DOCUMENT RESUME

ED 347 728 EC 301 339

AUTHOR	Cavalier, Al; And Others
TITLE	Technology To Enhance Special Education: Remediation of Problems in Logical Thinking and Memory. Final Report.
INSTITUTION	Association for Retarded Citizens, Arlington, TX.
SPONS AGENCY	Special Education Programs (ED/OSERS), Washington, DC.
PUB DATE	85
CONTRACT	300-84-0156
NOTE	602p.
PUB TYPE	Reports - Descriptive (141)
EDRS PRICE	MF03/PC25 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; Computer Oriented Programs; Computer Software Development; Elementary Secondary Education; Federal Programs; Intervention; *Learning Disabilities; *Memory; *Metacognition; *Mild Mental Retardation; *Moderate Mental Retardation; Remedial Instruction; Student Evaluation

ABSTRACT

A federally sponsored project was designed to incorporate a memory-assessment task and a memory strategy into a computer-based instructional system for assessing and assisting in remediating basic memory-processing and metacognitive deficiencies. The project resulted in an instructional system for school-aged children and youth with mild to moderate mental retardation as well as those with learning disabilities. The computer-based system is based upon an ordered recall task with a circular recall strategy. The system breaks the circular recall memory strategy into its component parts, trains each separately, and then chains the components together. The project involved substantiation of the useability and design of the instructional system, creation of the system, field testing and refinement, and preparation of a marketing plan. This final report describes activities carried out to complete project tasks. The report's appendices comprise the bulk of the document. Appendixes A and B provide the final design report and an addendum, outlining the significance of the problem, the population to benefit from the system to be developed, project timeline, computer screen layouts, plans for testing, commercial publishers' feedback, and other project information. Other appendices provide a user's manual, field test plan, final marketing plan, field test report, preliminary marketing plan, and software coding documentation. (Approximately 120 references in Appendix A) (JDD)

Association for Retarded Citizens of the United States
2501 Avenue J. P.O. Box 6109, Arlington, Texas 76005

TECHNOLOGY TO ENHANCE SPECIAL EDUCATION:

REMEDIATION OF PROBLEMS IN LOGICAL THINKING AND MEMORY

CONTRACT NO. 300-84-0156

FINAL REPORT

PROJECT STAFF:

AL CAVALIER, PH.D

BETH MINEO, PH.D.

SUE EADES

THE BIOENGINEERING PROGRAM

DEPARTMENT OF RESEARCH AND PROGRAM SERVICES

ASSOCIATION FOR RETARDED CITIZENS OF THE UNITED STATES

TABLE OF CONTENTS

TECHNOLOGY TO ENHANCE SPECIAL EDUCATION:

REMEDIATION OF PROBLEMS IN LOGICAL THINKING AND MEMORY

CONTRACT NO. 300-84-0156

FINAL REPORT

PROJECT STAFF:

AL CAVALIER, PH.D

BETH MINEO, PH.D.

SUE EADES

THE BIOENGINEERING PROGRAM

DEPARTMENT OF RESEARCH AND PROGRAM SERVICES

ASSOCIATION FOR RETARDED CITIZENS OF THE UNITED STATES

Task 301 - Substantiation of Useability and Design of the Instructional Systems

The Final Design Report submitted in the spring of 1985 contained the Substantiation Report, program design specifications incorporating reviewer-suggested changes, and comments regarding the feasibility, useability, and marketability of the software by educational software publishers. A copy of the Final Design Report may be found in Appendix A. The following paragraphs provide additional detail regarding the specific activities subsumed under Task 301.

301.1 - Completion of Substantiation Report

A substantiation report was prepared in January and February of 1985. This report described the nature and significance of the targeted problem in the education of students with mental retardation and learning disabilities based on an extensive literature review, detailed the specific features of the instructional packages and how they were to be assembled and programmed, and presented project timelines, a listing of components and costs, a work distribution plan, field test plans, a statement about the size and nature of the population, and a marketing distribution plan.

301.2 - Narrative Description of Programmatic Content

The narrative description, labeled The Program Narrative, was developed in January and February of 1985. Included in this package was a description of the program along with general program information; a document describing specifics of the program such as timing and criterion parameters written specifically for the programmers; a narrative containing screen-by-screen text detail regarding screen activity, verbal instruction, and text instructions; layouts of all screens comprising the program; and a description of the video game to be embedded in the software package.

301.3 - Solicitation and Incorporation of Feedback

Following completion of Tasks 301.1 and 301.2, these documents were sent to expert consultants in the fields of cognitive psychology and special education technology. They were asked to comment on the rationale underlying the project as well as on the appropriateness of the software design. After the Program Narrative and accompanying documents were sent to each reviewer, follow-up telephone calls were made upon receipt of their written feedback.

The reviewers' comments, directed towards the instructional design aspects and the technical use of the computer's features, were very favorable. Reviewer comments were positive and supportive of the manner in which the software design captured the essence of the target cognitive task. Project staff worked with the Program Design Consultant to incorporate specific recommendations on the structure and dynamics. Having made the necessary modifications to the written documentation, project staff submitted the Final Design Report. This document included the Substantiation Report along with the updated Program Narrative.

An additional activity to be included in the work subsumed under this task was the submission of copies of the Final Design Report to a sample of persons representative of the eventual producers and distributors of the software. The purpose of the review was to provide feedback regarding the useability, suitabilty, and marketability of the software. Owing to the competitive dynamics of the educational software industry, project staff experienced difficulty in securing the assistance of many of those approached. The time period required for their review and return of comments exceeded that originally projected, thus forcing project staff to submit the Final Design Report to the federal government without this information. Once feedback had been received from all reviewers, it was incorporated into an addendum document and forwarded to the federal government. A copy of this addendum report may be found in Appendix B.

Task 302 - Creation of the Instructional System

The task of creating an instructional system such as the one proposed in this effort is multi-faceted and complex. For instance, it requires that the project team maintain an awareness of hardware and software limitations without losing sight of the goals of the project. Further, owing to the rapidly changing educational computing market, an awareness of current market trends is essential in order that a software product might be responsive to the immediate and near-future needs of the nation's schools. The instructional system in its finished state is a much more refined entity than that proposed in the Final Design Report. This came about through a continual evolutionary process as we encountered choice points and determined the best solutions based on pedagogical and technological considerations.

A prime example of such a choice was the decision to request a modification to the contract that would allow the software to be designed for MS-DOS computers rather than the Commodore 64. At the time that the original proposal was written, the Commodore and the Apple // series of computers comprised the majority of machines in the schools. In the short time between proposal submission and the commencement of programming activity for this software, the market had shifted and the MS-DOS computers were a strong contender in the educational computing marketplace.

Project staff verified the validity of these trends through consultation with a number of experts before suggesting that a change be made to the contract. Consequences for the project beyond the obvious programming changes were the need to purchase MS-DOS computers and peripherals to replace the Commodore equipment purchased originally. This entailed the need for decisions regarding speech output systems, graphics packages, and a redesign of some components of the software. These changes added considerably to the length of time required for project completion.

The process by which project staff and programmers worked interactively as the programming activities were conducted was a strong point of this project. Programming proceeded according to the specifications set forth in the Program Narrative, and demonstration disks were sent to project staff for review following the completion of each component. Staff would review the disks and then discuss and return a list of necessary changes to the programmer. At times these changes would be a matter of relatively simple reprogramming; at other times the requested change was incompatible with hardware or software limitations, and in these cases the programmer and project staff would work together to identify the solution that was technically feasible while continuing to maintain task integrity. This process of programming and refinement was a very difficult and time-consuming one for a software package this complex. The original amount of time allotted for software completion turned out to be inadequate; thus, project staff requested no-cost time extensions to the contract in order to accommodate these time overruns. These were sound decisions as it was essential to the field tests that we have a complete, valid, and reliable software package to present to the student subjects.

The programming activities were begun in the spring of 1985 with the hiring of the first of three programming teams. The workscope was underestimated by all three teams and project staff found it necessary to terminate subcontracts with the first two teams. The third programming team, who also encountered severe time overruns, nevertheless demonstrated a keen understanding of the target product and a willingness to work with project staff toward the goal of a finished product. In order that the project might be completed as swiftly as possible, project staff determined that the benefits to retaining this programmer outweighed the risks inherent in continuing a relationship that had resulted in behind-schedule performance.

302.1 - Construction of the Flowchart and Operational Program -- Program A

The first programming team that was hired for the task became unable to complete the work in a timely manner due to the loss of key personnel. Therefore, a second programming team was subcontracted to do the work. The workplan called for the Apple version of the software to be programmed first, so the programming team began by working closely with project staff to review and discuss program flow, the student/joystick and light pen interfaces, the driver routines, and the static and moving graphics screens. Project staff periodically reviewed development work.

The team encountered unexpected difficulties in developing the machine code interface routines for the Gibson Light Pen and the Ufonic Speech System. To successfully develop these routines, listings of the products' machine code drivers were required. Steve Gibson, the designer of the sophisticated Gibson Light Pen, had left Koala Technologies and no one in the company was able to provide the support in his absence. A short time later, Steve Gibson was located and provided the necessary information. Unfortunately, this information turned out to be incomplete, and the programming team was required to completely rewrite the routines. Ufonic honored their agreement to provide their proprietary routines; however, their bureaucratic policies created a delay of close to four weeks prior to their release. When this information finally arrived, it turned out to be incompatible with the software's programming language, necessitating a large amount of reprogramming.

After solving the problems with the lightpen and speech routines in September 1985, programming work on the main body of the software proceeded fairly smoothly until December. At that time, when the programming team attempted to link together all of the completed modules using the "C" compiler's linker program, the system would lock up and the modules would not execute. Several unsuccessful attempts were made to circumvent the problems in the compiler's linker, and the technical support personnel at the company that developed the compiler were not helpful. Our programmers rewrote the faulty sections of the compiler code, and this permitted successful linking on an Intel 80186-based machine. Although communication between this machine and the Apple had been established, this communication was still faulty and hindered attempts to link completely the program on the Apple.

At this point project staff realized that, even considering the severe and unanticipated problems that had surfaced with the light pen and speech system driver routines and the problems with the compiler, the programming team was not making sufficient progress to allow the entire project to be completed within the timeframe set forth. Thus, applicants were interviewed for the position, and in April 1986 a new programming team was hired.

Although there was some initial difficulty in getting the old programming team to surrender all relevant materials to the new programmer, this was eventually achieved. It was at this time that project staff, after consultation with national experts, recommended to the federal government a revised workplan to develop the second version of the software for MS-DOS machines rather than the Commodore. Given this change, the new programming team presented project staff with a plan that would concentrate on finishing the MS-DOS version first. The rationale for this plan was that the MS-DOS machine was a much better machine on which to develop code, and once the bugs had been worked out of the MS-DOS version, it would be relatively simple to port to the Apple. This new approach involved an initial port of what had been completed on the Apple by the previous programmer to the MS-DOS machine, then all subsequent programming done on the MS-DOS machine, and then a final port of the MS-DOS version back to the Apple, at which time any sections of code that did not port well would be refined.

