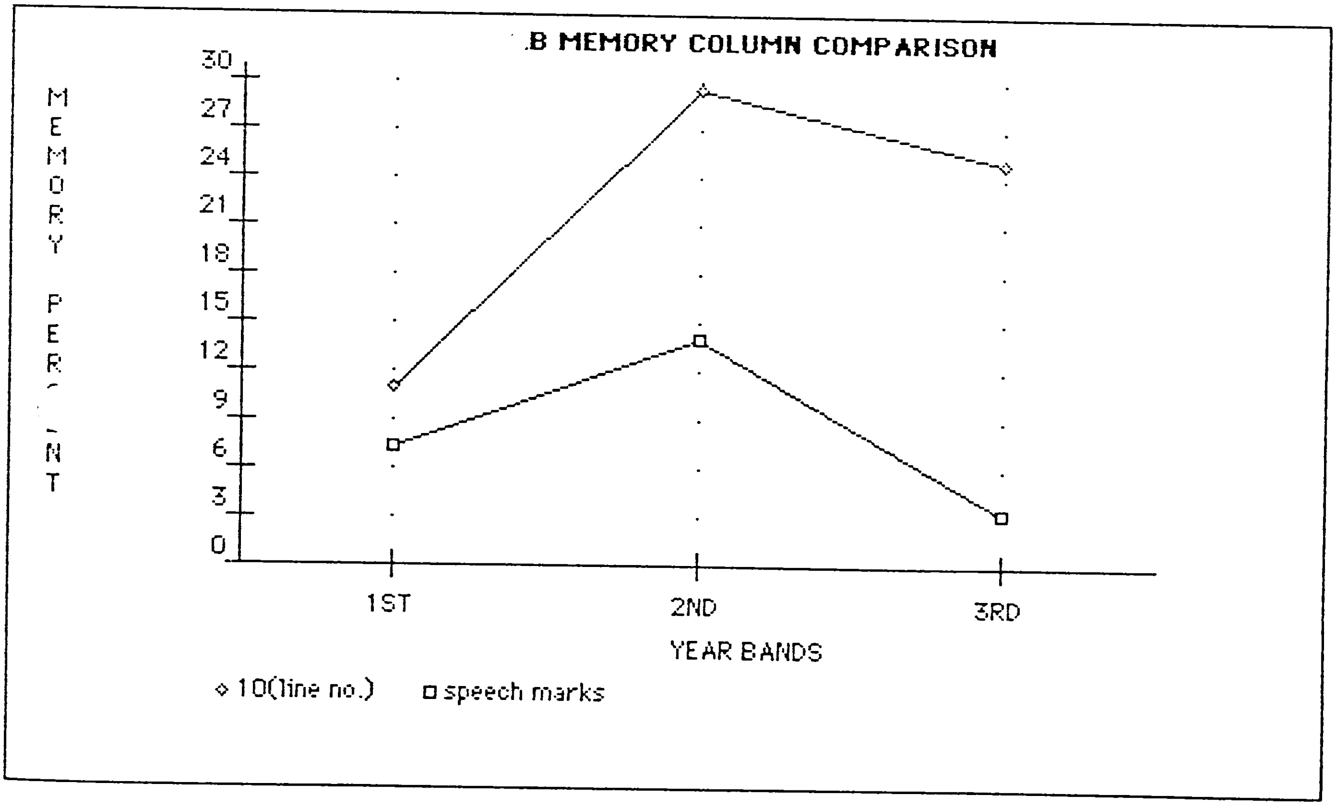


GRAPH (4)



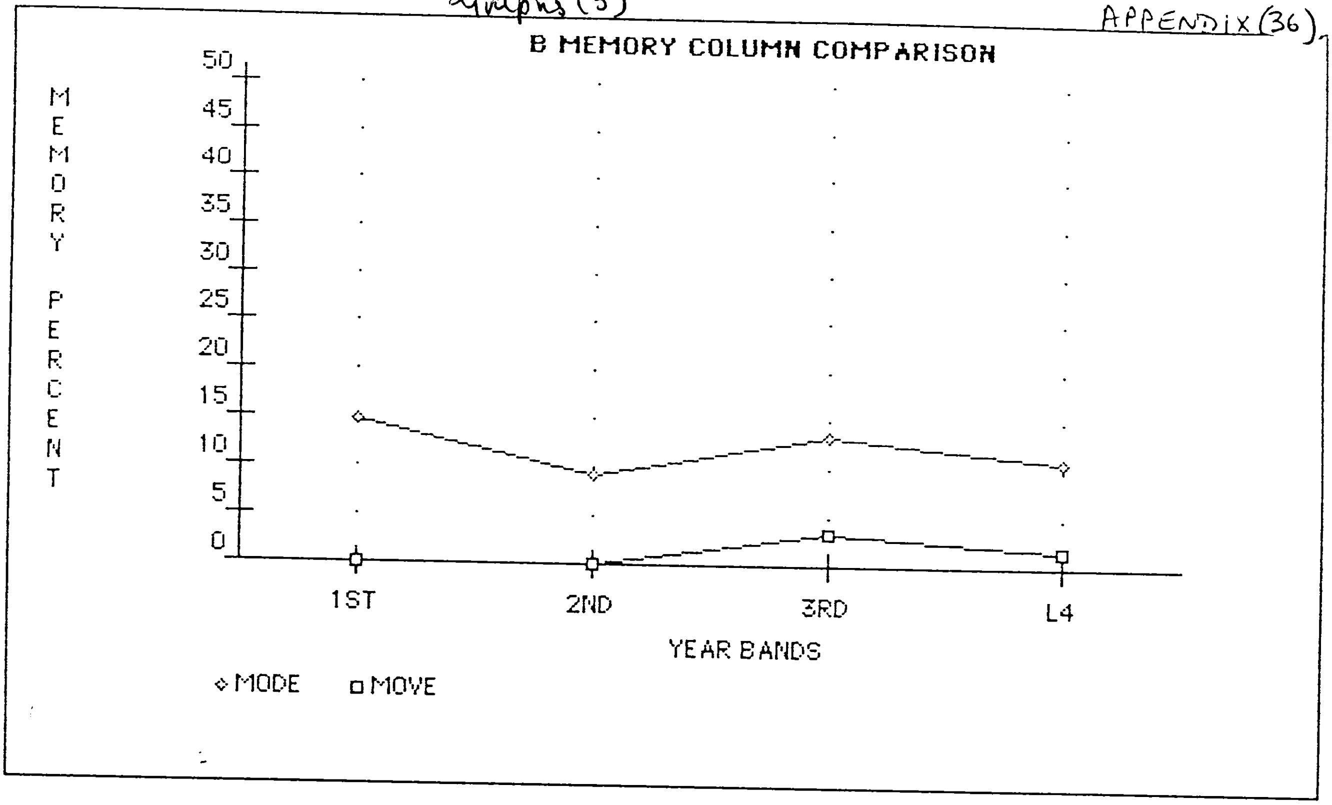
GRAPHS (3)

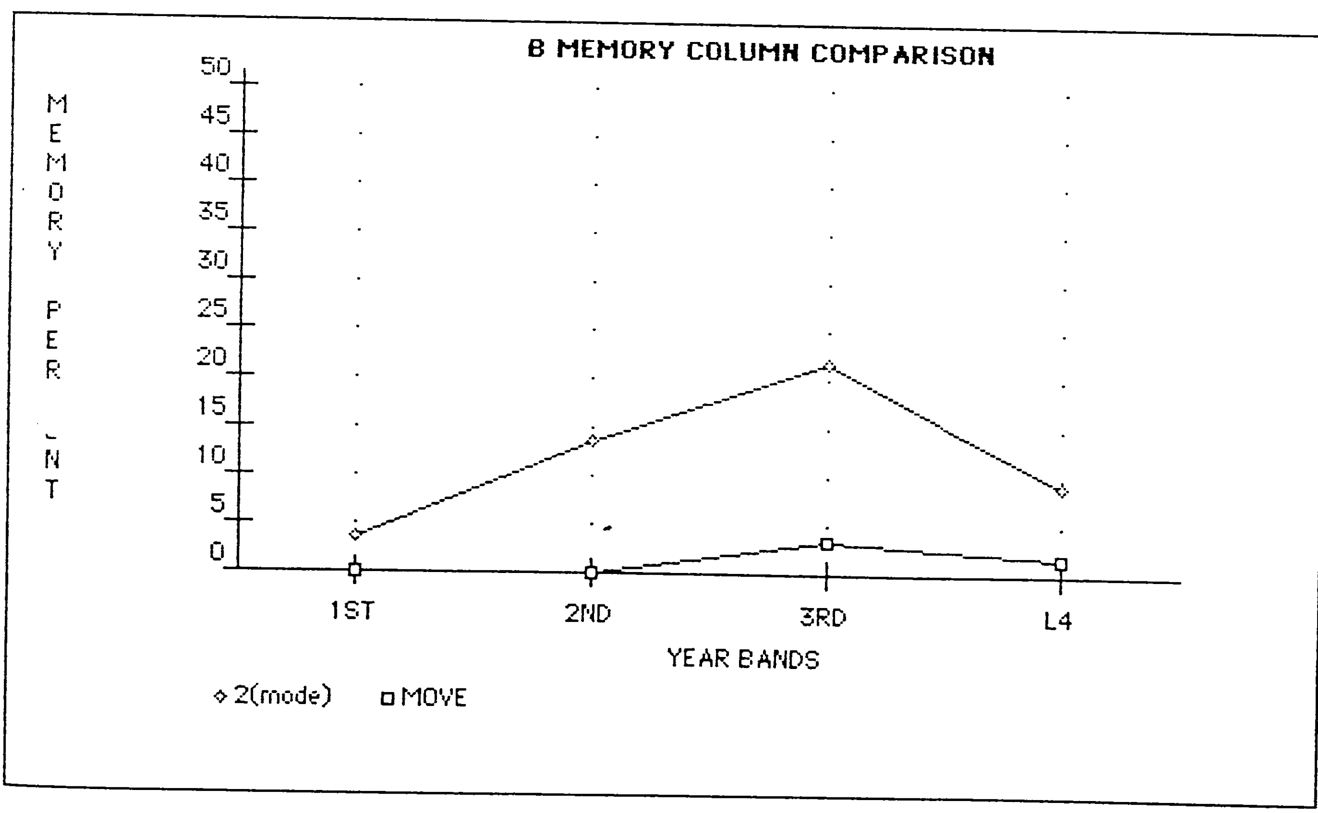

.B MEMORY COLUMN COMPARISON
MEMORY PERCENT
30
27
24
21
18
15
12
9
6
3
0
1ST
2ND
3RD
YEAR BANDS
comma
speech marks


.B MEMORY COLUMN COMPARISON
30
27
24
21
18
15
12
9
6
3
0
1ST
2ND
3RD
YEAR BANDS
10(line no.)
speech marks

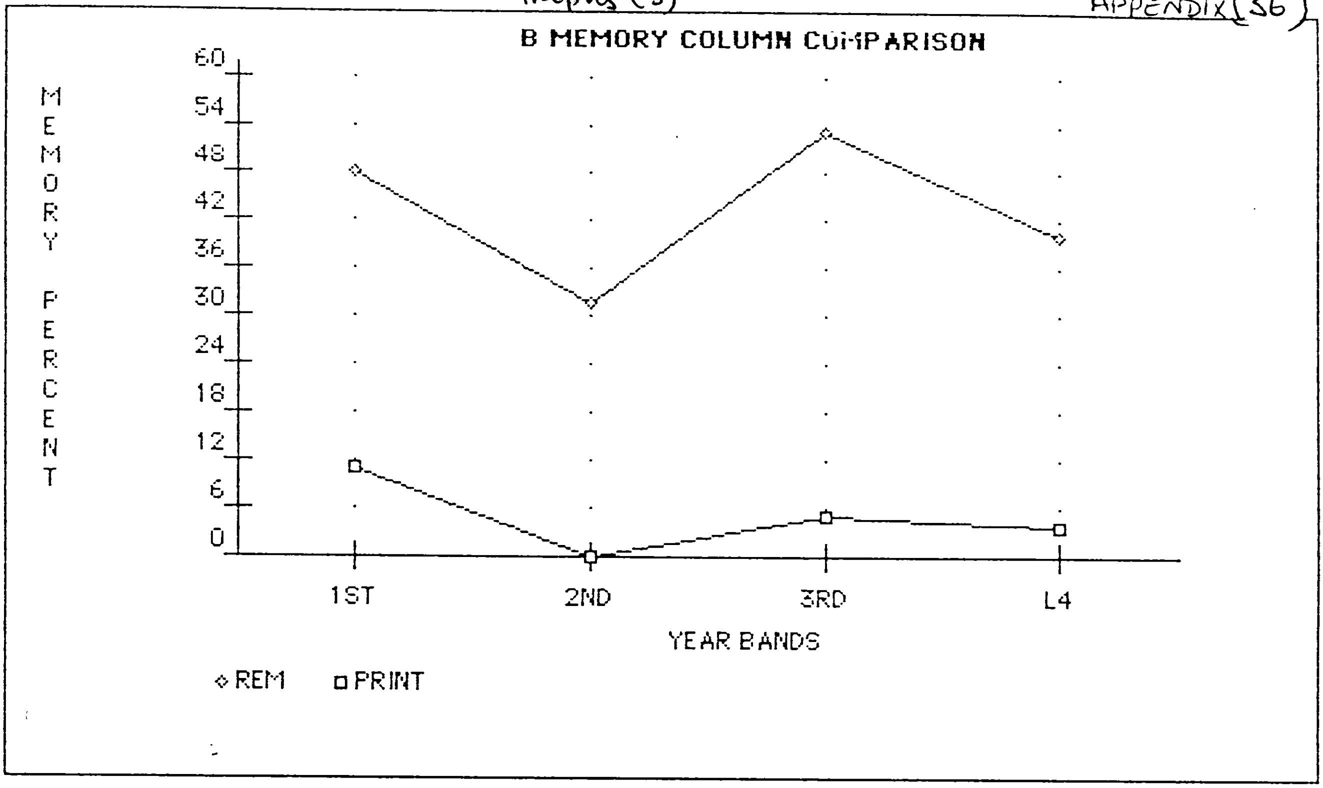
Graphs (3)

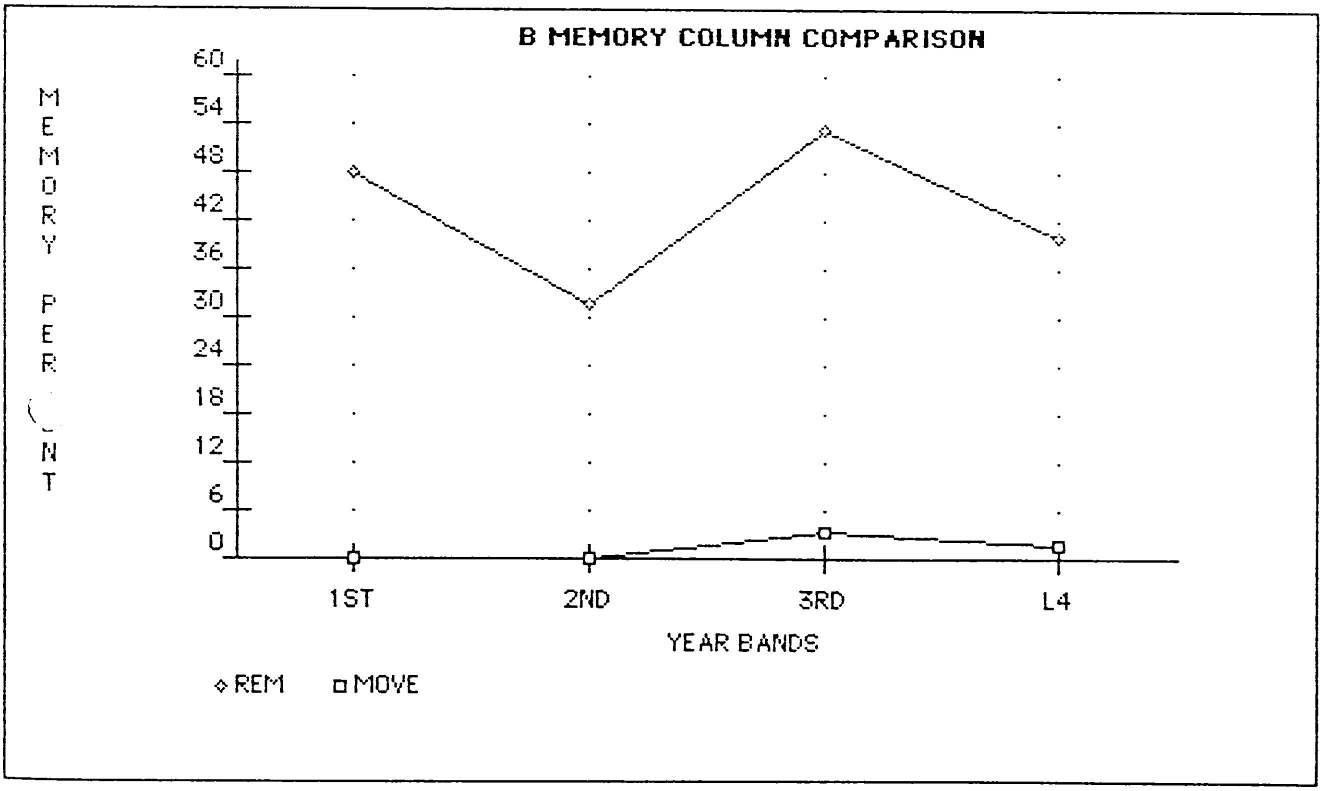
APPENDIX (36)

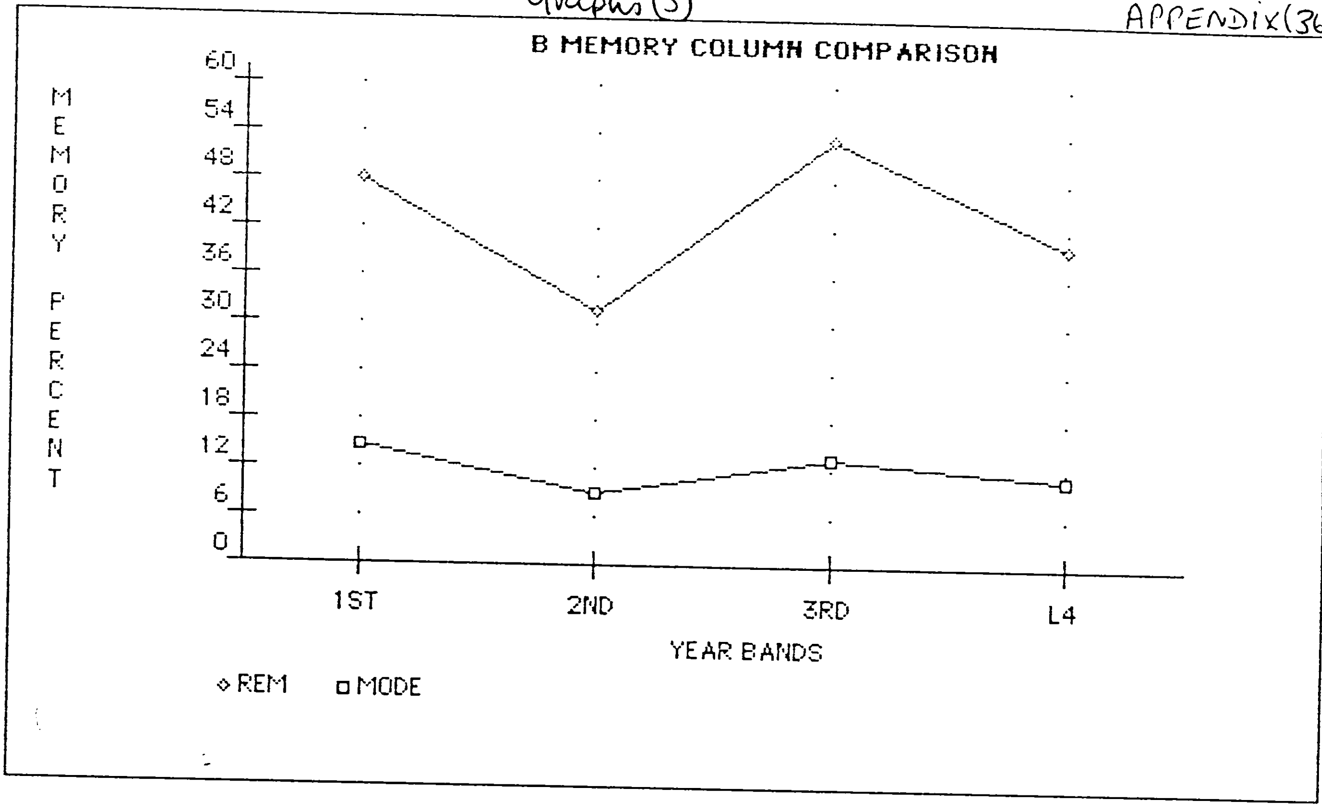


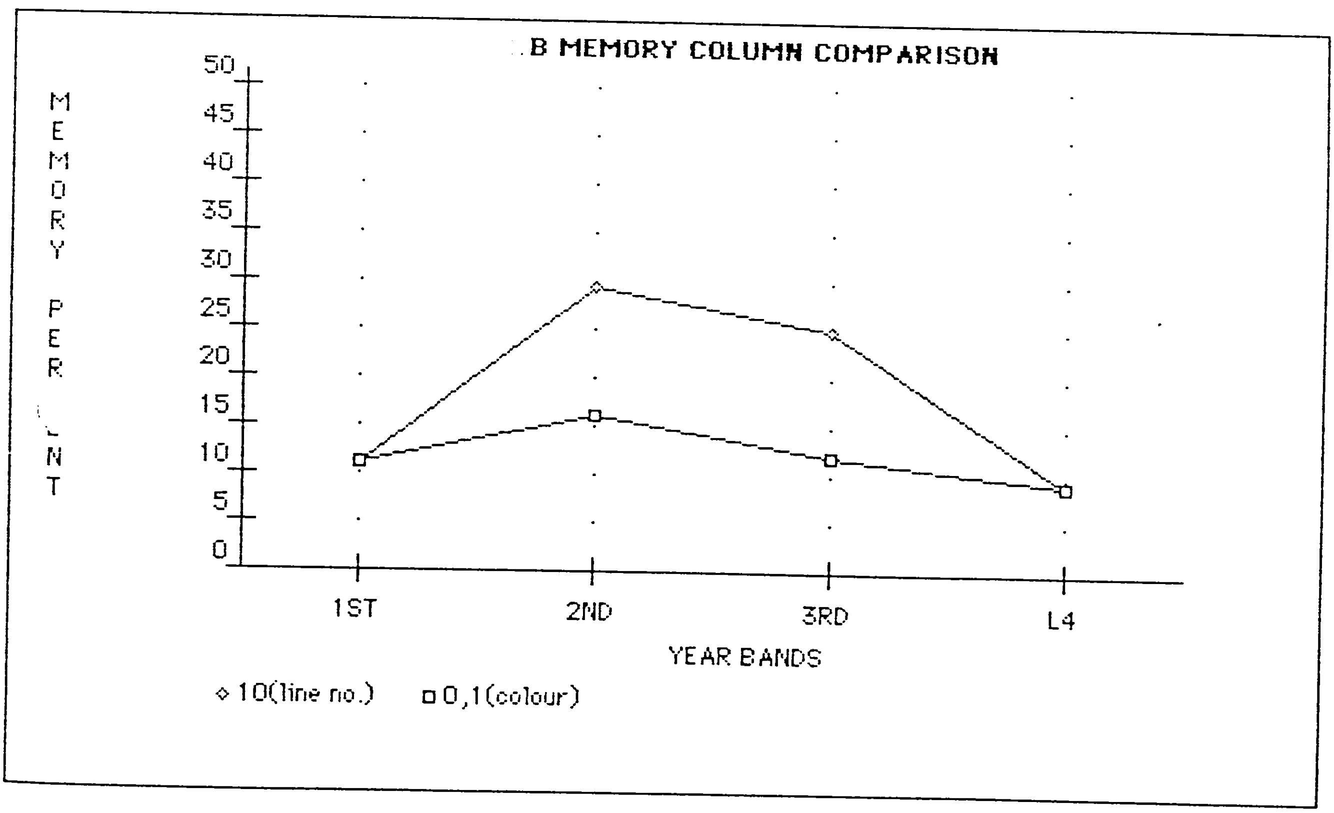


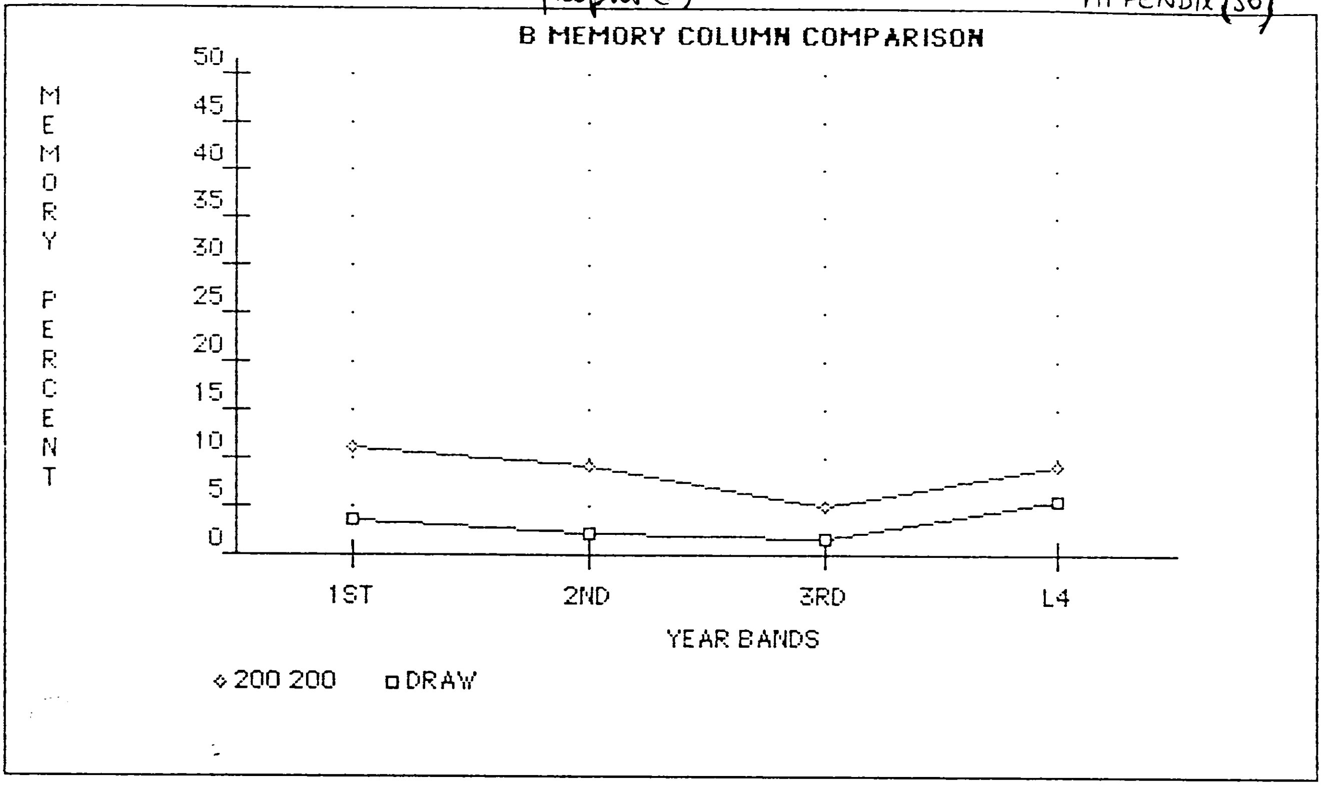
Graphs (3)