The new programmer worked with project staff to identify a speech system and a light pen that would be most appropriate for inclusion in the MS-DOS equipment package. After examination of several speech output systems, the Echo PC synthesizer was selected. To enhance the intelligibility of the speech, we opted to use digitized rather than synthesized utterances, which involved a significant amount of custom encoding of words. The benefits to using the Echo were several; foremost among these is that the Echo is the speech peripheral most commonly found in educational settings. In addition, the customization of words for the MS-DOS machines also makes them available for use with the Apple system as well. In regard to the light pen, the programmer's investigation of reasonably-priced light pens led him to choose one from FTG Data Systems. This proved to be a very good choice, and the FTG representatives further assisted the project by making a loan of five additional pens to assist us in our field test endeavors.

The new programming team required a good deal of time to become familiarized completely with the work that had been done previously and with the specifications for the remainder of the work. Among the problems that the chief grogrammer needed to solve were the refinement of the routines involving the input interface; all confounds needed to be removed from the system so that we could be sure that the data gathered reflected student performance only and not any additional factors introduced by the hardware or software. This was especially crucial in regard to the lightpen activation subroutines and the routine that measured student pause times.

The amount of time allotted for student activity and the branching routines that were activated in response to student input were other aspects of the original programming that required quite a bit of refinement. The programmer also had to rework the video game and create the reward screens that were intended to increase student motivation. The programming language chosen for its portability to other machines is not a particularly good language in which to code graphics, and consequently we opted to modify the heavily graphics-dependent reward screens to present short musical selections along with a very simple graphic.

302.2 - Construction of the Flowchart and Operational Program -- Program B

As described above, since the MS-DOS computers are better development machines, the Apple IIe version of the software became Program B. Once construction and refinement of Program A was completed, it could then be ported to the Apple and refinement of Program B could begin. The porting was relatively straight-forward. Our expectations on areas of the Program A code that may not port well were borne out. Extensive re-coding and new coding were required in the following areas: (a) the graphics prompting screens, (b) graphics characters, (c), the videogame, (d) the music transition screens, (e) precise real-time pause time measurement, (f) the lightpen routines, (g) limited RAM for the assessment overlays, and (h) cursor control.

302.3 - Alpha Testing and Refinement -- Program A

As mentioned previously, an efficient process evolved whereby the programmer delivered work-to-date to project staff, and staff conducted a thorough review, testing each operation and the paths leading to and from various modules. A review period typically consumed three full days or more. Following completion of the review, specific comments would be returned to the programmer that specified the location and nature of the problem and presented potential solutions if one was not readily apparent. These feedback lists typically elicited some protestations from the programmer in which he cited the inability of the hardware to support various software functions or the extreme time costs of proposed changes. Project staff attempted whenever possible to take the programmer's comments into consideration when determining the extent to which these issues needed to be addressed for the final version of the software. In some instances project staff conceded and removed an item from the list of necessary changes; in other cases, the rationale for the importance of the item in relation to the integrity of system design was highlighted for the programmer and an appropriate course of action was determined by those involved.

A number of persons were involved in review of the software at this point. At least six professionals and research assistants conducted thorough examinations, and several students in the public schools also participated in working out the bugs in the field test-ready version. These students were not included in the subject pool for the eventual research study.

A great number of months were spent debugging the software according to the procedure described above. With over 175 separate modules, testing was a very time consuming and precise undertaking. Making this testing even more difficult was the transient nature of some of the problems. In addition, the complete software needed to be recompiled every time a modification was made, and this process itself often caused new problems to surface. The finished code is quite sophisticated and represents extensive refinement and enhancement.

302.4 - Alpha Testing and Refinement -- Program B

The alpha testing and refinement of Program B, the Apple IIe version, proceeded in much the same manner as described in Activity 302.3 above. Because of the comprehensive nature of the instructional system, including student demographics, student customization, assessment, strategy training, reward transition, videogame, data recording, statistical analysis, data reporting, pre-test, and post-test sections, the alpha testing and refinement process was labor-intensive. An increased number of overlays were necessitated by the Apple's limited memory addressing capability and available RAM.

302.5 - Preparation of Program Documentation -- Programs A and B

Although the user's manual was initially scheduled to be prepared after the completion of programming, it was decided that it would be beneficial to the programming task if the manual were prepared much earlier in the process. In this way, the programmers would have ready access to a description of the program as it should run. Our programmers concurred that this sort of a description was extremely helpful to them because it allowed them to see how the system should look to an eventual user. Therefore, a draft of a manual that was in accordance with the specifications contained in the Program Narrative was prepared before programming commenced. We realized at the time of its creation that it would not be the final version of the manual; as refinements to the software occurred, information within the manual would be changed accordingly. Thus, the manual that is presented as Appendix C represents the final project draft upon which ab eventual publisher could build a commercial version.

As the software development proceeded, technical documentation of the coding also progressed. This documentation is vital to the provision of technical support, and enhancement, by an evenetual publisher. This technical documentation of the coding is presented in Appendix H.

Task 303 - Field Testing and Refinement

Field testing was a vital component in the development process because field test results verified the validity of the instructional package's design. The activity subsumed under this task involved the development and presentation of a research proposal to representatives of the Dallas Independent School District, training of research assistants, the actual field evaluation with 60 subjects, refinement of the software based on observations made during field testing, analyses and interpretation of the data, and the preparation of a field test report.

303.1 - Preparation of the Field Test Plan

A field test plan was one of the required deliverables for this project. Included in the plan sent to the project officer were a description of the main research questions; a discussion of the significance of the problem and the rationale underlying the software design; a description of the subject population, measurement instruments, general procedures, research design; presentation of personnel requirements; and samples of the interview questions and consent letters developed for use in the schools. A copy of this plan is included as Appendix D.

303.2 - Teacher Training

In the original proposal the field testing was to be conducted by teachers in their classrooms. Given the overwhelming demands already placed on classroom teachers, however, it was decided to conduct the research under the supervision of research assistants. The benefits to this choice were two-fold: first, the teacher did not have to take time away from classroom activities; and second, having multiple research assistants allowed trial-by-trial monitoring of performance. The rigors of the research design dictated that such precise monitoring be assured. This change to the original plan was reflected in the Field Test Plan.

303.3 - Evaluation with Students and Teachers

A total of 60 students were identified to serve as subjects. These students were located in a total of 15 schools in the Dallas Independent School District. The district had difficulty securing informed consent for several students' participation, and some new subjects needed to be identified. All students were pretested using the pretest capability of the software, and half were randomly assigned to the training condition. In total, 3-1/2 months were needed to permit 30 students to complete the intervention training software and to complete the subsequent post-testing of all 60 students. In addition to their participation as subjects in a study of the efficacy of the system, students' opinions regarding the software package were solicited and information was gathered concerning their familiarity with computer-assisted instruction and other computer-related activities.

Evaluations regarding the software's appeal and useability were also to be sought from teachers. Arrangements were made with the school district to release selected teachers from their classroom duties for one day to allow them to attend a workshop in which the software would be presented for their study and evaluation. The workshop was designed to assess teachers' opinions of the design of the system before they had hands-on experience with it, and then to follow-up the hands-on portion with an evaluation of the software's appropriateness for their students. Due to the programming delays, which in turn placed extreme time constraints on the field testing, the school year ended before we had an opportunity to conduct the teacher workshop. Since this activity has already been designed and approved, it will be easy to conduct the workshop during the next school year.

303.4 - Analysis and Interpretation of Field Test Data

The analyses to be conducted on the field test data were described in the Field Test Plan. Analyses and interpretations can be found in the Field Test Report section. Additional analysis and subsequent interpretation is planned. The results of this research project will be documented and submitted for publication in leading professional journals. All materials resulting from the project will acknowledge the support of the Department of Education, and copies of all materials developed will be forwarded to the project officer.

303.5 - Final Refinement of Hardware and Software

As was mentioned previously in this report, software refinement was an ongoing process. A smoothly-running instructional package is the final product of this project; however, there are several recommendations that we would suggest to the eventual publisher of the system to make it even more effective and responsive to the needs of various user populations. These enhancements are based on information gained from the field test results. More information on this topic is located in the Final Marketing Plan, which can be found in Appendix E.

303.6 - Submission of Field Test Report

The Field Test Report was prepared in order to describe the data collection efforts and to present the data analyses and interpretation. As one of the final deliverables, it is included in this report as Appendix F.

Task 304 - Preparation of Marketing Plan and Recommendations

The work subsumed under this task concerns the activities designed to facilitate transfer of the product of this project to a capable software publisher for eventual commercial distribution. Although this is an area that often receives little emphasis, this project was active in seeking the opinions and recommendations of eventual publishers throughout the course of the project period and in motivating several publishers to consider this software package for inclusion in their commercial offering.

304.1 - Submission of Preliminary Marketing Plan

A preliminary marketing plan was one of the deliverables under this contract. It was submitted to the federal government during the course of the project; a copy of this report may be found in Appendix G.

304.2 - Submission of Marketing Plan for Evaluation by at Least Two Potential Producers/Distributors

Approximately 20 publishers were contacted with information about the project at the conclusion of data collection when a refined version of the software was available for demonstration. They were invited to contact project staff if they were interested in participating in a full-day demonstration of the software that would include discussions about its eventual commercial marketing. Several companies responded, and at this writing two have travelled to Texas for demonstrations, and two others are considering making the trip. The results of these activities are discussed in the Final Marketing Plan, which is enclosed as Appendix E.

304.3 - Submission of Final Marketing Plan Incorporating Feedback from the Field

The Final Marketing Plan is a deliverable under this contract and is included as Appendix E. It serves as a summary of our activities with commercial publishers to date and outlines our plans for timely transfer of the software to a marketer and our commitment to continued involvement and information dissemination in regard to the products of this project.

APPENDIX A

Final Design Report

U.S. DEPARTMENT OF EDUCATION

OFFICE OF SPECIAL EDUCATION PROGRAMS

CONTRACT NO. 300-84-0156

TECHNOLOGY TO ENHANCE SPECIAL EDUCATION:
REMEDIATION OF PROBLEMS IN LOGICAL THINKING AND MEMORY

FINAL DESIGN REPORT

PROJECT STAFF:

AL CAVALIER, PH.D

BETH MINEO, PH.D.

CINDY OLIVER

THE BIOENGINEERING PROGRAM

DEPARTMENT OF RESEARCH AND PROGRAM SERVICES

ASSOCIATION FOR RETARDED CITIZENS OF THE UNITED STATES

TABLE OF CONTENTS

Statement of the Problem

The most common informal observation about mentally retarded children and youth is that they do not "learn" as quickly or thoroughly as their nonretarded peers. Over the past 15 years of research, these observations have been well substantiated. However, a large volume of investigations indicates that these learning problems in many persons with mental retardation are primarily caused not by deficiencies in learning ability per se, but by deficiencies in the person's memory which underlies learning (Belmont & Butterfield, 1969; Detterman, 1979; Ellis, 1970). Several researchers have succeeded in their attempts to improve the memory processes of persons with learning difficulties (e.g., Belmont & Butterfield, 1977; Bray, 1979; Brown, 1978; Kramer & Engle, 1981; Lindgren & Richman, 1984; Swanson, 1983; Torgesen, Murphy & Ivey, 1979).