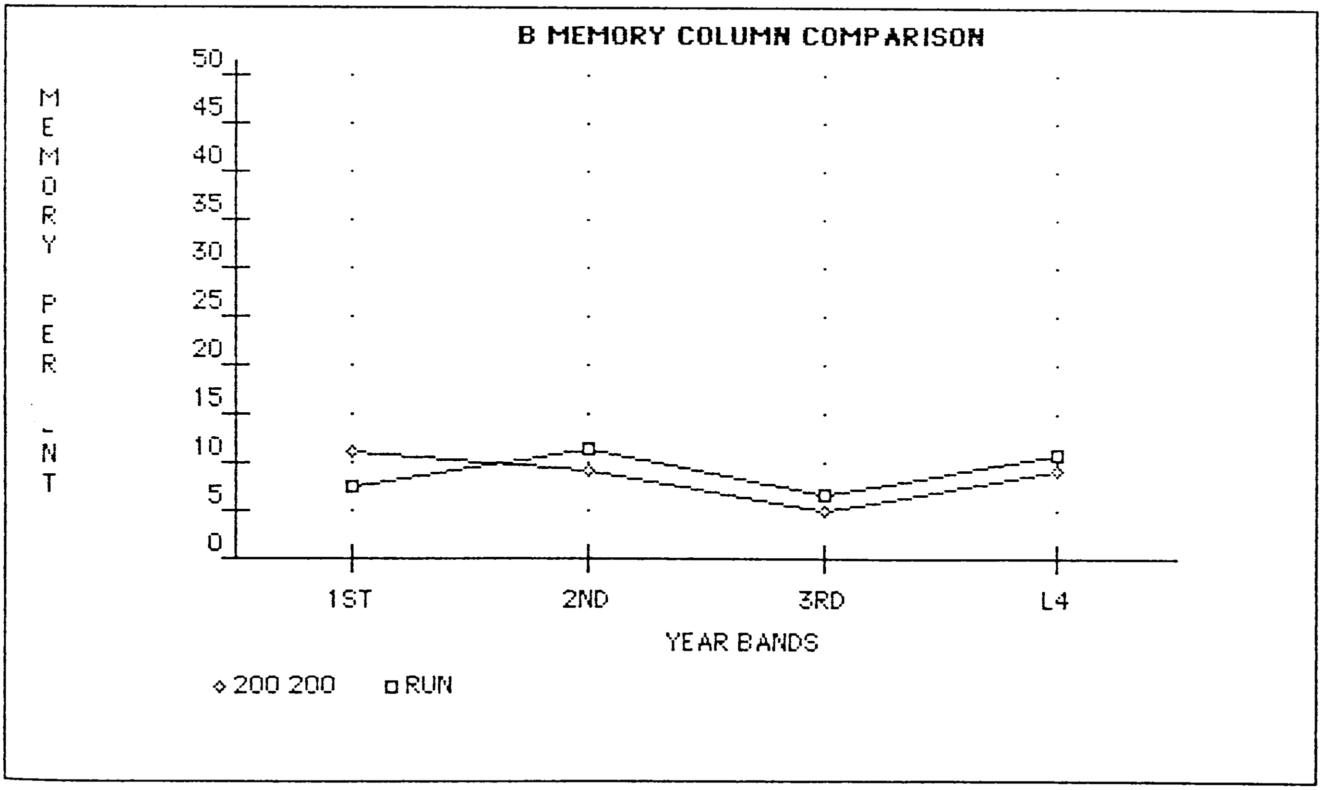

B MEMORY COLUMN COMPARISON
MEMORY PERCENT
60
54
48
42
36
30
24
18
12
6
0
1ST
2ND
3RD
L4
YEAR BANDS
◇ REM
□ PRINT

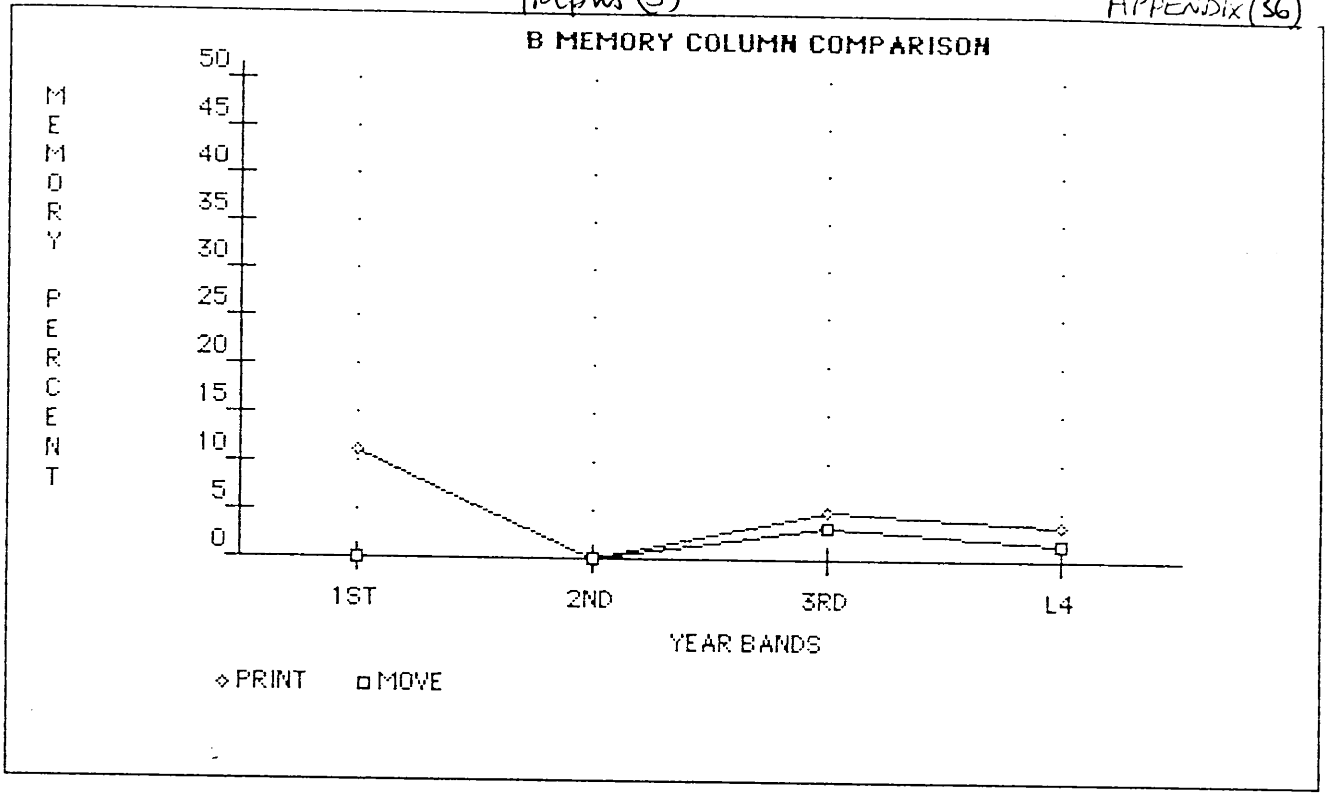

B MEMORY COLUMN COMPARISON
MEMORY PERCENT
60
54
48
42
36
30
24
18
12
6
0
1ST
2ND
3RD
L4
YEAR BANDS
◇ REM
□ MOVE

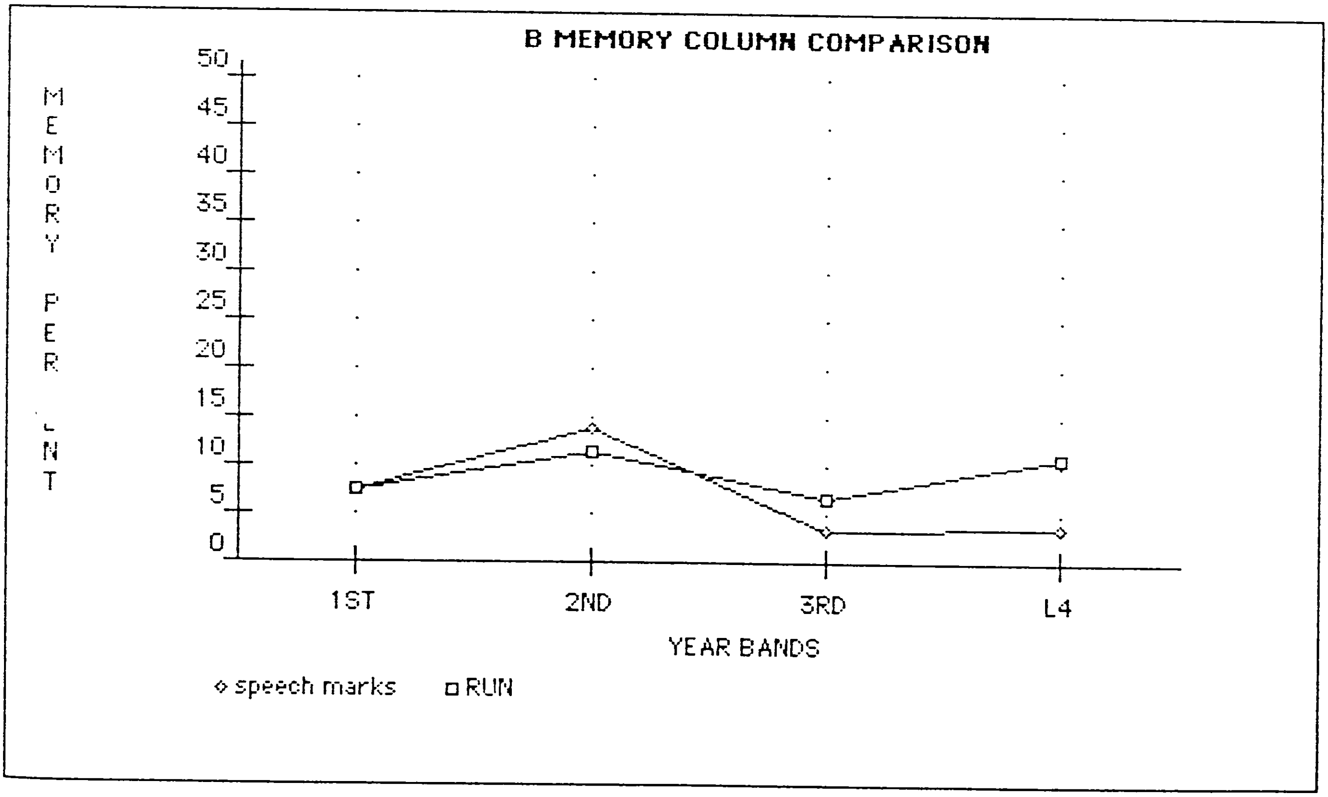

B MEMORY COLUMN COMPARISON
MEMORY PERCENT
60
54
48
42
36
30
24
18
12
6
0
1ST
2ND
3RD
L4
YEAR BANDS
REM
MODE


B MEMORY COLUMN COMPARISON
MEMORY PER NT
50
45
40
35
30
25
20
15
10
5
0
1ST
2ND
3RD
L4
YEAR BANDS
10(line no.)
0,1(colour)

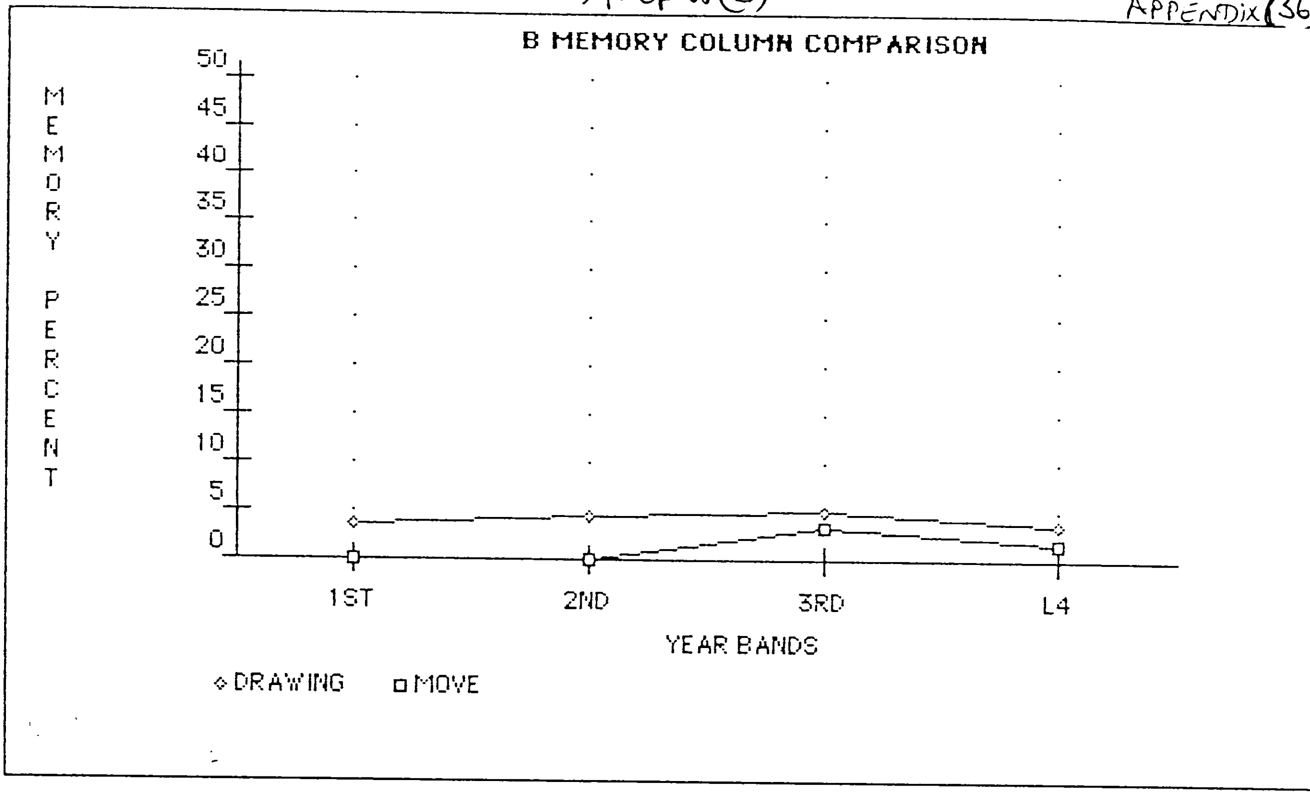

B MEMORY COLUMN COMPARISON
MEMORY PERCENT
50
45
40
35
30
25
20
15
10
5
0
1ST
2ND
3RD
L4
YEAR BANDS
200 200
DRAW

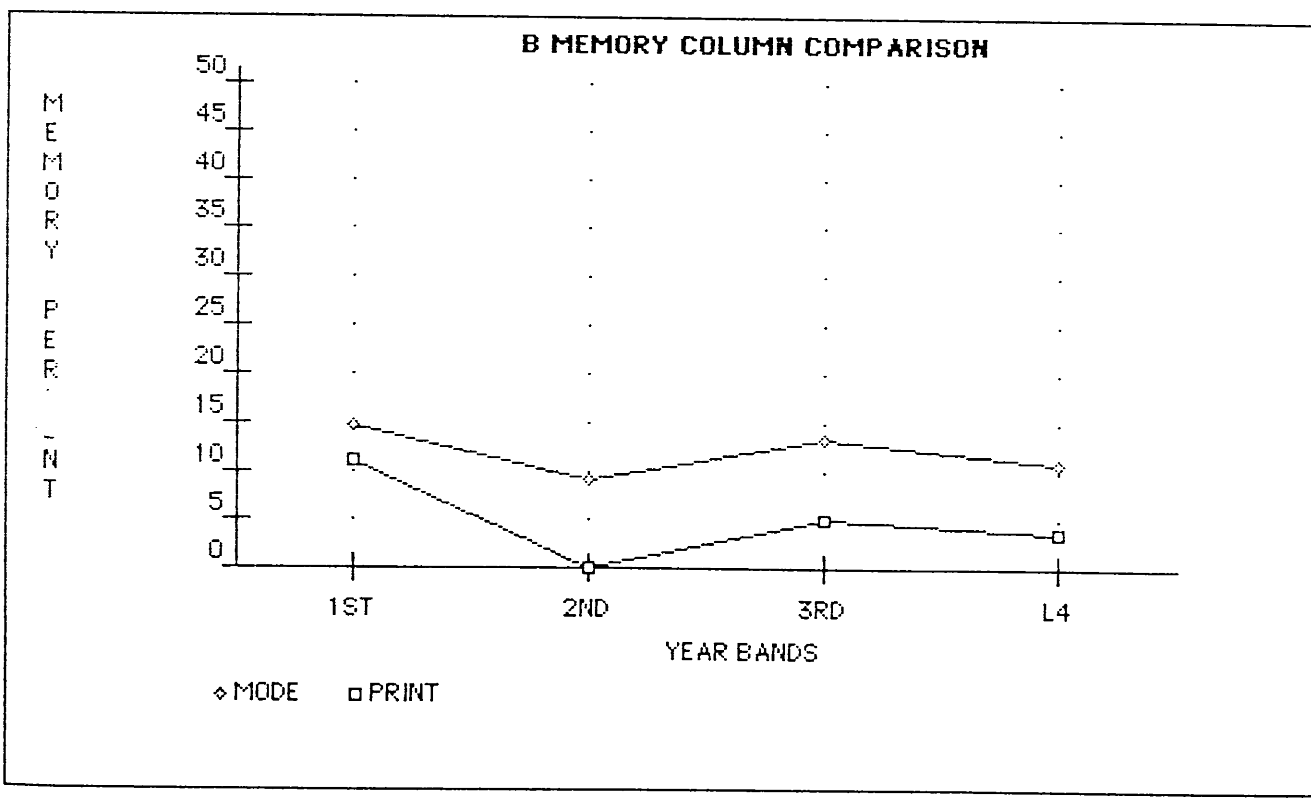

B MEMORY COLUMN COMPARISON
MEMORY PER NT
50
45
40
35
30
25
20
15
10
5
0
1ST
2ND
3RD
L4
YEAR BANDS
200 200
RUN


B MEMORY COLUMN COMPARISON
MEMORY PERCENT
50
45
40
35
30
25
20
15
10
5
0
1ST
2ND
3RD
L4
YEAR BANDS
PRINT
MOVE

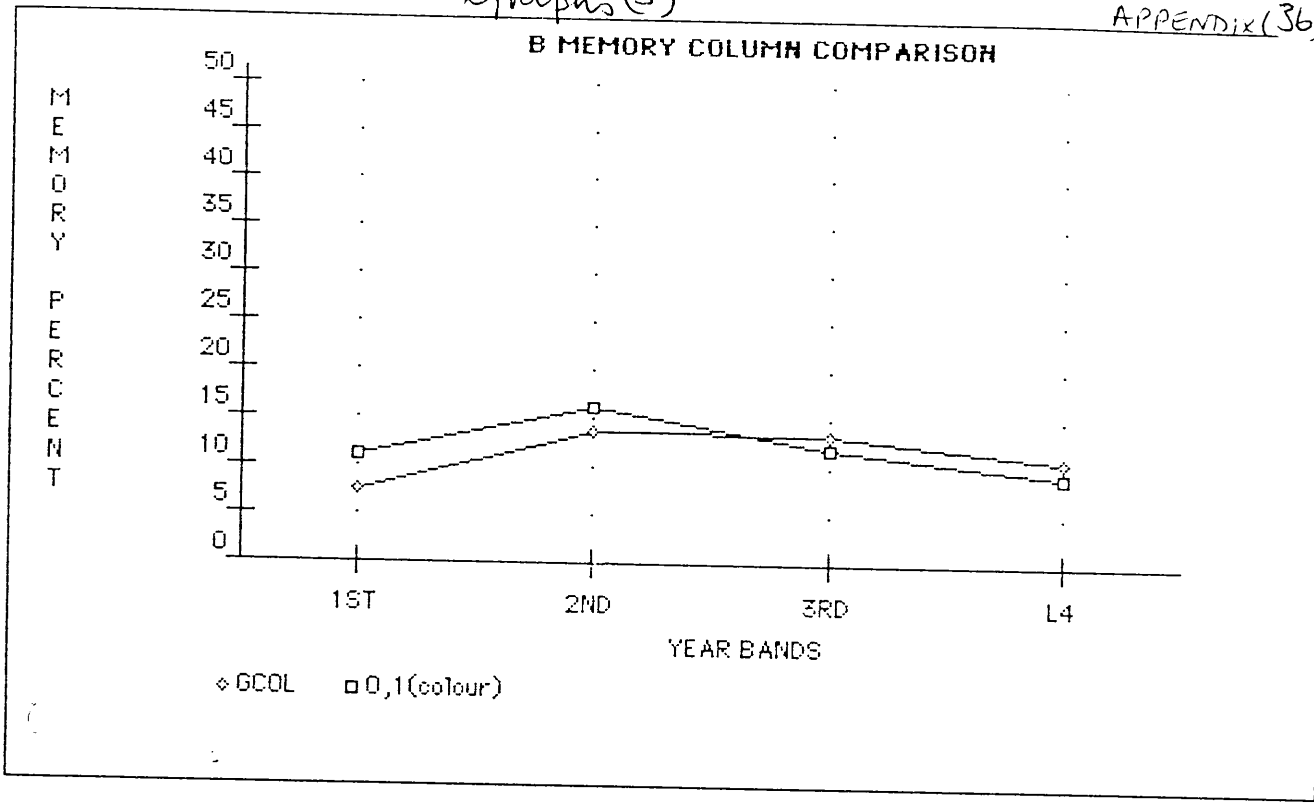

B MEMORY COLUMN COMPARISON
MEMORY PER NT
50
45
40
35
30
25
20
15
10
5
0
1ST
2ND
3RD
L4
YEAR BANDS
speech marks
RUN

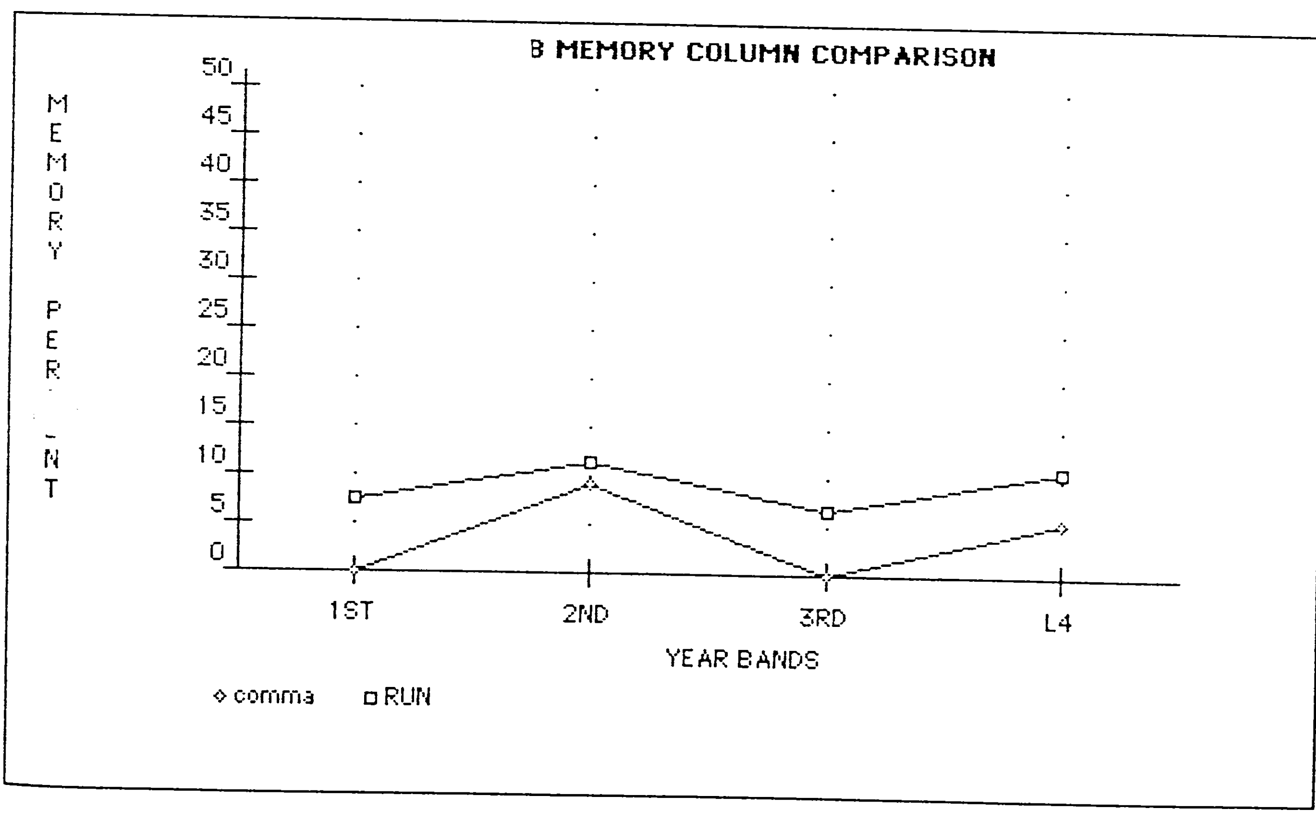
Graphs (3)