This project incorporates one of the best and most frequently used memory-assessment tasks along with the most effective memory strategy for that task into a computer-based instructional system for assessing and assisting in remediating basic memory-processing deficiencies. The computer-based system heightens the potential for learning since it incorporates many of the features found to enhance retention by leading cognitive psychologists and special educators.

Significance of the Problem

Initially, the poor memory of persons with mental retardation was attributed to immutable defects in their neurological system (Ellis, 1963). As research techniques and theories become more refined, however, the precision in the understanding of memory deficiencies steadily increased. The most important influence in this movement was the development of sophisticated theories of memory based on computer information-processing models of mental functioning in nonretarded persons (Atkinson & Shiffrin, 1968; Waugh & Norman 1965). When translated from the field of theoretical cognitive psychology to the field of mental retardation (Ellis, 1970), they prescribed a whole new way of conceptualizing the mental activity of a person with mental retardation and pointed the way to a number of possible causes for memory deficiencies.

In the new conceptualization, memory is held to be comprised of two components, short-term memory (STM) and long-term memory (LTM). Short-term memory is limited in capacity and relatively brief in duration, i.e., approximately 30 seconds. Success in dialing a telephone number that a person has just looked up in a telephone book but the failure to recall it 30 minutes later is an example of the use of STM. Long-term memory, on the other hand, is considered to be of unlimited capacity and of permanent duration (Waugh & Norman 1965). Recalling the name of a favorite dog from childhood is an example of LTM.

The important task of transferring needed information from STM to LTM is primarily a function of active mental processing of that information. There are a number of voluntary rehearsal or encoding strategies that an efficient learner can employ to store the necessary information in LTM for later retrieval and use (Atkinson & Shiffrin, 1968, 1971). The more a person uses a cognitive strategy, the less mental effort it requires and the more automatic it becomes (Shiffrin & Schneider, 1977; Sternberg & Wagner, 1982).

While developing a repertoire of memory strategies, it is important for a learner to recognize the type of memory demands that a particular task presents, recognize the memory strategies (s)he has available, and choose the most appropriate one to use (Brown, 1978). These strategies about how to use one's memory strategies have been called "metamemory" or "metacognitive" skills. In essence, these are memory-management and logical-reasoning skills, and have been likened to an executive who makes decisions about how and when to use his resources (Butterfield & Belmont, 1977). Without metacognitive skills, a person who possesses memory strategies that would be adequate for successful performance on a particular task is passive and fails to employ them. Consequently, it has become increasingly recognized in instructional psychology that metacognitive skills are at least equally as important as memory skills in the cognitive functioning of successful learners (Brown, 1978; Flavell & Wellman, 1977).

Given this growing body of knowledge on efficient cognitive processing. it was logical for researchers in the field of mental retardation to heavily investigate these fundamental processes underlying successful performance. One such area to be investigated was the presence or absence of the use of metacognitive processes by mentally retarded persons (Butterfield & Belmont, 1977; Campione & Brown, 1977). Serious deficiencies are pervasive across the population of mildly and moderately retarded persons. Little scientific data exists in this area for severely and profoundly retarded persons, largely because of the lack of any identified reliable methodology by which such information can be obtained. The basic presumption, however, that memory deficiencies are much more pronounced in the more severely handicapped persons is starting to receive some empirical support (Ashman, 1983; Ellis, Deacon, Harris, Poor, Angers, Diorio, Watkins, Boyd, & Cavalier, 1982).

Prior Attempts to Solve the Problem

With the consensus that memory and metacognitive process deficiencies represented a critical problem for persons with mental retardation, interest became very intense in determining the extent to which they could be remediated. Extensive research attention turned towards developing an array of effective instructional techniques to impart to deficient information processors the rehearsal and metacognitive strategies of efficient information processors. The basic assumption underlying this research, and the work of this project, was that if basic process deficiencies exist and remain uncorrected, they will compound higher-level areas of functioning and frustrate instructional efforts. As a result of this new research, increasingly sophisticated techniques to identify the specific process deficiencies and then to remediate these deficiencies have been emerging (Belmont & Butterfield, 1977; Bray, 1979; Brown, 1978; Campione & Brown, 1977; Glidden, 1979; Hagen & Stanovich, 1977; Kramer & Engle, 1981).

Given the stunning success of many of these instructional techniques in training persons with mental retardation to significantly enhance their memory and learning performance through the use of cognitive strategies, research efforts have most recently been broadened to include the critical development of instructional techniques for the maintenance and generalization of the use of these cognitive strategies across time and situations (Belmont, Butterfield, & Borkowski, 1978; Belmont, Butterfield & Ferretti, 1982; Borkowski & Cavanaugh, 1979; Butterfield, 1981; Butterfield & Ferretti, in press).

The conclusions that must be drawn from the information presented above are that (a) a significant and pervasive problem in memory exists in the lives of persons with mental retardation, (b) these problems are the result of deficiencies in basic memory and metacognitive processes, (c) assessment techniques are available to identify the specific process deficiencies, and (d) instructional techniques are available to remediate those deficiencies.

Given the reliability and sophistication of the information on memory processes in mental retardation derived from the work of instructional and cognitive psychologists in the area, it is disconcerting how little influence this information has had on direct instructional techniques in special education for students with mental retardation (Pressley, Levin, & Bryant, 1983). While there has been some cross-over (e.g., Taylor & Turnure, 1979), the impact has been relatively small, considering the magnitude of the problem and the importance of this research for remediation. The factors accounting for this lack of transfer to special education are not definitively known, but they most likely include (a) the relatively short time that the information has been in the public domain, (b) the somewhat separate vehicles for professional communication in the two fields, e.g., different journals and conferences, and (c) the relatively unwieldly nature of much of the equipment and materials required by this assessment and instruction. This project will take major steps to both increase the speed with which the necessary cross-disciplinary dialogue is taking place and provide a powerful technology-based instructional aid for special education classroom use which circumvents the major technical problems associated with this type of assessment and instruction and capitalizes on the unique educational attributes of computers.

In recent years, special education professionals concerned with another handicapping condition, learning disabilities, have begun to investigate the applicability of the theories and techniques of cognitive psychology and mental retardation to learning disabled children and youth (Bauer, 1977; Cohen & Netley, 1981; Dawson, Hallahan, Reeve, & Ball, 1980; Lindgren & Richman, 1984; Swanson, 1983; Tarver, Hallahan, Cohen, & Kauffman, 1977; Torgesen, Murphy, & Ivey, 1979; Torgesen & Goldman, 1977). Torgesen (1977) provided direction for special educators in this area by positing in a theoretical paper based largely on the previous work of cognitive psychologists that most of the performance deficits of learning disabled children are based on either their inability to employ efficient, task-appropriate cognitive strategies or their lack of awareness that such strategic processing will be effective.

These contentions agreed with long-standing clinical observations by teachers and therapists in the field. In 1968, the National Advisory Committee on Handicapped Children of the U.S. Office of Education proposed a definition of learning disabilities which became part of the Learning Disabilities Act of 1969. Reference to deficiencies in basic psychological processes is the most prominent factor in this definition (Mercer, Forgnone, & Wolking, 1976). Clements (1966), in an extensive review of clinical literature, listed "disorders of memory and thinking" as one of the 10 most frequently mentioned symptoms of learning disabled children.

The initial research investigations that were stimulated by this common observation and Torgesen's (1977b) theorizing have confirmed that, while the information processing problems experienced by learning disabled children are complex, the knowledge gained in the area of cognitive psychology and mental retardation on memory and metacognitive process deficiencies has direct relevance to practically all major aspects of the assessment and remediation of these problems in learning disabled children (Bauer, 1979, 1982; Jacobs, 1983; Rose, Cundick, & Higbee, 1983; Torgesen, 1977a; Torgesen & Houck, 1980; Wong, Wong, & Foth, 1977). Consequently, given the extent of the information available today, we believe the conclusions that can be drawn on memory and metacognitive process deficiencies in learning disabilities are very similar to those in mental retardaiton: the deficiencies are serious and pervasive and direct instructional strategies are available for their remediation.

Nature of the Population to Benefit from the Aid

People with mental retardation are generally considered to be deficient in basic memory processes. Ellis (1970), in a major publication in the field of mental retardation, implies that differences in memory performance can be used to define retarded behavior. Within the years of research since 1970, a vast amount of empirical evidence has appeared that supports the general position espoused by Ellis. In their review of the literature, Borkowski and Cavanaugh (1979) concluded that deficits in effective memory strategies "are considered by many educators and theoreticians as the major problem characterizing the retarded, especially the educable mentally retarded" (pg. 569). At this point, it is plausible that <u>most if not all people with mental retardation</u> demonstrate a general deficit in memory functioning. Campione and Brown (1984) identified three types of memory deficiencies in the population. The first is that children with mental retardation fail to generate and use strategies that subjects of comparable age are likely to adopt spontaneously (Brown, Campione, Bray, & Wilcox, 1973; Campione & Brown, 1974). The second is that children with mental retardation need explicit instructions before they demonstrate strategic performance (Belmont & Butterfield, 1977; Butterfield, Wambold, & Belmont, 1973; Campione & Brown, 1977, 1978) and before they transfer strategy usage to new untrained situations (Borkowski & Cavanaugh, 1979; Brown, Campione, & Day, 1981; Campione, Brown & Ferrara, in press). The third is that children with mental retardation tend to cease employing a trained strategy when instruction is withdrawn; i.e., it does not become "their own".

Researchers studying cognition in persons with mental retardation have not systematically examined differences between the levels of severity of mental retardation. In the past, most research has utilized people from the mild and moderate ranges of retardation. Those people diagnosed as severely or profoundly retarded have been included in only a very few studies (Ashman, 1983; Ellis et.al., 1982).

Persons with severe and profound mental retardation are capable of acquiring skills in a variety of domains (Matson & McCartney, 1981; Mukherjie, 1977; Sailor & Guess, 1983). The question of whether or not the knowledge and skill deficits exhibited by persons with this degree of mental retardation are due to memory problems is still unanswered. It is possible, however, to infer some answers from the existing data on the relationship between I.Q. and memory.