B MEMORY COLUMN COMPARISON
MEMORY PERCENT
50
45
40
35
30
25
20
15
10
5
0
1ST
2ND
3RD
L4
YEAR BANDS
DRAWING
MOVE


B MEMORY COLUMN COMPARISON
MEMORY PER NT
50
45
40
35
30
25
20
15
10
5
0
1ST
2ND
3RD
L4
YEAR BANDS
MODE
PRINT

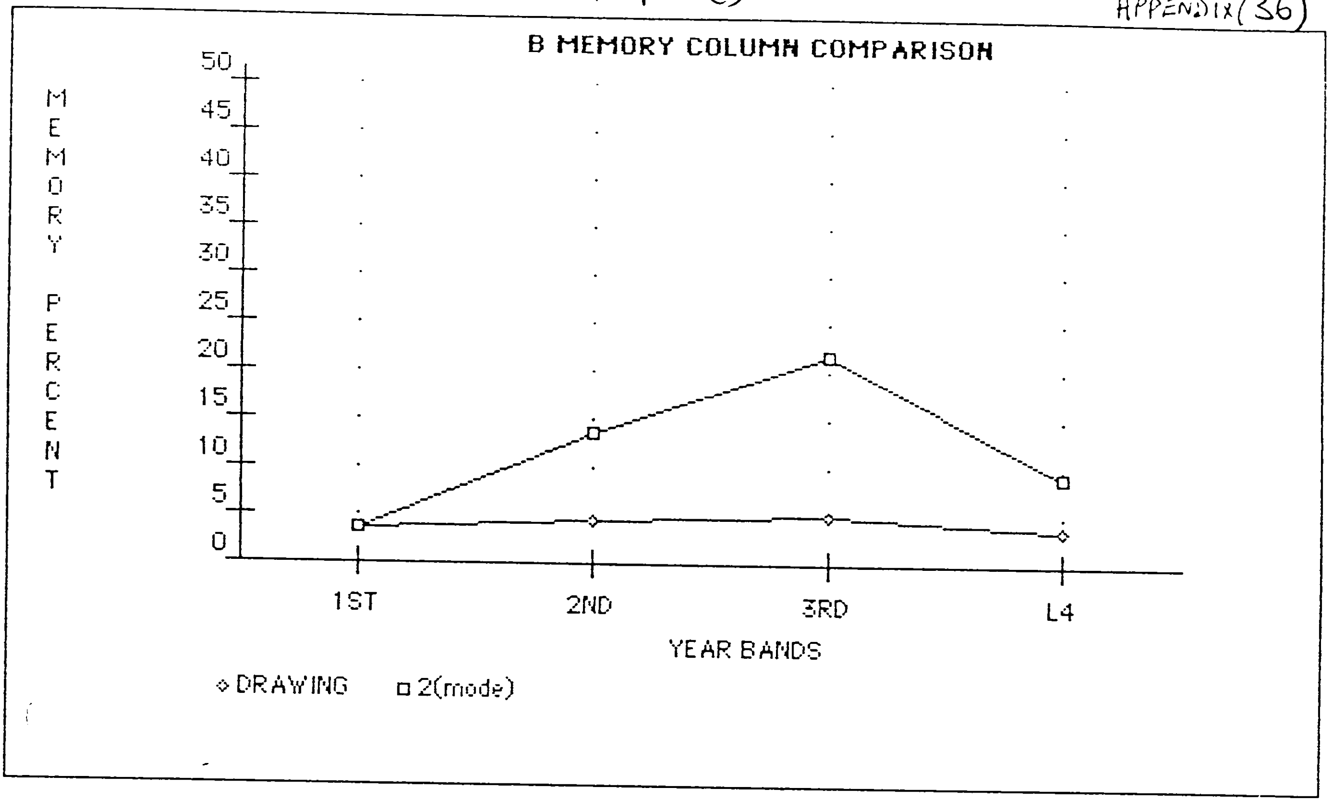
Graphs (3)

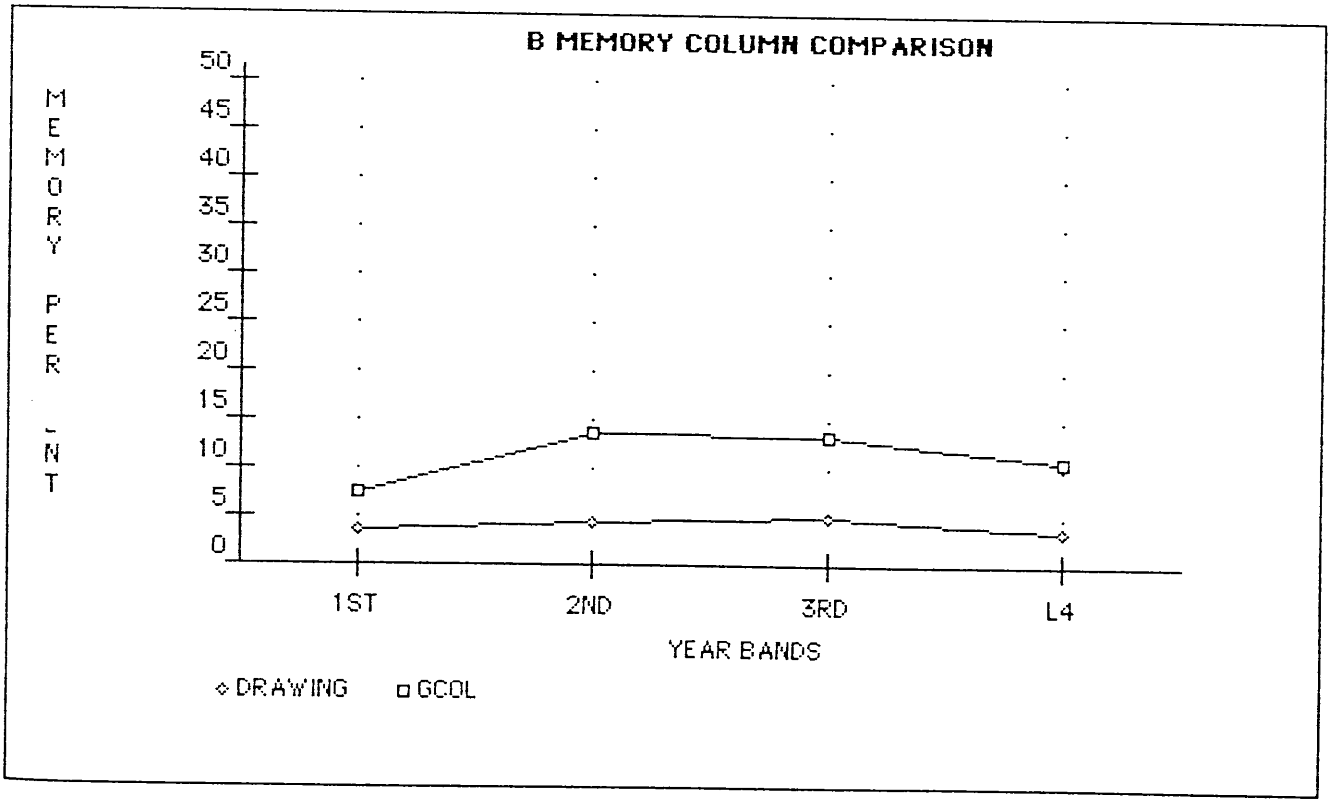

B MEMORY COLUMN COMPARISON
MEMORY PERCENT
50
45
40
35
30
25
20
15
10
5
0
1ST
2ND
3RD
L4
YEAR BANDS
GCOL
0,1(colour)

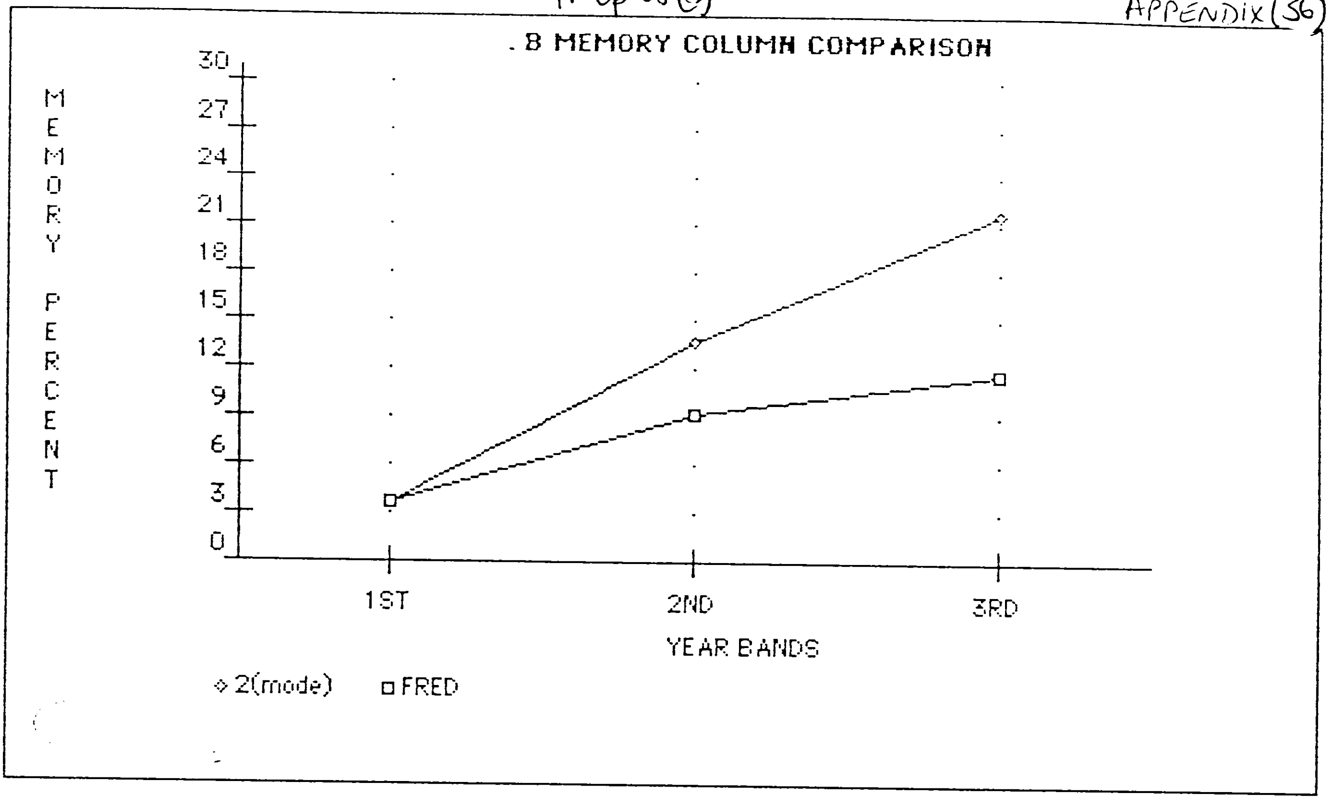

B MEMORY COLUMN COMPARISON
MEMORY PER NT
50
45
40
35
30
25
20
15
10
5
0
1ST
2ND
3RD
L4
YEAR BANDS
comma
RUN

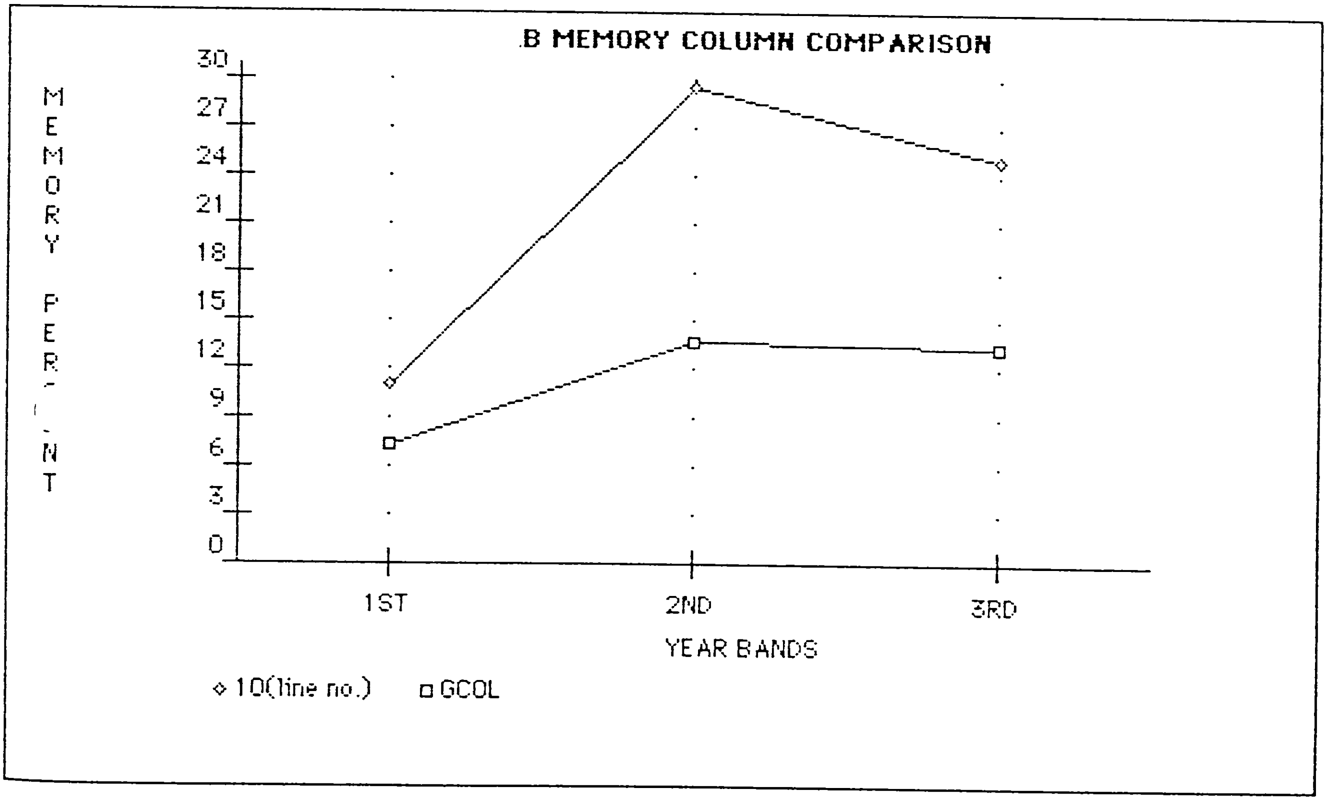
Graphs (3)

APPENDIX (36)





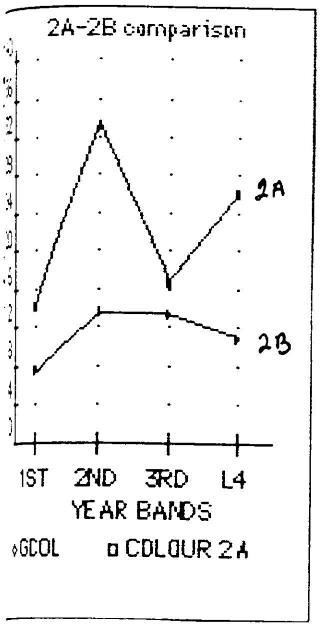

.B MEMORY COLUMN COMPARISON
MEMORY PERCENT
30
27
24
21
18
15
12
9
6
3
0
1ST
2ND
3RD
YEAR BANDS
◇ 2(mode)
□ FRED

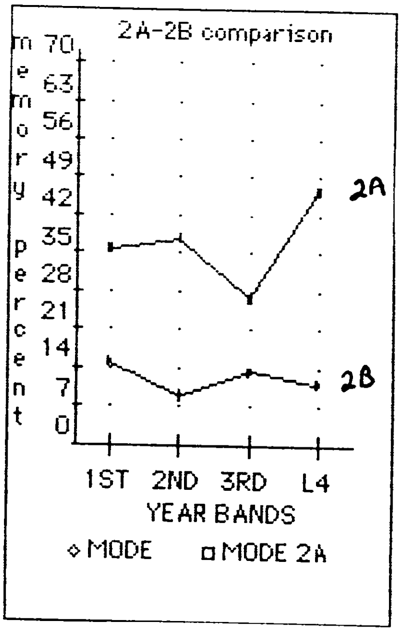

.B MEMORY COLUMN COMPARISON
MEMORY PERCENT
30
27
24
21
18
15
12
9
6
3
0
1ST
2ND
3RD
YEAR BANDS
◇ 10(line no.)
□ GCOL

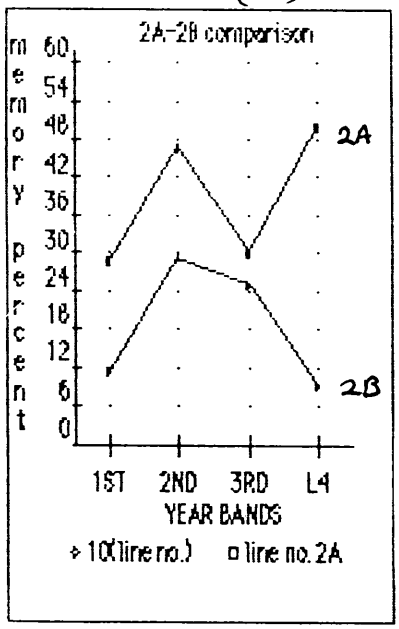
APPENDIX 37

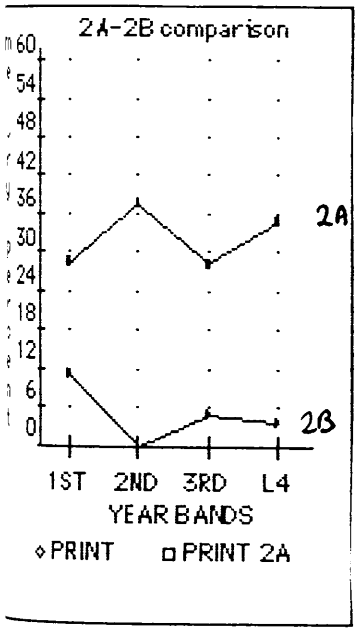
APPENDIX (37)

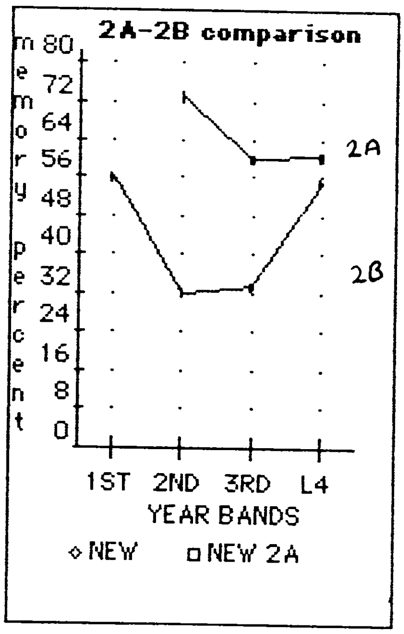
GRAPH SHEET (5)

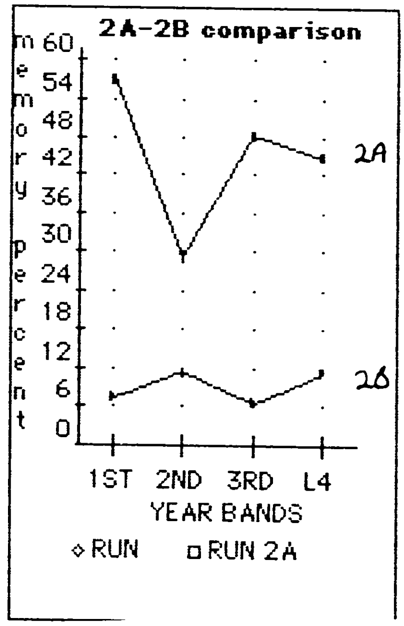


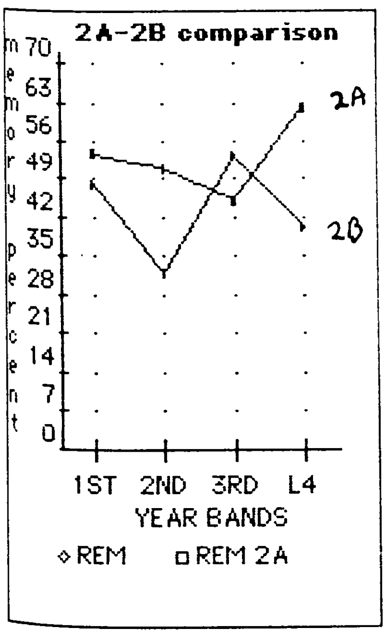


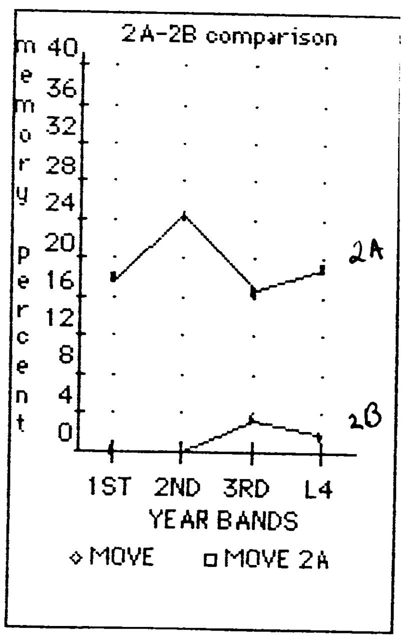


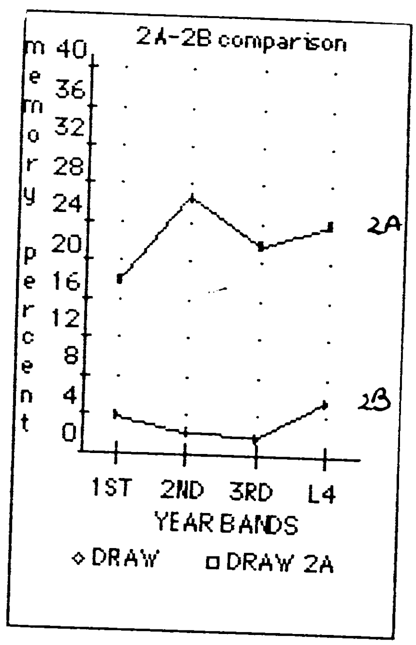












APPENDIX 38

KENDALLS COEFFICIENT OF CONCORDANCE W;STUDYC **APPENDIX[38]**

YEAR	CODE													
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	9	12.5	2	9	3	12.5	12.5	7	1	12.5	4	5	9	6
L4	6	11.5	1.5	11.5	3	11.5	11.5	11.5	1.5	11.5	4	7.5	7.5	5
U4	6.5	2.5	2.5	10.5	6.5	4	10.5	6.5	1	10.5	10.5	13.5	13.5	6.5
TOT	21.5	26.5	6	31	12.5	28	34.5	25	3.5	34.5	18.5	26	30	17.5
SUM	**315**	**M=**	**22.5**											
TIES:														
K1	7													
K2	18.33333													
K3	11.16667													
WG	36.5													
S=	1	16	272.25	72.25	100	30.25	144	6.25	361	144	16	12.25	56.25	25
TOT	**1256.5**													
W=	**0.63592**													
CRITICAL VALUE AT 0.05 =22.36														
OBTAINED VALUE	=	**X^2=**	**24.8**	**DF=13**										

APPENDIX 39

Study C. VARIANCE ANALYSIS

1. GRAND TOTAL: GT= (77+44+12)^2/49
=**361**

2. SS_{total} = (271+164+30) - GT
= 465-361 = **104**

3. $SS_{between}$ = 77^2/24+ 44^2/14 + 12^2/11 -GT
=247.04167 + 138.28571+13.090909- 361
=398.41829- 361
= **37.41829**

4. SS_{within} = SS_{total} - $SS_{between}$
=104-37.41829
= **66.58171**

5. DFsstotal =(N-1) = **48**
DFssbetween= (k-1)=**2**
Dfsswithin= (N-k)=**46**

6. VARbetween= SSbetween/DFbetween=37.41829/2 =**18.709**
VARwithin = SSwithin/DFwithin =66.58171/46= **1.4474285**

7. F=VARbetween/VARwithin=18.709/1.4474285=**12.925682**

ANALYSIS OF VARIANCE TABLE:

Source of Variation	Sum of squares	Df	Variance	F
Between groups	37.41831	2	18.709	12.93
Within groups	66.581681	46	1.4474279	
Total	103.99999	48		

CRITICAL VALUES: at 0.05 3.1504< F <3.2317 and at 0.01 5.1785 > F > 4.9774
OBTAINED VALUE: F=12.93

Study C prediction of the modified program.

Scheffe Test for comparing means:

I=S*(SQR(VARwithin)(Wg)

=S* (SQR(1.4474279)(0.366))

=S*(0.7278452)

S=SQR(k-1)(F.05) Or S=SQR(k-1)(F.01)

S= 2.53 or S=3.16

I for 0.05 = 1.838537 and I for 0.01 = 2.2963516

To be significant I should be smaller than the difference between any two means:

M1-M2 = 0.06

M1-M3= 2.11

M2-M3=2.05 therefore both significant at 0.05 level. The mean for U4 group is significantly different to the means of the 1st and l4yrs.