One variable dependent upon I.Q. level is that of effective rehearsal utilization; there is a positive correlation between I.Q. and the occurrence of rehearsal strategies in recall (Brown, Campione, Bray, & Wilcox, 1973). By extrapolating from the data gathered primarily from comparisons between persons with mild to moderate mental retardation and nonretarded persons, it is very likely that the memory deficiencies exhibited by persons with severe mental retardation are due to low or nonexistent levels of rehearsal. This supposition has received support from Ellis et al. (1982) and Ashman (1983), whose findings suggest that persons with severe and profound mental retardation do seem to have serious memory deficiencies. Persons with severe and profound mental retardation have a higher incidence of maladaptive behaviors (Frankel & Simmons, 1976) and physical disabilities (O'Conner, Justice, & Payne, 1970) than do those with mild or moderate retardation. These considerations, as well as their more limited intellectual abilities, would most likely preclude most of this group's use of the aid under development.

Another factor that influences memory development is chronological age. Several researchers have noted developmental changes in memory (Campione & Brown, 1977; Hagan & Huntsman, 1971; Hagan & West, 1970). Studying children and adults with and without retardation, Belmont and Butterfield (1971) concluded that both active rehearsal and strategic nonrehearsal changed systematically with age. They noted that spontaneous rehearsal is a strategy that develops late (in early adolescence) in nonretarded persons and "might never be expected to occur in the mentally retarded" (p. 239).

A third factor influencing memory development may be socioeconomic status (SES). Mild mental retardation is disproportionately found among the economically disadvantaged and less well-educated segments of society (Ramey & Finkelstein, 1981). The report of a comparison between low SES children who had received early childhood education and those who had not revealed that differences in memory abilities were among the significant effects (Ramey & Campbell, 1979). This suggest that low SES may be implicated in the existence of memory deficiencies, and leads to the assumption that children of low SES generally display poorer memory functioning than do children of middle or upper SES.

Memory deficiencies in children and youth with learning disabilities have been found to be widespread. Those memory deficits found in persons with learning disabilities are now being recognized as similar to those observed among persons with mental retardation (Hagen, Barclay, & Schwethelm, 1982). The specific aspects of memory deficiencies in both populations are those that involve the acquisition and use of rehearsal strategies.

Metacognitive strategies play an important role in memory. Flavell (1979) presents metacognition as a guide to the selection of appropriate cognitive strategies for task performance. Cognitive strategies can be divided into two different types: control processes, which are specific strategies used to obtain a goal, (e.g., clustering); and executive functioning, which is the ability to select the appropriate control process to reach the goal (Atkinson & Shiffrin, 1968).

Persons with mental retardation exhibit a distinctive lack of executive functioning. Two obvious deficiencies become apparent. First, this population does not typically differentiate between those processes that require rehearsal and those that do not require rehearsal. Second, they do not coordinate retrieval strategies (Butterfield, Wambold, & Belmont, 1973). Both of these deficiences are indicative of deficiencies in metacognition.

Ineffective metacognitive strategies are also apparent among persons with learning disabilities. Torgesen (1979) observed that this population is less efficient in spontaneously utilizing strategies although they can use appropriate strategies when instructed to do so. In her review, Jacobs (1984) implies that poor performance "is a function of metacognition, their awareness of the possibility and need to use such strategies..." (pg. 215). In general, a common characteristic of people with learning disabilities is held to be that they are deficient in both cognitive and metacognitive strategies -- in both control and executive functions (Pearson & Spiro, 1980; Seidenburg, 1982).

Persons with mental retardation and those with learning disabilities have been shown to respond positively to instructional intervention. Training has been proven to be an effective means of increasing the use and effectiveness of rehearsal strategies. Performance of post-intervention handicapped groups similar to that of equal chronological age non-handicapped groups has been reported by Butterfield, Wambold, and Belmont (1973), Brown, Campione, Bray, and Wilcox (1973) and others. Campione and Brown (1977) concluded in their review of metamemory and memory that the evidence from effective training invalidates the "structural limitation" model; that is, performance levels are not completely determined by fixed limitations in the nature of the populations under study.

The population to benefit from this aid is defined not only in terms of demonstrated memory deficiencies but in terms of possession of the prerequisite skills necessary for interaction with the instructional package. Adequate visual acuity is essential. Since the package offers the option of text on the screen, a hearing-impaired person could interact with the system as long as his/her reading skills are adequate. If reading is inadequate, then hearing must be appropriate for speech discrimination at conversational intensity levels. Users of the instructional package must also know alphabet letters and the numbers 0 through 9, and must have motor abilities sufficient to allow them to control the input modes.

In summary, this aid will benefit the population of persons with memory difficulties who are able to interact with the computer system as it is configured. In general, this population would include school-aged children and youth with mild to moderate mental retardation as well as those with learning disabilities. Because of the nature of the task demands, most of the children and youth with severe and profound mental retardation would be precluded from using the system.

Size of the Population to Benefit from the Aid

The failure to use cognitive strategies to facilitate recall results in a general deficit in memory. Since intervention strategies designed to intervene in these areas appear to be effective, the number of people who could potentially benefit from programs of this nature is of interest. Under the general conclusion that most, if not all, persons with mental retardation and learning disabilities have fundamental problems in memory, it is possible to calculate the approximate number of people who could benefit.

The target population is composed of school-aged children and youth with learning disabilities or mild to moderate mental retardation. At least 90% of persons with mental retardation are classified in the category of mild to moderate retardation (Baroff, 1974; Tarjan, Wright, Eyman, & Keeren, 1973).

In 1981-82, 10.5% of the total elementary and secondary enrollment required special education and related services (U.S. Department of Education, 1983). According to the Fifth Annual Report to Congress on Public Law 94-142, approximately 4,233,282 students were utilizing these facilities. Nineteen percent were classified as mentally retarded and 38% as learning disabled; according to the proportion given above, the 19% figure translates into an estimate of 17% for those with mild to moderate mental retardation. Thus, 55% (that is, 2,328,315) of the students enrolled in special education could benefit from this computer-based intervention.

Description of the Aid

In this project, ARC/US will design, develop, field test, and refine a computer-based instructional system to assess and assist in remediating the serious and pervasive problems in memory and metacognition of mentally retarded and learning disabled children and youth. Software design will be guided by powerful and sophisticated instructional techniques which have been developed in the areas of cognitive psychology and mental retardation/learning disabilities and will fully exploit the strengths of the computer. Centrally involved in the preparation of this design will be one of the leaders in cognitive instructional design.

The proposed instructional system is not curriculum-specific but instead focuses on some of the fundamental cognitive skills which underlie learning and performance across every content area. The software is structured around a memory task frequently used in assessment and instructional applications, which requires many of the same cognitive strategies for successful performance that underlie efficient information processing across a wide variety of situations (Latham, 1978). Field testing will take place in the natural classroom environment with typical mentally retarded and learning disabled students. Since memory and metacognitive process deficiencies are considered to characterize the large majority of such students and since the software provides graphic as well as vocal cues, the proposed instructional system should apply to all mildly and moderately mentally retarded students and severely learning disabled students.

With such a system, a teacher will be able to assess whether a student has significant memory process deficiencies, identify the nature of the deficiencies, and provide him/her individualized instruction on efficient memory and metacognitive processing strategies. Ultimately, this technologically-based aid will provide teachers a powerful means by which they can begin to remediate serious and pervasive cognitive problems encountered in the education of their mentally retarded and learning disabled students.

The following section provides a more detailed description of the memory task embedded in the instructional system. Appendix A contains general program information and Appendix B contains more detailed information for the programmer that further defines the workings of the system. Appendix C is the complete program narrative, which contains the instructions to the computer programmer regarding the use of screen layouts, vocal and orthographic text, and activity on the screen. Appendix D contains the screen layouts, Appendix E contains a description of the video-game interlude, Appendix F is the software evaluation form that was sent to program evaluators, and Appendix G is the informed consent letter that will be signed by participants in the field testing.

The instructional system will be programmed for use on two of the most widely-used computers in public school systems and homes: the Apple II series of computers and the Commodore 64. For all intents and purposes, the two programs are functionally the same, except for minor differences dictated by the hardware.

The instructional system being developed in this project is based upon the ordered recall task. The ordered recall task has been one of the vehicles used in establishing the general fund of knowledge in this area. It is steeped in a well-developed theory which yields systematic and reliable predictions; it is sensitive to the influence of strategic cognitive processes thereby distinguishing between sophisticated and ineffective information processors; it permits variation on a number of student and task-related factors which have direct educational impact; it yields orderly data; it permits both assessment of a student's mnemonic competence and instruction on efficient, effective, and generalizable cognitive strategies; and it is relatively easy to explain to students (Belmont, Ferretti, & Mitchell, 1982; Butterfield, Siladi, & Belmont, 1980; Butterfield, Wambold, & Belmont, 1973; Brown & Barclay, 1976).

This array of positive attributes that the ordered recall task has for cognitive and special education research notwithstanding, it has not received much use in special education applications. The primary reasons have been the unwieldy nature of the equipment typically used to administer the task, the mathematical complexity required to derive meaningful assessment information, and the individualization that this information dictates for the subsequent instructional components. These very reasons place the computer as the most logical medium for assessing and training ordered memory skills of mentally retarded and learning disabled students, for the computer has unique strengths in each of the problematic areas mentioned above. In addition, the computer has many other attributes which make it the near perfect choice as the medium by which the logic and memory of these students is assessed and trained.

In an ordered recall task, the student is requested to recall in the order presented a list of items that (s)he has seen only once. The items are serially-presented, with only one item exposed at any one time. The student is asked to first recall the subset of the last items presented and then circle back and recall the subset of the remaining items which were presented first. This aspect of the task is called "circular recall" (Butterfield, Siladi, & Belmont, 1980).

For example, if there are seven items presented serially and the student is in a "3/4 circular recall" task, (s)he would attempt to recall the last 3 items first, followed by the first 4. Thus, for the list Q,P,X,J,N,B,T, the correct 3/4 circular recall would be N,B,T,Q,P,X,J. If there are 8 items presented, for which the last 3 are to be recalled prior to the first 5, the student is in "3/5 circular recall" task.

In the ordered recall task, each item is displayed for a fixed period of time (e.g., 0.5 seconds), but the student controls the pace of the presentation (i.e., the timing of the presentation of the next item). Difficulty of the task and memory load requirements can be varied by changing the number of items in the to-be-recalled list and the type of items to be recalled (e.g., letters, numbers, words).

Extensive research has shown that performance on the terminal items (N,B,T in the example above) reflects a student's STM abilities, and the task permits precise manipulation of variables which pertain specifically to STM limitations and instructional strategies. Similarly, performance on the initial terms (Q,P,X,J in the example above) reflects a student's LTM abilities and is sensitive to a number of manipulations directly related to strategic cognitive activity (Belmont & Butterfield, 1969, 1971a, 1971b; Brown & Barclay, 1976).

One of the most revealing measures of strategic cognitive activity in this task is the length of time the student <u>pauses</u> after the presentation of each item in the list. Non-retarded, non-learning-disabled students generally exhibit high recall accuracy when their pauses steadily increase across the initial items, followed by very brief pausing over the terminal items.

This pause pattern reflects an effective memory strategy for this and many other tasks: active cumulative rehearsal during the pauses of the initial items, i.e., those most difficult to recall in an ordered list, followed by fast passive glancing at the terminal items (Belmont & Butterfield, 1969, 1971a; Butterfield, Siladi, & Belmont, 1980).

In the example presented above, a student using this "cumulative rehearsal-fast finish" strategy would mentally repeat the "Q" to him/herself after it was presented, then repeat "Q-P" a few times after the P was presented, then repeat "Q-P-X" a greater number of times after the X was displayed, followed by "Q-P-X-J" after the J was displayed. An efficient information processor would mentally test him/herself on his/her ability to successfully recall these initial items before proceeding to the terminal items, thereby accounting for the longest pause after the fourth item in the list. This self-monitoring has been variously labeled as "executive control", "metamemory", and "metacognition", and is held to be one of the definitive attributes of intelligence (Butterfield & Belmont, 1975; Brown, 1975, 1978; Campione & Brown, 1977; Flavell, 1971; Flavell & Wellman, 1977).

To continue with the example, after achieving a satisfactory level of retention on his/her self-testing, the student would proceed to the terminal items and expose the N, briefly glance at it, expose the B, briefly glance at it, expose the T, glance at it and then proceed to the actual recall test. The cognitive strategy described above conforms to well-established theory in the information processing realm, and consequently is viewed as the theoretically ideal cognitive solution for ordered recall (Atkinson & Shiffrin, 1968; Waugh & Norman, 1965).

When a student is found to be developmentally young (Brown & Campione, 1974) in his/her approach to ordered recall tasks, i.e., (s)he uses a less-than-optimal cognitive strategy, cognitive psychologists and special education researchers engage in an instructional sequence designed to impart the basic components of the ideal solution to the student (Belmont & Butterfield, 1977; Brown & Barclay, 1976; Butterfield, Siladi, & Belmont, 1980; Butterfield, Wambold, & Belmont, 1973). The remedial strategy focuses training on learning the terminal items, secondly on encoding the initial items, thirdly on retrieval of the initial items and self-checking, and finally on coordination of all strategic components.

The first component of the instructional sequences involves training the student to pace quickly through the terminal items. The second component has the student cumulatively rehearsing the growing list of initial items as each one is presented. Integrated into this component is the practice of self-checking, in which the student mentally tests himself/herself to be certain of his/her accurate retrieval of the subset prior to exposing the next item in the list. The third component is the introduction of a delay between the last item seen by the child and the beginning of his recall attempt. This delay is to insure that in practice the initial items are successfully recalled from LTM only, and also to enhance the student's understanding of the necessity for active rehearsal of the initial items. The fourth component instructs the student to put all of these cognitive strategies together and provides practice on the smooth coordination of the strategies.

During the instructional sequence, a student is typically trained and brought to a criterion on one circular recall requirement, e.g., 3/4. Transfer of the cognitive strategy can then be tested upon the student's first encounter with a different circular recall requirement, e.g., 3/5.

The instructional system under development will bring the unique attributes of the computer to bear on the task demands for ordered recall and the instructional demands for assessing and assisting in remediating cognitive deficiencies. The end product will be an easy-to-use, informative, and powerful instructional tool for special educators.

Explanation of How the Aid Assists in Solving the Problem

Simultaneous with these exciting trends in cognitive psychology, mental retardation, and learning disabilities have been the revolutionary developments in microcomputer technology and their resultant applications in rehabilitation, education, and special education. The general opinion is that we are standing on the threshold of a new era in improvements to the quality of life of handicapped persons. Much of the basis for this optimistic view has been provided by research supported by SEP which has shown that "in instances where technology is made available, is of good quality, and is used knowledgeably, it has enormous potential for improving the education, independence, and employability" of persons across a wide variety of handicapping conditions (Johnson & Kaufman, 1983).

The latest market projections provide an image of the magnitude of the computer movement in special education. Vest (1983) states that the special education market had an estimated spending segment of approximately $10 billion in 1983, up from about $4.6 billion in 1976. Translated to hardware, approximately 150,000 microcomputers were in the public schools with approximately 25,000 used primarily for special education. Of these special education units, 10,000 were used in administration and 15,000 for instruction. By 1985-86, approximately 500,000 microcomputers will be in public schools, 150,000 of which will be used primarily for special education. Of these, approximately 20,000 will be used for administrative purposes (Blaschke, 1983).

While the benefits of assistive devices have been substantial and the magnitude of the movement is impressive, only approximately 10% of the school-aged handicapped population actually receive these benefits. They are children and youth with visual impairments, hearing impairments, mobility impairments, and multiple handicaps. The 90% who are, for the most part, unserved by these powerful new tools consist of children who are mentally retarded, learning disabled, speech impaired, or emotionally disturbed (Report to Congress, 1981).

The Office of Special Education Programs clearly recognize both the extreme potential of computers for remediating problems in the education of mentally retarded and learning disabled children and the disproportionate effort that professionals in the area have devoted to persons with other handicapping conditions, thereby leaving this potential largely unrealized. With the unique strengths that a computer offers to education, particularly in the areas of logic, memory, and motivation, it is encumbent upon educators to begin to employ computer-based aids to solve or minimize the serious problems in logic and memory which characterize mentally retarded and learning disabled students.

Given the availability of computers for educational purposes, the challenge now is to design instructional packages that will capitalize not only on the computer's unique capabilities but also on the most sound and effective educational practices. A review of the cognitive psychology and special education literatures has resulted in the consolidation of several recommendations for the enhancement of learning. None of these recommendations were made specifically in regard to computer-assisted instruction, yet in many cases these recommendations can best be met by computer-based instructional systems.

Lieberman (1982) made several suggestions concerned with the optimization of learning that have been voiced by other researchers and educators. The first of these suggestions is to incorporate relatively novel presentations of information; this will attract attention to the task. The medium should be structured so that the teacher's messages are "open to the learner's inspection", meaning that objectives should be stated, relationships highlighted, and help given through the use of cues and prompts. It is important that the child have the prerequisites necessitated by the task; this requires that the task be analyzed and the component parts be taught in an appropriate sequence. A child should have access to a model of correct performance and should have an opportunity to imitate the model. Multisensory demonstrations are helpful, as is the active engagement of the student in practice. The student's reliance on prompts can then be gradually withdrawn. Additionally, it is vital that learning conditions be pleasant, avoiding unreasonable demands, and providing challenges, immediate feedback and rewards. Similar recommendations for training have been made by Borkowski and Cavanaugh (1979), Brown (1978), Carter (1984), Dawson, Hallahan, Reeve, and Ball (1980), Lewis (1983) and Sheinker, Sheinker, and Stevens (1984).

The computer-based instructional system under development combines oft-proven training techniques with the unique capabilities of the microcomputer. The circular recall paradigm has been used in previous research with persons with mental retardaton for the purposes of assessment and remediation (e.g., Belmont and Butterfield, 1971; Belmont, Butterfield, & Borkowski, 1978; Belmont, Ferretti, & Mitchell, 1982). This computer-based system breaks the circular recall memory strategy into its component parts, trains each separately, and then chains the components together. The students will work with a number of different list lengths and circular recall requirements; these variations on the same general task should increase the students' understanding of the basic strategy because the students are able to witness and participate in its application in a number of situations. The system also provides the student with additional practice in those areas in which (s)he is experiencing difficulty.

One of the problems inherent in previous studies employing circular recall was the unwieldy nature of the apparatus, which typically included several projectors, switches, and a viewing panel. An obvious benefit to the use of a microcomputer is that the necessary hardware is already located in many schools and homes, and that the software is portable.

Project Timeline

As described in the Administrative Report filed with SEP for December, 1984, a revised staff resource plan was proposed to accomplish some project objectives at a faster pace than was orginally planned. This plan is intended to recover some of the time lost due to some unusual delays in initiating the project at full staff capacity. This revised staff resource plan permits project tasks and activities to be accomplished according to the timeline on the following page.

It is probable that it will be beneficial to re-instate the increases in percentages of FTE for project staff that comprise the revised staff resource plan for January-April again in August and in January, 1986 in order to complete the tasks at a faster pace than was originally scheduled. This strategy of increasing the percentage of time devoted to the project would result in no increased cost to the federal government.

Timeline

Tasks & Activities

	Task	Jan.	Feb.	March	April	May	June	July	Aug.	Sept.	Oct.	Nov.	Dec.	Jan.	Feb.
301	Substantiation of the Useability and Design of the Instruction Systems														
301.1	Completion of Substantiation Report	→	→												
301.2	Narrative Description of Programmatic Content - Programs A & B	→													
301.3	Solicitation and Incorporation of Feedback - Programs A & B	→	→												
301.4	Completion of Final Design Report			→											
302	Creation of the Instructional Systems														
302.1	Completion of Computer Programming Program A		→	→	→	→									
302.2	Completion of Computer Programming Program B						→	→							
302.3	Alpha Testing and Refinement Program A					→									
302.4	Alpha Testing and Refinement Program B								→						
302.5	Preparation of Program Documentation (Manual) - Programs A & B									→					
302.6	Submission of Aid - Description Document, Program Manual, & Test-Ready Aids									→					
303	Field Testing and Refinement														
303.1	Completion of Field Test Plan Programs A & B								→	→					
303.2	Inservice Training - Programs A & B									→					
303.3	Evaluation with Students and Teachers - Systems A & B										→	→			
303.4	Analysis of Field-Test Data Systems A & B												→		
303.5	Final Refinement of Hardware and Software - Systems A & B													→	→
303.6	Submission of Field-Test Report Systems A & B														→
304	Preparation of Marketing Plan and Recommendations														
304.1	Submission of Preliminary Marketing Plan												→		
304.2	Submission of Market Plan for Evaluation by at least two Potential Producers/Dist.													→	→
304.3	Submission of Final Marketing Plan Incorporating Feedback from the Field													→	→

Components and Costs

The computer-based instructional systems that will be developed in this contract are comprised of off-the-shelf computer components and peripherals, the cognitive software that is being designed and programmed, and the software documentation that is being developed. The design of the systems incorporates some specific computer I/O components and peripherals that provide powe. .l educational features and that promise to be increasingly incorporated in educational computer applications in the future, i.e., light pens, joysticks, and speech synthesizers/digitizers.

Other than the specific I/O components mentioned above, the Commodore and Apple computer systems upon which the project's instructional systems are based were intended not to differ from the configuration of the systems typically found in public schools. That is, to insure the widest use of the cognitive software developed in the project, idiosyncratic hardware requirements were eliminated.

The fundamental Apple based computer system required by the Apple version of the software being developed consists of an Apple II series computer with a minimum of 48K of RAM, two floppy disk drives, a color monitor, a parallel printer interface, a parallel printer, a light pen, a potentiometer-type joystick and a speech synthesizer. The fundamental Commodore 64-based computer system required by the Commodore version of the software being developed consists of a Commodore 64 computer, one or two floppy disk drives, a color monitor, a parallel printer interface, a parallel printer, a light pen, a switch-type joystick, and a speech synthesizer. Attached is a copy of a listing of the system components and costs submitted by a local computer vendor. In this project, two Apple-based systems and two Commodore systems will be developed.