Study C Raw scores from 14 categories. Allocated across the age bands

as follows- raw scores

YEAR	CODE 1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	0	21	1	20	0	0	2	22	0	16	5	1	3
L4	2	0	12	0	11	0	0	0	12	0	9	1	1	3
U4	2	4	4	1	2	3	1	2	5	1	1	0	0	2

StudyC; One-way Anova items 3,5,9,11 used
in the modified program prediction

1st year		L4 year		U4 year	
X1	X1^2	X2	X2^2	X3	X3^2
3	9	0	0	0	0
2	4	0	0	0	0
4	16	3	9	0	0
4	16	3	9	0	0
4	16	4	16	2	4
4	16	4	16	3	9
4	16	4	16	0	0
4	16	4	16	3	9
4	16	4	16	2	4
4	16	4	16	2	4
4	16	4	16	0	0
4	16	4	16		
4	16	3	9		
2	4	3	9		
4	16				
2	4				
3	9				
3	9				
3	9				
3	9				
3	9				
3	9				
2	4				
0	0				
SUM					
77	271	44	164	12	30

APPENDIX 40

Study C Loops

P=PRINT repetition loop D=DELAY (time) loop

S=SPACE creating loop N=No loop mentioned

RAW SCORES

CATEGORY	1ST(24)	L4(14)	U4(11)
P	16	8	3
D	4	1	0
S	1	1	0
N	2	1	4
%age	96%	79%	64%
Fault anticipated by			
	1	2	4
%age	4%	14%	36%

APPENDIX 41

When you want the computer to type your name on the screen you write a program which includes:

PRINT "CHARLOTTE"

When your name appears on the screen it appears without the inverted commas (or speech marks). Why aren't they there?

YEAR 1

1. The computer forgets it.
2. I think they stay in the computer's memory till you ask them to come out.
3. The "" go to the memory.
4. The speech marks get blotted out.
5. The speech marks tell the computer which part to print and not to print.
6. The speak marks go back into the computers memory so that they can be used again, when you press shift and number 2 for the next time you need to use the speech marks again.
7. I think it is covered up by a black splode.
8. The speech marks are printed when you press a button it you don't press a button the print won't print them and it is put away.
9. They don't come on the screen because I told it not to by moving the cursor right.
10. They will not print it in the program because it hasn't finished the line."" is important.
11. I think they go into the memory bank but it won't come back until we do a special thing.
12. They don't keep them because it won't understand if you didn't. There essential. They go back into the computer.
13. The computer keeps them in his memory.
14. They aren't on the screen any more because we ask a question and they give it back with no speech mark because we tell them to print it out nicely with no print.
15. Their blots go behind the screen when you press curtain buttons.
16. I think that the speech marks just help in the printing and when you list it the computer doesn't need them any more.
17. You press copy and the curser keys to make it go away. The computer takes them away and stores them away. You can't get them back if you take them away.
18. Their dots behind and when you press a knob it comes up.
19. You do this because it needs to underline the sentence because it is in capitals no one can print it because it in the micro chip.It doesn't take in the line numbers.

cont......

20. The "" just go into the computers (brain) memory but you must put them there out of the computers memory.
21. I think they go to the CPU.
22. I think it goes into the computers memory.
23. I think they go to the computer's memory.
24. The speech marks dissapear into the computers memory and come back when we press the right button.
25. I think that they are speach marks that dissapear becuase they are not part of the wording you are trying to print.
26. I think the computer keep the speach marks in its memory but remembers what is inside them and prints the words inside.
27. The computer keep it in his memory.

2BLUE

1. I think that the inverted commas are there to help the computer print and they just disappear back into the computer.
2. The inverted comas stay with the word PRINT because if the inverted comas were there the PRINT would have to be in too.
3. I think the inverted commas go into the memory of the computer so if the program was run again the computer would remember to print it.
4. I think that they would not come up on the screen but stay in the memory.
5. I think they stay in the program(list makes the whole program come back with "")
6. I think the inverted cammas would disappear and so you would only have Matthew Haslum.(child's name)
7. I think they would disappear.
8. they disapeer.
9. I think that they are deleted off of the computer and forgotton about.
10. The inverted commas go away because they are only to tell the computer what to print.
11. The inverted commas disappear.
12. They help the computer understand but they just go inside the computer I think.
13. They go into the computers memory bank.
14. I think they go into the back of the machine where they are stored.
15. The inverted commas are telling the computer what to print.
16. The inverted commas get discharged from the screen and sent back to the computers memory where it waits for next time.
17. The inverted commas are kept in the program for other commands.
18. As the program goes the inverted commas go with it into the computers memory.

cont.....

19. The go inside the computer because we dont want to see them.
20. I think that the speech marks go away because it is a command and is telling want to print on the screen.
21. I think it goes to the computers brain.
22. The inverted commas tell the computer when to start and stop printing. When it has received the message it will discard the inverted commas.
23. I think the inverted comas disappear into the computer.

2RED

1. They stay on the programme and they just show that whatever is inside the inverted commas should be printed.
2. They inverted cammas go to the back of the computer to tell the words in them to be printed.
3. Its just to understand to be printed it won't appear after.
4. I think the inverted commas go back to memory.
5. Disapear in its memory banks to be stored for the programme.
6. The dissapear.
7. They disapear.
8. They disappear.
9. Dissapear to another place on the screen.
10. I think the inverted commas go away like the print does.
11. I think the inverted commas are put in the memory until needed again.
12. I think that the computer writes on the screen no inverted commas at line 10?.
13. I think the inverted commas becomes an error.
14. They disappear when the names spelt out.
15. It makes it clear what you want the computer to print.
16. The move off the screen sideways.
17. They are part of the PRINT command so they are stored inside the computer.
18. I think the inverted commas go into the memory of the program and appear when they are needed.
19. They disappear when you press return.
20. They disappear when you press return.
21. They go back into the computer then store them up and use them again when other people want to print there names.
22. I don't know.

End

APPENDIX 42

<u>Raw Scores for Categories A,B,C,D,E.</u>

Year	A	B	C	D	E	TOTAL
1	3	2	6	11	5	27
2	6	16	11	10	2	45

Categories A and E combined to increase all expected incidence scores to above 5

Chi Square kxn table

<u>Observed</u>

YEAR	B	C	D	A/E	TOTAL
1	2	6	11	8	**27**
2	16	11	10	8	**45**
	18	**17**	**21**	**16**	**72**

<u>Expected</u>

6.75	6.38	7.88	6
11.25	10.63	13.13	10

X^2 = 3.34+2.0+0.02+0.01+1.24+0.75+0.67+0.4

<u>X^2 =8.43</u> df=3 (Critical p<0.05 7.81)

APPENDIX 43

NAME......................................DATE......................................FORM...............

(a) How many doors are there on the corridor between the staff room and the steps to the photo-copying room?

(b) How did you arrive at your answer to part (a)?

Answer Classification:

A. 'guessed' given as an answer

B. 'remembered' given as an answer- no indication of visualisation

C. visualisation of the route indicated in the answer

D. actual route given as answer

E. spoiled paper or answer not classified above.

APPENDIX 44

PICTURE EVOKED BY WORD	'A'			Appendix[44]
Word	S1	S2	S3	S4
	37 pupils	36 pupils	55 pupils	53 pupils
Computer	37	34	50	49
VDU	12	16	19	27
K/Board	33	33	51	43
Memory	4	9	9	8
Variable	4	1	2	3
Basic	2	4	1	5
Econet	2	9	10	5
Syntax	6	5	6	6
Storage	12	9	17	9
Brain	35	29	46	41
WORD EVOKED BY WORD	'B'			
Word	S1	S2	S3	S4
Computer	0	2	2	3
VDU	9	10	15	12
K/Board	3	3	3	8
Memory	31	20	36	41
Variable	23	25	37	41
Basic	26	28	47	37
Econet	15	13	27	17
Syntax	13	13	39	25
Storage	16	21	25	35
Brain	2	3	8	11
NOTHING EVOKED BY WORD	'C'			
Word	S1	S2	S3	S4
Computer	0	0	3	1
VDU	16	10	21	14
K/Board	1	1	1	2
Memory	2	7	10	4
Variable	10	10	16	9
Basic	9	4	7	11
Econet	20	14	18	31
Syntax	18	18	10	22
Storage	9	6	13	9
Brain	0	4	1	1

APPENDIX 45

EXAMPLES OF ANSWER CATEGORY 'C': followed by subject year-band in bracket

1. Made a picture in my mind(S1)
2. I pictured it(S1)
3. I had a picture in my mind as if I was following the route(S1)
4. I pictured the school in my head(S1)
5. I walked it out in my mind.(S1)
6. By picturing it(S1)
7. I made a mental picture in my mind(S1)
8. I had the picture on my mind(S1)
9. I imagined I was at the front door, walked upstairs and then came in (S1)
10. By walking the passage through mentally (S2)
11. I imagined myself walking along the corridor (S2)
12. I counted them in my head - I pictured the corridor(S2)
13. I imagined the corridor and counted the doors(S2)
14. I counted the doors in my mind (S2)
15. I imagined myself entering from the front door till the end (S2)
16. I thought back over walking up the stairs(S2)
17. I had a picture in my mind and pretended walking but my mind pictured as I walked(S3)
18. I picture it and count(S3)
19. I counted all the doors in my head (S3)
20. I pictued myself walking there (S3)
21. 2 including the class room door, I just pretended I was walking to the class from the front door.
22. I had a picture in my mind of the passage(S3)
23. I imagined I was doing the journey(S3)
24. Put a pictue in my mind(S3)
25. Visualising picture(S3)
26. Visualised the area in my mind(S3)
27. Pictured myself walking up the stairs and counted (S3)
28. I walked it mentally in my mind(S3)
29. Thought of myself coming up the stairs, turning left and so on(S3)
30. Shut my eyes and pictured the scene from the above(S4)
31. I counted them in my head(S4)

32. I imagined a plan of the area in question(S4)
33. I counted the doors in my head(S4)
34. I mentally walked the route (S4)
35. By simulating the walk again in my mind(S4)
36. I just imagined I was walking the route (S4)
37. Because I imagined that I was there (S4)
38. I imagined them and counted them (S4)
39. I went through my mind imagining myself walking up the stairs past the doors(S4)
40. I just 'looked' in my memory imagining what I was doing and looking on each side to see the doors(S4).

APPENDIX 46

VISUALISING A GIVEN ROUTE: MENTAL IMAGES

CLASS	NUMBER	A	B	C	D	E	C%
1A	18	1	1	16	0	0	89
1B	19	0	3	13	1	2	68
2A	17	0	0	13	1	3	76
2B	19	0	3	13	0	3	68
3A	19	0	1	17	0	1	89
3B	18	1	0	17	0	0	94
3C	18	0	0	17	0	1	94
4A	19	0	1	14	1	3	74
4B	19	0	0	18	1	0	95
4C	15	1	0	13	0	1	87
%	**(181)**	**1.7**	**5.0**	**83.4**	**2.2**	**7.7**	

YEAR	NUMBER		A	B	C	D	E	C%
1st	37	1	4	29	1	2	78	
2nd	36	0	3	26	1	6	72	
3rd	55	1	1	51	0	2	93	
4th	53	1	1	45	2	4	85	

APPENDIX 47

Read each word in the list below and then put a tick in the column which best describes your reaction to that word.

TICK ONE COLUMN ONLY

	PICTURE	WORD	NOTHING
COMPUTER			
VDU			
KEYBOARD			
MEMORY			
VARIABLE			
BASIC			
ECONET			
SYNTAX			
STORAGE			
BRAIN			

APPENDIX 48

Picture evoked by a word Category A

WORD	S1(37)%	S2(36)%	S3(55)%	S4(53)%
COMPUTER	100.0	94.4	90.0	92.5
VDU	32.4	44.4	34.5	50.9
K/BOARD	89.2	91.7	92.7	81.1
MEMORY	10.8	25.0	16.4	15.1
VARIABLE	10.8	2.8	3.6	5.7
BASIC	5.4	11.1	1.8	9.4
ECONET	5.4	25.0	18.2	9.4
SYNTAX	16.2	13.9	10.9	11.3
STORAGE	32.0	25.0	30.9	17.0
BRAIN	94.6	80.6	83.6	77.4

WORDS RANKED IN ORDER OF PICTURE EVOKING PERCENTAGE:

	S1	S2	S3	S4
1.	computer	computer	keyboard	computer
2.	brain	k/board	computer	k/board
3.	k/board	brainbrain	brain	
4.	storage	vdu	vdu	vdu
5.	vdu	storage/ memory/ econet	storage	storage
6.	syntax		econet	memory
7.	memory/ variable		memory	syntax
8.		syntax	syntax	Basic econet
9.	Basic/ Econet	Basic	variable	
10.		variable	Basic	variable

Giving the 3 most 'picture evoking' words of the set across all years:

COMPUTER KEYBOARD and BRAIN

And the 3 least ' picture evoking' words of the set across all years:

BASIC SYNTAX and VARIABLE

APPENDIX 49

A(49)

LOMOND SCHOOL - PICTURE-EVOKING WORDS

ALL YEAR	WORD	S1	S2	S3	S4	Rj	E(Rj)/N	(Rj-E(Rj)/N)	S	W	N=10	X^2=K(N-1)W
%age											K=4	=
94	COMPUTEI	1	1	2	1	5	21.4	268.96	**1122**	**0.8503**		**30.6108**
88.4	KEYBOARI	3	2	1	2	8		179.56				**p<.001**
83.4	BRAIN	2	3	3	3	11		108.16				
41	VDU	5	4	4	4	17		19.36				
26	STORAGE	4	5	5	5	19		5.76				
16.6	MEMORY	7	5	7	6	25		12.96				
14.4	ECONET	9	5	6	8	28		43.56				
12.7	SYNTAX	6	8	8	7	29		57.76				
6.6	BASIC	9	9	10	8	36		213.16				
5.5	VARIABLE	7	10	9	10	36		213.16				

APPENDIX 50

APPENDIX[5C]

Results ranked for PICTURE response:

WORD	PICTURE%	WORD%	NOTHING%
COMPUTER	94.0	3.9	2.1
K/BOARD	88.4	8.8	2.8
BRAIN	83.4	13.3	3.3
VDU	41.0	25.0	34.0
STORAGE	26.0	53.6	20.4
MEMORY	16.6	70.7	12.7
ECONET	14.4	39.7	45.9
SYNTAX	12.7	52.5	34.8
BASIC	6.6	76.2	17.2
VARIABLE	5.5	69.6	24.9

BIBLIOGRAPHY

BANNON L.J (1986), Computer-mediated Communication, in User Centred Design: New Perspectives on Human Computer Interaction. Norman,D.A. & Draper S.W.(Eds.), Lawrence Erlbaum Ass.Inc., Pub. New Jersey.