Project staff have begun to be inundated with inquiries about the project from teachers and parents around the country, many of which include requests for recommendations on "the best" light pen, synthesizer, monitor, etc., to purchase. Given these requests for guidance, it may be beneficial for the project to compare alternate versions of these components within the two Apple systems and the two Commodore systems. This should result in no additional cost to the federal government, as these items can be compared in the normal course of the project.

Computerose

May 16, 1984

Dr. Al Cavalier
Association for Retarded Citizens of U.S.
2501 Avenue "J"
Arlington, Texas 76011

Dear Dr. Cavalier:

The following quote is submitted in response to your request for bids for Apple and Commodore computers as well as peripheral equipment.

QUANTITY	ITEM	UNIT COST	EXTENDED COST
2	Commodore 64 Computer	$ 219.00	$ 438.00
4	Commodore 1541 Disk Drive	249.00	996.00
2	Commodore 1702 Color Monitor	249.00	498.00
2	Apple IIe Starter System	1,095.00	2,190.00
2	Apple Disk Drive	395.00	790.00
4	C.Itoh Prowriter 10" Printer	399.95	1,599.80
2	Amdek I Color Monitor	287.00	574.00
2	Symtec Light Pen (Commodore)	175.00	350.00
2	Symtec Light Pen (Apple)	250.00	500.00
2	ComVoice Speech Synthesizer	139.95	279.90
2	Echo II Speech Synthesizer	139.95	279.90
2	Newport Prostick II Joystick	45.00	90.00
2	Kraft Joystick	49.00	98.00
2	Cardco Graphics Interface (P)	99.95	199.90
2	Grappler + Interface (P)	119.00	238.00
	Total		$9,121.50

Thank you for according us the opportunity to assist you in meeting your computer needs.

Sincerely,

Sam Barklis

Sam Barklis
Chief Executive Officer

Labor Distribution

Dr. Al Cavalier, Project Director, has overall responsibility for achievement of the project's objectives. He will be centrally involved in the instructional systems design, evaluation and refinement as well as the financial administration of the project.

Dr. Beth Mineo, Assistant Project Director, is responsible for assisting Dr. Cavalier in the implementation of the project and the system's design, evaluation and refinement, with primary responsibility for coordinating the day-to-day operation of the project.

Dr. Ralph Ferretti, Program Design and Research Consultant, has primary responsibility for instructional program design in accordance with current cognitive theory and research and shares responsibility with Dr. Cavalier and Dr. Mineo for analysis and interpretation of the evaluative data and subsequent refinement of the program design.

A computer programmer with extensive experience in developing and documenting software for popular microcomputers, proficiency in assembly language programming for the 6502 family of micro-processors and specific skills in animated graphics and file management will devise the object codes, source codes and documentation for the instructional systems according to the specifications in the program narrative developed by project staff and consultants.

Consultants and reviewers with recognized expertise in a variety of relevant areas such as cognitive psychology, special education and computer science will provide feedback on the program narrative prior to its translation into a computer program and on the evaluative data obtained from the Beta tests.

Mr. Robert Dubin and Ms. Nancy Sullivan, Marketing Specialists, have primary responsibility for the development and implementation of the ARC/US marketing plan and negotiation with independent software publishers.

Ms. Cindy Oliver, Project Secretary, is responsible for all secretarial and clerical support to the project staff.

Project responsiblities in person-days for each major project task are as follows:

Personal Assignment by Person Days

	Task 1	Task 2	Task 3	Task 4	Task 5	Total
Project Director	14	40	18	9	36	117
Asst. Proj. Director	32	98	43	22		195
Proj. Design Conslt.	13	40	18			71
Computer Prog.		195	44			239
Marketing Spec.				33		33
Project Consultants	8		4			12
Project Secretary	26	80	35	18	36	195
TOTAL	93	453	162	82	72	862

Plans for Testing

Alpha Tests

Purpose:

* To review the first version of the software as an integrated whole, and identify any aspects of the system in need of refinement;
* to examine the ways in which subjects interact with the software;
* to derive average pause-time data;
* to gauge the speed with which students complete the package;
* to evaluate selection modes in terms of ease of use and subject preference;
* to evaluate the game interlude for appropriate timing, difficulty levels and motivational qualities.

Subjects: Eight staff members will take the role of student and run through the program attempting to provide the widest diversity of interactive responses. Five children in the non-retarded range of intelligence and a few children with mental retardation/learning disabilities will also interact with the system.

Procedure: In addition to the program instuctions, the subjects will be told that they are assisting in the development of some new software and that they should do their best at the tasks. The subjects will be asked subjective questions regarding their experiences with the computer and the software.

Beta Tests

Purpose:

* To determine if the computerized version of the assessment/instruction procedure yields data similar to those derived from previous laboratory and classroom research;
* to ascertain the validity of the assessment; that is, does it identify memory problems;
* to ascertain the nature of the memory problems that individual students with learning disabilities and mental retardation have and any differences between groups;
* to determine if the instructional techniques employed can assist in remediating the rehearsal deficiencies identified in the assessment, i.e., can a foundation of computerized remediation strategies begin to be laid down; and
* to determine if students generalize the use of strategies trained directly to instances for which they have receive no training.

Subjects: Approximately 60 students matched for chronological age will serve as subjects. Twenty subjects with mental retardation, twenty with learning disabilities, and twenty nonhandicapped students will participate. Half of the subjects in each group will serve as controls while half will receive intervention with the computer-base instructional package.

Procedure: Pre- and Post-Test

All subjects will receive training to familiarize them with the computer, after which a computer-based pre-test will be administered. The subjects will receive six trials for each of 12 different circular recall requirements in which they will be shown items in a list and be asked to recall the list. The last three of these trials will be used in the data compilation. During this pre-test, no subjects will receive strategy training. This procedure will be the same for the post-test that follows the training.

Training

Half of the subjects in each subject classification will receive the computer-assisted instruction. The other half will receive no intervention between pre- and post-tests. A total of six circular recall requirements will be addressed in training.

Experimental Design and Data Analysis

There will be several factors or independent variables addressed in the experimental design and data analysis: subject classification (learning disabilities, mental retardation, non-handicapped), instructional level (training, control), test (pre, post), serial position, and circular recall requirement (e.g., 3/2, 4/3). The dependent measures are a processing measure (as reflected by the omega^2 value) or an accuracy measure (as reflected by percent correct).

Three types of group analyses will be conducted:

The first is to establish a relationship between the measures of processing and recall accuracy; theoretically, if a subject revises his processing to match that taught in the instructional package, his recall accuracy should improve. This correlation will be computed for every recall requirement.

The second type is an aggregate analysis of variance of the pre- and post-test measures of recall accuracy and processing as a function of subject classification and instructional effect. Again, these will be computed for every circular recall requirement.

It may happen that a subject's recall accuracy would be satisfactory without his/her use of the strategies that were instructed. To determine this, a third type of analysis looking at serial position in relation to the factors above would be necessary. This would entail a four-way analysis of variance for subject classification, pre-/post-test, instructional effect, and serial position (the specific number of serial positions is dependent on circular recall requirement). This analysis of variance would be computed for every circular recall requirement.

Depending on the nature of the accumulated data, it may be appropriate to do some analyses of individual subject's data in terms of the relationship between specific circular recall requirements and the processing and/or recall accuracy data. Since training will be conducted on only half of the circular recall requirements assessed in the pre- and post-test, performance on the untrained circular recall requirements will serve as an index of generalization of strategy use to similar but different tasks. It would be premature at this juncture to assess generalization to less similar ones, i.e., those that require the same basic underlying strategies but have different surface structure. If generalization is evident, future research efforts should explore the extent to which it occurs and the conditions that optimize its occurrence.

Channels of Distribution

Once the development, field testing and refinement stages have been completed, the instructional package of software and documentation will be ready to fulfill its primary purpose, which is assisting teachers in the assessment and remediation of memory deficiencies. Vital to the attainment of this end are effective channels for distribution of the product.

The most likely distributor would be a software publisher with an established reputation in the education market. Since there are literally hundreds of software companies, the pool of relevant potential marketers would be comprised of those offering a product line consistent with the offering of the ARC/US project. Three types of product lines potentially offer this compatibility: those with regular educational software, those with software designed for special needs populations, and those with software designed specifically around cognitive tasks. These three product lines are not mutually exclusive; in fact, from our compilation of information on software publishers, there are a few companies promoting software appropriate to all three categories.

The responsibility of ARC/US in the distribution process is six-fold. The first responsibility is the specification of the capabilities of the instructional system. The second is the identification of the target population. The third is to establish and document the need for software of this type among the target population. The fourth responsibility is to highlight the features and capabilities of the system. The fifth is to identify, based on the target population and needs assessments, marketing strategies that a software publisher could employ to increase networks of information dissemination.

Through the field-testing process, ARC/US will obtain the information necessary to meet the first of these outlined responsibilities. We have devised a detailed plan for testing in which participants will be comprised of persons with mental retardation, those with learning disabilities, and nonhandicapped persons. The instructional package will be evaluated in terms of its validity (that is, its ability to differentiate among ability groups) and its instructional value (that is, its ability to assist in the remediation process). This research will allow us to draw conclusions regarding the capabilities of the software for assessment and instructional purposes.

This research will also allow us to determine the breadth of effective application of the package across the populations in need. Although the appropriateness of this software for the population of nonhandicapped students is not the primary focus of the development project, a determination of such appropriateness is incorporated in the research design to provide a base of comparison for the handicapped users. As a result, there is a distinct possibility that the results will show that the larger market of non-handicapped students could derive enhancements in memory fuctioning through use of this software. Project staff conducted an extensive review of the cognitive psychology and special education literatures to identify the characteristics and nature of the populations to benefit from the aid. We have determined that approximately 55% of the students enrolled in special education classrooms in this country could benefit from this instructional package. This figure represents a market of close to two and one-half million students. More detailed information will be provided to the eventual marketer.

ARC/US's extensive literature review also assisted in establishing the need for this type of software. One of the conclusions that can be drawn from this review is that the memory task around which the package is constructed is a valid and "pure" means by which to assess and train memory skills, and also that the computer is a near-perfect vehicle for this package because of its ability for logical analysis and its capacity to deal with large amounts of information in an interesting, effective, and efficient manner. This package accomplishes the marriage of a theoretical knowledge base to a practical, educationally-sound assessment and training package.

ARC/US has the responsibility for highlighting the capabilities and positive attributes of the system to potential marketers and eventually to consumers. Detailed description of there would be lengthy; only the major points will therefore be summarized as follows. First, the package offers both assessment and remedial components, and remediation is based logically on the assessment results. This assessment permits the remediation components to be individually tailored to each student's needs. Second, the package uses the unique features of the computer to their fullest extent in assisting the child to understand and perform the required tasks; and third, the child's performance is analyzed and interpreted by the computer, which allows the teacher to obtain information practical for classroom purposes. This analysis and interpretation is guided by the consultation of the leading cognitive psychologists in the country.