BARRETT, MARTYN D.(1983)Roehampton Institute, Early Child Development and Care, Vol.12,pp19-28.

BAYMAN P & Mayer R.E.(1982) Novice Users' Misconceptions of Basic Programming Statements Report 82-1, Bayman P, California University, Santa Barbara.

BOBROW D.G.&NORMAN D.A.(1976/7) Some principles of Memory Schemata. In Representation and Understanding (D.G.Bobrow & A.M.Collins eds.)New York, Academic Press.

BONAR J. (1982) Natural Problem Solving Strategies and Programming Language Constructs. Proceedings of the Fourth Annual Conference of the Cognitive Science Society. Ann Arbor MI August 1982.

BORGMAN C.L.(1982), Mental Models: Ways of Looking at a System. Bulletin of the American Society for Information (p38-9).

BROWN J.S.(1986), From Cognitive to Social Ergonomics & Beyond, in User Centred Design: New Perspectives on Human Computer Interaction. Norman,D.A. & Draper S.W.(Eds.), Lawrence Erlbaum Ass.Inc., Pub. New Jersey.

BRUNER J.S.(1957), On Perceptual Readiness, Psychological Review 64, 123-152 and 340-358.

CARROLL J.M.(1985) Minimalist Design for Actual Users. In B.Shackel(Ed) Interact'84 First Conference on Human-Computer Interaction. Amsterdam, N.Holland.

CAVINE C.S, CAVINE E.D. & LAGOWSKI J.J(1981). The effect of Computer Assisted Instruction on the Attitudes of College Students Towards Computers & Chemistry. Journal of Research in Science Teaching 18,4. 329-333

CHENG T.T & STEVENS D.J.(1985) Prioritizing Computer Literacy Topics. Computer Education Vol.9 No.1 pp9-13.Pergamon Press.

CHERNE. L(1977) Creative Computing in AHL D.H.(1986)The best of Creative Computing Vol.1 Creative Computer Press.

CRAIK, K. (1948)'Theory of the Human Operator in Control Systems'. British Journal of Psychology., Vol 38.

de KLEER J & BROWN J.S. (1981)Mental Models of Physical Mechanisms and their Acquisition. In J.R. Anderson Ed. Cognitive Skills and Their Acquisition.Hillsdale, N.J Lawrences Erlbaum Assoc.

deKLEER J & BROWN J.S.(1983) Assumptions and Ambiguities in Mechanistic Mental Models, in Mental Models, Gentner & Stevens(Eds.). L.Erlbaum Assoc.Inc. New Jersey.

DIEM R.A.(1955) A Study of Children's Attitudes and Reactions to the New Technology. Social Education, April p318-320

diSESSA A.(1986) Models of Computation.User Centered System Design Eds.Norman&Draper.LEA

DIXON N.F.(1981), Pre-Conscious Processing, Wiley, London.

DREVER J.(1974),Dictionary of Psychology,Penguin Books.

duBOULAY & O'SHEA(1976) Presenting Computer Concepts to Novices. Edinburgh Dept. Artificial Intelligence Paper No.4.

duBOULAY, O'SHEA &MONK(1980) The Black Box inside the Glass Box, presenting computing concepts to novices. Edingburgh Dept. Artificial Intelligence Paper No.4. 1976 and paper 133.

EYSENCK M.W.(1984), Handbook of Cognitive Psychology. University of London Berbeck College.

FENNEMA, E.H. & SHERMAN, J.A. (1978) Sex Related Differences in Mathematics Achievement and Related Factors: A Further Study, Journal for Research in Mathematics Education 189-202.

FERNALD M.R.(1912),The Diagnosis of Mental Imagery, Psychological Monographs No.58.

FLAVELL J.H. (1963) The Developmental Psychology of Jean Piaget. Van Nostrand Reinhold Co. New York.

FRASER B.J(1977) Attitude to the Socal Implications of Science: its measurement and correlates. Australian Science Teachers Journal 23,1 96-99.

GARDNER P.L.(1975) Attidtudes to Science: a review, Studies in Science Education 2, 1-41.

GARDNER J.R., McEWEN A. & CURRY C.A.(1986)A Sample Survey of Attitudes to Computer Studies, Computer Education Vol 10 No.2. Pergamon Press. (Rec. 24.6.85)

GENTNER D & STEVENS A.L.(Eds.)(1983) Mental Models. Lawrence Erlbaum Assoc. Publishers N.Jersey.

GIBSON, ELEANOR J(1969) Principles of Perceptual Learning and Development, Appleton-Century-Crofts, New York.

GREENO J.C. & STEINER T.E.(1964) Markovian processes with Identifiable states. General Considerations and Applications to all-or-none learning. Psychocentrika 29 309-33.

HELANDER Martin Ed.(1988) Handbook of Human-Computer Interaction Elsevier Science Publishers. Netherlands.

JAMES W.(1890) Principles of Psychology, Holt, New York.

JOHNSON-LAIRD P.N. (1983) Mental Models: Medical Research Council, Applied Psychology Unit Cambs. Cambridge University Press.

JOHNSON-LAIRD P.N. (1988) Fontanna Press ISBN000686089

KAGAN J.(1971) Change and Continuity in Infancy , New York, Wiley.

KAHNEY H. & EISENSTADT M.(1982) Programmer's Mental Models of Their Programming Tasks; The Interactions of Real-World Knowledge and Programming Knowledge. Proceedings of the Fourth Annual Conference of the Cognitive Science Society. Ann Arbor MI August 1982.

KIERAS D.E. & BOVAIR S.(1984) The Role of a Mental Model in Learning to Operate a Device. Cognitive Science 8, 255-273.

KOSSLYN S.M.(1983) Ghosts in The Mind's Machine. Creating and Using Images in the Brain. Image. W.W.Norton & Co.Ltd., New York.

KURLAND D.M. & PEA, R.D.(1983) Children's Mental Models of Recursive Logo. Technical Report No.10. New York Bank Street College of Education.

LARKIN, JILL & RAINARD, BARBARA(1984) A Research Methodology for Studying How People Think, in Journal of Research in Science Teaching Vol 21 No.3 225-234. Uni.Pittsburgh, Pennsylvania 1523 & Community College of Allenghevy County, P15212.

LAVINE D.E.(1965) The Prediction of Academic Performance. Russell Sage, New York.

LEWIS, CLAYTON(1986), Understanding What's Happening in System Interactions, in User Centered Design, New Perspectives on Human Computer Interaction. Norman,D.A. & Draper S.W.(Eds.), Lawrence Erlbaum Ass.Inc., Pub. New Jersey.

LOYD B. & GRESSARD(1984) The Effects of Sex, Age and Computer Experience. A.E.D.S Journal 18.2. p67-77.

MAWBY Ronald et al(1984) Technical Report (No.19) Centre for Children and Technology N.York. Structured Interviews and Children's Conception of Computers, Bank Street College of Education.

MAYER & BAYMEN(1981) Psychology of Calculator Languages.
A Framework for Describing Differences in user's knowledge.
Communications of the ACM. 24. 511-520.

MAYER R.(1982) Diagnosis and Remediation of Computer Programming Skills for Creative Problem Solving Vol.2. Summary of Research for Practitioners. Santa Barbara. California University.

MOORE J.L.(1985) An Empirical Study of Pupils' Attitudes to Computers and Robots. Vol.1 N2 87-98. Journal of Computer Assisted Learning.

NEISSER ULRICH(1976) Cognition & Reality, W.H.Freeman & Co. San Fancisco.

NEWBOLD C.A.(1982) School Examinations in Computer Studies. Computer Education 42, 2-4.

NORMAN D.A (1982) Five Papers on Human-Machine Interaction. Centre for Human Information Processing; San Diego, California University(mfische).

NORMAN D.A.(1984) Stages and Levels in Human-Machine Interaction. International Journal of Man-Machine Studies. 21, 365-375.

NORMAN D.A.(1986), New Views of Information Processing: Implications for Intelligent Decision Support Systems. In Hollnagel, Mancini & Woods(Eds.) Intelligent Decision Aids in Process Environments.
New York, Springer-Verlag.

O'MALLEY C.E.(1986) Helping Users to Help Themselves in User Centred Design: New Perspectives on Human Computer Interaction. Norman, D.A. & Draper S.W.(Eds.), Lawrence Erlbaum Ass.Inc., Pub. New Jersey.

OWEN DAVID (1986) Answers First, Then Questions, in User Centred Design. New Perspectives on Human Computer Interaction. Eds.D.A.Norman & S.W.Draper. Pub. Lawrence Erlbaum Ass. Inc. New Jersey.

PAIVIO A.(1972), A Theoretical Analysis of the Role of Imagery In Learning and Memory. In Sheehan R.W.(Ed) The Function and Nature of Imagery. Academic Press. New York.

PAIVIO A.(1975) Imagery and Long Term Memory. In Kennedy A & Wilkes A(Eds) Studies in Long term Memory, Wiley, London.

PAPERT, S.(1980)Mindstorms: Children, Computers & Powerful Ideas. Sussex, Harvester Press.

PAYNE S.(1991)A Descriptive Study of Mental Models; High Street Bank Machines. Behaviour & Information Technology. Pub. Taylor & Francis London. Vol 10. No.1 3-21.

PIAGET, J. (1952) The Origins of Intelligence in Children. International Universities Press, New York.

REED C.(1985) Why Young and Knowing Hackers Never Play the Game. Computer Guardian 17.10.85 p14.

RICHARDSON J.T.E.(1980),Mental Imagery and Human Memory, MacMillan Press Ltd., London.

RILEY, MARY S.(1986), User Understanding in User Centred Design: New Perspectives on Human Computer Interaction. Norman,D.A. & Draper S.W.(Eds.), Lawrence Erlbaum Ass.Inc., Pub. New Jersey.

RUPIETTA W. (1990) Mental Models in Human/Computer Interaction Eds. Ackerman & Tauber. Elsevier Science. North Holland.

RUSHINEK A, RUSHINEK S & STUTZ J(1984) University Miami, Journal of Educational Technology Systems Vol.13(4). Relationship of Computer Users' Performance to their Attitude Towards Interactive Software.

SHAW D.G, SWIGGER K.M. & HERNDON(1985) Childrens' Questions: A Study of Questions Children Ask While Learning to Use a Computer. Computer Education Vol 9. No.1 pp15-19.

SIMPSON R.D. (1978) 'Relating Student Feelings to Achievement in Science' What Research says to The Science Teacher. Rowe M.B(Ed) Washington DC: National Science Teachers Assoc.

SOLOWAY E., BONAR J. & EHRLICH K.(1983) Cognitive Strategies and Looping Constructs: An Empirical Study. Comm. ACH 1983.

STEVEN'S HANDBOOK OF EXPERIMENTAL PSYCHOLOGY(1988) Eds.Atkinson,Herrnstein, Lindzey & Lucy. Vol.1 Perception & Motivation. Vol.2. Learning & Cognition. Wiley. New York.

TICHNER E.B. (1909), Lectures on Experimental Psychology of Thought Processes. New York MacMillan.

TOMKINS(1962), Affect, Imagery and Consciousness. Vol.1 New York.

WOODWORTH R.S(1938)Experimental Psychology New York, Holt.

YUEN CHI-YIN(1985) Attitudes of Trade and Industrial Teachers Towards the use of micro-computers. Ass.Prof. of Industrical Education, University Southern Mississippi, Hattiesburg MS. Journal of Vocational Educational Research V10, No. 2, 51-64.

STATISTICS

SIEGEL SYDNEY(1988) Nonparametric Statistics for the Behavioural Sciences, McGraw-Hill Book Company

HOLLIDAY M & COHEN L(1982) Statistics for Social Scientists. Harper & Row Ltd.