The student's performance data is also permanently recorded for later review by the teacher.

In meeting the fifth responsibility, ARC/US will suggest marketing strategies for use by a potential publisher based upon the factors addressed above. We will assist the publisher in highlighting this program's appeal and value to parents, teachers, and school districts.

ARC/US's final responsibility to the distributor will be met through its ability to disseminate information across a nationwide network. Through our network of 1600 state and local affiliates, our core of 200,000 members, our national publications including our national newspaper which is distributed five times a year to each member, computerized data base, national electronic mail and bulletin board system, and our Bioengineering Program, we are in a unique position to raise the awareness of school personnel on the availability of quality educational software in the marketplace.

By the end of the sixteenth month of the project, a preliminary marketing plan will be developed and submitted to the contracting officer. The final marketing plan will take into account the views of the users, potential users, project consultants, and potential marketers in addressing such specific factors as appropriate unit-price, potential for widespread use, the value to the target population in assessing and remediating memory deficiencies and to improve learning capacity, as well as the interest among potential producers and distributors to market the compensatory educational aid that is developed.

Feedback from Commercial Publishers

In compliance with SEP's request for feedback on the marketability, useability, and suitability of the product, ARC/US identified several commercial software companies having product lines compatible with the software under development in this project. The company presidents and/or product developers were contacted and their participation was requested. Several companies denied our request, citing most frequently the non-remunerative or time-consuming aspects of the task. Non-disclosure agreements were obtained from three marketers who agreed to participate.

These companies were sent an information packet including a statement of the problem addressed in the project, production and marketing plans, and the program narrative with accompanying documentation. The company representatives were guided in their review by the survey form created by ARC/US project staff. The form was intended to direct the reviewers' comments to the specific aspects of suitability, useability, and marketability (see Appendix F).

The review period that is typically observed in the software industry is 30 days or longer. As of this writing, only one software publisher has returned the completed evaluation form. Telephone, written, and personal contact with the remaining publishers has failed to result in return of their evaluations, although both have acknowledged the time constraints under which they have been placed and have given their personel assurances that feedback will be returned shortly. Project staff have approached additional publishers to serve as evaluators should the original evaluators fail to return comments by June 30, 1985.

Project staff will summarize all evaluative remarks in an addendum to the Final Design Report will be forwarded to the Project Officer by July 15, 1985.

REFERENCES

Alabiso, F. (1972). Inhibitor functions of attention in reducing hyperactive behavior. American Journal of Mental Deficiency, 77, 259-282.

Ashman, A. (1983). On the theory and practice of remediating coding deficiencies. Australia and New Zealand Journal of Developmental Disabilities, 9(1), 8-15.

Ashman, A. (1982). Coding, strategic behavior and language performance of institutionalized young adults. American Journal of Mental Deficiency, 86, 627-636.

Ashman, A., Molloy, G., & Das, J. (1981). Coding, planning and mental retardation: Theory, evidence and implications. Part II. Australian Journal of Developmental Disabilities, 1, 57-64.

Atkinson, R.C., & Shiffrin, R.M. (1971). The control processes of short-term memory. Scientific American, 225, 82-90.

Atkinson, R.C., & Shiffrin, R.M. (1968). Human memory: A proposed system and its control processes. In K.W. Spence & J.T. Spence (Eds.), The psychology of learning and motivation. (Vol. 2). New York: Academic Press.

Baroff, G. (1974). Mental retardation: Nature, cause, and management. Washington, DC: Hemisphere Publishing Corporation.

Bauer, R.H. (1982). Information processing as a way of understanding and diagnosing learning disabilities. Topics In Learning & Learning Disabilities, 3(2), 33-45.

Bauer, R.H. (1979). Memory, acquisition, and category clustering in learning-disabled children. Journal of Experimental Child Psychology, 27, 365-383.

Bauer, R.H. (1977). Memory processes in children with learning disabilities: Evidence for deficient rehearsal. Journal of Experimental Child Psychology, 24, 415-430.

Belmont, J.M. (1972). Relations of age and intelligence to short-term color memory. Child Development, 43, 19-29.

Belmont, J.M. (1967). Perceptual short-term memory in children, retardates and adults. Journal of Experimental Child Psychology, 5, 114-122.

Belmont, J.M. (1966). Long-term memory in mental retardation. In N.R. Ellis (Ed.), International review of research in mental retardation. (Vol. 1). New York: Academic Press.

Belmont, J.M., & Butterfield, E.C. (1969). The relations of short-term memory to development and intelligence. In L. Lipsitt & H. Reese (Eds.), Advances in child development and behavior. (Vol. 4.) New York: Academic Press.

Belmont, J.M., & Butterfield, E.C. (1971a). What the development of short term memory is. Human Development, 14, 236-248.

Belmont, J.M., & Butterfield, E.C. (1971b). Learning strategies as determinants of memory deficiencies. Cognitive Psychology, 2, 411-420.

Belmont, J.M., & Butterfield, E.C. (1977). The instructional approach to developmental cognitive research. In R.V. Kail, Jr. & J.N. Hagen (Eds.), Perspectives on the development of memory and cognition, (pp.437-481). Hillsdale, NJ: Lawrence Erlbaum Associates.

Belmont, J.M., Butterfield, E.C., & Borkowski, J.G. (1978). Training retarded people to generalize memorization methods across memory tasks. In M.M. Gruneberg, P.E. Morris, & R.N. Sykes (Eds.), Practical aspects of memory. London: Academic Press.

Belmont, J.M., Butterfield, E.C., & Ferretti, R.P. (1982). To secure transfer of training instruct self-management skills. In D.K. Detterman & R.J. Sternberg (Eds.), How and how much can intelligence be increased. Norwood, NJ: Ablex Publishing.

Belmont, J.M., Ferretti, R.P., & Mitchell, D.W. (1982). Memorizing: A test of untrained mildly mentally retarded children's problem solving. American Journal of Mental Deficiency, 87(2), 197-210.

Bilsky, L., & Evans, R.A. (1970). Use of associative clustering techniques in the study of reading disability: Effects of list organization. American Journal of Mental Deficiency, 74, 771-776.

Blaschke, C.L. (1982). Microcomputers in special education: Trends and projections - 1983. Journal of Special Education Technology.

Boersma, F., & Muir, W. (1975). Eye movements and information processing in mentally retarded children. Rotterdam: University Press.

Borkowski, J.G., & Cavanaugh, J.C. (1979). Maintenance and generalization of skills and strategies by the retarded. In N.R. Ellis (Ed.), Handbook of mental deficiency: Psychological theory and research. (2nd Ed.). Hillsdale, NJ: Lawrence Erlbaum Associates.

Bray, N.W. (1979). Strategy production in the retarded. In N.R. Ellis (Ed.), Handbook of mental deficiency: Psychological theory and research. (2nd Ed.). Hillsdale, NJ: Lawrence Erlbaum Associates.

Brooks, P., & McCauley, C. (1984). Cognitive research and mental retardation. American Journal of Mental Deficiency, 88(5), 479-486.

Brown, A.L. (1975). The development of memory: Knowing, knowing about knowing, and knowing how to know. In H.W. Reese (Ed.), Advances in child development and behavior. (Vol. 10). New York: Academic Press.

Brown, A.L. (1978). Knowing when, where, and how to remember: A problem of metacognition. In R. Glaser (Ed.), Advances in intructional psychology. (Vol. 1.). Hillsdale, NJ: Lawrence Erlbaum Associates.

Brown, A.L. (1972). Context and recency cues in the recognition memory of retarded children and adolescents. American Journal of Mental Deficiency, 77, 54-58.

Brown, A.L., & Barclay, C.R. (1976). The effects of training specific mnemonics on the metamnemonic efficiency of retarded children. Child Development, 47, 71-80.

Brown, A.L., Campione, J.C., Bray, N.W., & Wilcox, B.L. (1973). Keeping track of changing variables: Effects of rehearsal training and rehearsal prevention in normal and retarded adolescents. Journal of Experimental Psychology, 101, 123-131.

Brown, A.L., Campione, J.C., & Day, J.D. (1981). Learning to learn: On training students to learn from texts. Educational Researcher, 10, 14-21.

Butterfield, E.C. (1981). Instructional techniques that produce generalized improvements in cognition. In P.E. Mittler (Ed.), Frontiers of knowledge in mental retardation: Proceedings of the 5th Congress of IASSMD. (Vol. 1). Baltimore: University Park Press.

Buterfield, E.C., & Belmont, J.M. (1975). Assessing and improving the executive cognitive functions of mentally retarded people. In I. Bialer & M. Sternlicht (Eds.), Psychological issues in mental retardation (pp. 277-318). New York: Psychological Dimensions.

Butterfield, E.C., & Belmont, J.M. (1977). Assessing and improving the cognition of mentally retarded people. In I. Bialer & M. Sternlicht (Eds.), Psychology of mental retardation: Issues and approaches. New York: Psychological Dimensions.

Butterfield, E.C., & Ferretti, R.P. (in press). Some extensions of the instructional approach to the study of cognitive development and sufficient condition for transfer of training. In P.H. Brooks, C. McCauley, & R. Sperber (Eds.), Learning and cognition in the mentally retarded. Hillsdale, NJ: Lawrence Erlbaum Associates.

Butterfield, E.C., Siladi, D., & Belmont, J.M. (1980). Validating theories of intelligence. In H.W. Reese & L.P. Lipsett (Eds.), Advances in child development and behavior. (Vol. 15). New York: Academic Press.

Butterfield, E.C., Wambold, C., & Belmont, J.M. (1973). On the theory and practice of improving short-term memory. American Journal of Mental Deficiency, 77, 654-669.

Campione, J., & Brown, A. (1978). Toward a theory of intelligence: Contributions from research with retarded children. Intelligence, 2, 279-304.

Campione, J.C., & Brown, A.L. (1977). Memory and metamemory development in educable retarded children. In R.V. Kail & J.W. Hagen (Eds.), Perspectives on the development of memory and cognition. Hillsdale, NJ: Lawrence Erlbaum Associates.

Campione, J., Brown, A., & Ferrara, R. (In press). Mental retardation and intelligence. In R. Sternberg (Ed.), Handbook of human intelligence. New York: Cambridge University Press.

Carter, J. (1984). Instructional learner feedback: A literature review with implications for software development. The Computing Teacher, 12(2), 53-55.

Clements, S.D. (1966). Minimal brain dysfunction in children. NINDB Monograph No. 3. Washington, D.C.: U.S. Department of Health, Education & Welfare.

Cohen, R.L. (1983). Reading disabled children are aware of their cognitive deficits. Journal of Learning Disabilities, 16(5), 286-289.

Cohen, R.L. (1982). Individual differences in short-term memory. In N.R. Ellis (Ed.), International review of research in mental retardation. (Vol. 11.). New York: Academic Press.

Cohen, R.L., & Netley, C. (1981). Short term memory deficits in reading disabled children in the absence of opportunity for rehearsal strategies. Intelligence, 5, 69-76.

Cohen, R.L., Netley, C., & Clarke, M. (1984). On the generality of the short term memory/reading ability relationship. Journal of Learning Disabilities, 17(4), 218-221.

Copeland, A., & Weissbrod, C. (1983). Cognitive strategies used by learning disabled children: Does hyperactivity always make things worse? Journal of Learning Disabilities, 16(8), 473-477.

Crosby, K., & Blatt, B. (1968). Attention and mental retardation. Journal of Education, 150, 67-81.

Dawson, M., Hallahan, D., Reeve, R., & Ball, D. (1980). The effect of reinforcement and verbal rehearsal on selective attention in learning disabled children. Journal of Abnormal Child Psychology, 8, 133-144.

DBS Corporation. (1982). United States Department of Education 1980 elementary and civil rights survey: National summaries. Office of Civil Rights.

Detterman, D.K. (1979). Memory in the retarded. In N.R. Ellis (Ed.), Handbook of mental deficiency: Psychological theory and research. (2nd Ed.). Hillsdale, NJ: Lawrence Erlbaum Associates.

Division of Educational Services, Special Education Programs. (1983). Fifth annual report to congress on the implementation of public law 94-142: The education for all handicapped children act. U.S. Department of Education.

Ellis, N.R. (1970). Memory processes in retardates and normals. In N.R. Ellis (Ed.), International review of research in mental retardation. (Vol. 4). New York: Academic Press.

Ellis, N.R. (1963). The stimulus trace and behavioral inadequacy. In N.R. Ellis (Ed.), Handbook of mental deficiency. New York: McGraw-Hill.

Ellis, N.R., Deacon, J.R., Harris, L.A., Poor, A., Angers, D., Diorio, M.S., Watkins, R.S., Boyd, B.D., & Cavalier, A.R. (1982). Learning, memory & transfer in profoundly, severely, & moderately mentally retarded persons. American Journal of Mental Deficiency, 87(2), 186-196.

Ellis, N., Deacon, J., & Wooldridge, P. (1985). Structural memory deficits of mentally retarded persons. American Journal of Mental Deficiency, 89(4), 393-402.

Flavell, J.H. (1979). Metacognition and cognitive monitoring, a new view of cognitive developmental inquiry. American Psychologist, 34, 906-916.

Flavell, J.H., & Wellman, H.M. (1977). Metamemory. In R.V. Kail, Jr., & J.W. Hagen (Eds.), Perspectives on the development of memory and cognition. Hillsdale, NJ: Lawrence Erlbaum Associates.

Frankel, F., & Simmons, T. (1976). Self-injurious behavior in schizophrenic and retarded children. American Journal of Mental Deficiency, 80, 512-522.

Gerber, M. (1983). Learning disabilities and cognitive strategies: A case for training or constraining problem solving? Journal of Learning Disabilities, 16(5), 255-260.

Gerjuoy, I.R., Winters, J.J., Pullen, M., & Spitz, H. (1969). Subjective organization by retardates and normals during forced recall of visual stimuli. American Journal of Mental Deficiency, 73, 791-797.

Glidden, L.M. (1979). Training of learning & memory in retarded persons: Strategies, techniques, and teaching tools. In Handbook of mental deficiency: Psychological theory and research. (2nd Ed.). Hillsdale, NJ: Lawrence Erlbaum Associates.

Goulet, L.R. (1968). Verbal learning and memory research with retardates: An attempt to assess developmental trends. In N.R. Ellis (Ed.), International review of research in mental retardation. (Vol. 3). New York: Academic Press.

Hagen, J., & Huntsman, N. (1971). Selective attention in mental retardates. Developmental Psychology, 5, 151-160.

Hagen, J., & West, R. (1970). The effects of a payoff matrix on selective attention. Human Development, 13, 43-52.

Hagen, J.W., Barclay, C.R., & Schwethelm, B. (1982). Cognitive development of the learning disabled child. In N.R. Ellis (Ed.), International review of research in mental retardation. (Vol. 11). New York: Academic Press.

Hagen, J.W., & Stanovich, K.E. (1977). Memory: Strategies of acquisition. In R.V. Kail, Jr., & J.W. Hagen (Eds.), Perspectives on the development of memory and cognition. Hillsdale, NJ: Lawrence Erlbaum Associates.

Hall, R.J. (1980). Cognitive behavior modification and information-processing skills of exceptional children. Exceptional Education Quarterly, 1(1), 9-15.

Jacobs, L. (1984). Cognition and learning disabilities. Teaching Exceptional Children, 16(3), 213-218.

Johnson, J.S., & Kaufman, M.J. (1982). Special education technology: A federal perspective. Journal of Special Education Technology, 5, 6-8.

Keogh, B., & Glover, A. (1980). The generality and durability of cognitive training effects. Exceptional Education Quarterly, 1(1), 75-82.

Kramer, J.J., & Engle, R.W. (1981). Teaching awareness of strategic behavior in combination with strategy training: Effects on children's memory performance. Journal of Experimental Child Psychology, 32, 513-530.

Kramer, J., Nagle, R., & Engle, R. (1980). Recent advances in mnemonic strategy training with mentally retarded persons: Implications for educational practice. American Journal of Mental Deficiency, 85, 306-314.

Krupski, A. (1979). Are retarded children more distractible? Observational analysis of retarded and nonretarded children's classroom behavior. American Journal of Mental Deficiency, 84, 1-10.

Latham, L.L. (1978). Construct and ecological validity of short-term memory measures in retarded persons. American Journal of Mental Deficiency, 83(2), 145-155.

Lewis, R. (1983). Learning disability and reading: Instructional recommendations from current research. Exceptional Children, 50(3), 230-240.

Lieberman, L. (1982). Learning principles and teaching models. Journal of Learning Disabilities, 15,(8), 506-507.

Lindgren, S., & Richman, L. ('984). Immediate memory functions of verbally deficient reading disabled children. Journal of Learning Disabilities, 17(4), 222-225.

Loper, A. (1980). Metacognitive development: Implications for cognitive training. Exceptional Education Quarterly, 1(1), 1-8.

Matson, J., & McCartney, J. (1981). Handbook of behavior modification with the mentally retarded. New York: Plenum Press.

McLeskey, J., Levie, W., & Fleming, M. (1982). Selective attention to visual stimuli: The need to combine theory and research. Journal of Special Education Technology, 5(2), 23-32.

Meichenbaum, D. (1980). Cognitive behavior modification with exceptional children: A promise yet unfilled. Exceptional Education Quarterly, 1(1), 83-88.

Mercer, C.D., Forgnone, C., & Wolking, W.D. (1976). Definitions of learning disabilities used in the United States. Journal of Learning Disabilities, 9, 376-386.

Mercer, C., & Snell, N. (1977). Learning theory research in mental retardation: Implications for teaching. Columbus, Ohio: Merrill.

Miller, G. A. (1956). The magical number series, plus or minus two: Some limits on our capacity for processing information. Psychological Review, 63, 81-97.

Mukherjie, A.K. (1977). A description of ambulatory profoundly retarded: An empirical survey. In L.N. Talkington, C.C. Cleland, & R.S. O'Grady (Eds.), Research with the profoundly retarded: Third annual conference proceedings. (pp.66-76).

O'Connor, G., Justice, R., & Payane, D. (1970). Statistical expectations of physical handicaps in institutionalized retardates. American Journal of Mental Deficiency, 74, 541-547.

O'Leary, S. (1980). A response to cognitive training. Exceptional Education Quarterly, 1(1), 89-94.

Pearson, P.D., & Spiro, R.J. (1980). Toward a theory of reading comprehension instruction. Topics in Language Disorders, 1(1), 71-88.

Pressley, M., Levin, J., & Bryant, S. (1983). Memory strategy instruction during adolescence: When is explicit instruction needed? In M. Pressley & J. Levin (Eds.), Cognitive strategy research: Psychological foundations. (pp. 25-49). New York: Springer-Verlag.

Ramey, C., & Campbell, F. (1979). Early childhood education for psychosocially disadvantaged children: Effects on psychological processes. American Journal of Mental Deficiency, 83, 645-648.

Ramey, C., & Finkelstein, N. (1981). Psychosocial mental retardation: A biological and social coalescence. In M. Begab, H. Garber, & H.C. Haywood (Eds.), Causes and prevention of retarded development in psychosocially disadvantaged children. Baltimore: University Park Press.

Report to Congress (1981). Unanswered questions on educating handicapped children in local public schools. HRD-81-43, Feb. 5.

Rose, M.C., Cundick, B.P., & Higbee, K.L. (1983). Verbal rehearsal and visual imagery: Mnemonic aids for learning-disabled children. Journal of Learning Disabilities, 16, 352-354.

Sailor, W., & Guess, D. (1983). Severely handicapped students: An instructional design. Boston: Houghton Mifflin.

Seidenberg, P.L. (1982). Implications of schemata theory for learning disabled readers. Journal of Learning Disabilities, 15, 35^-354.

Senf, G., & Freunal, P. (1972). Sequential auditory and visual memory in learning disabled children. Proceedings of 80th Annual Convention of the American Psychological Association, 1, 511-512.

Sheinker, A., Sheinker, J., & Stevens, L. (1984). Cognitive strategies for teaching the mildly handicapped. Focus on Exceptional Children, 17(1), 1-15.

Shiffrin, R.M., & Schneider, W. (1977). Controlled and automatic hu... information processing: II. Perceptual learning, automatic at...ding, and a general theory. Psychological Review, 84, 127-190.

Spitz, H.H. (1972). Consolidating facts into schematized learning and memory systems of educable retardates. In N.R. Ellis (Ed.), International review of research in mental retardation. (Vol. 6). New York: Academic Press.

Sternberg, R., & Wagner, R. (1982). Automization failure in learning disabilities. Topics in Learning and Learning Disabilities, 2(2), 1-11.

Swanson, H. (1978). Verbal encoding effects on the visual short term memory of learning disabled and normal readers. Journal of Educational Psychology, 70, 539-544.

Swanson, L. (1983). A study of nonstrategic linguistic coding on visual recall of learning disabled readers. Journal of Learning Disabilities, 16(4), 209-216.

Tarjan, G., Wright, S., Eyman, R., & Kearaen, C. (1973). Natural history of mental retardation: Some aspects of epid emiology. American Journal of Mental Deficiency, 77, 369-379.

Tarver, S., Hallahan, D., Cohen, R., & Kauffman, J. (1977). The development of visual selective attention and verbal rehearsal in learning disabled boys. Journal of Learning Disabilities, 10, 491-500.

Tawney, J.W., & Sniezek, K.M. (1985). Educational programs for severely mentally retarded elementary-age children: Progress, problems and suggestions. In D. Bricker & J. Filler (Eds.), Severe mental retardation: From theory to practice. Lancaster, Pennsylvania: Lancaster Press Inc.

Taylor, A.M., & Turnure, J.E. (1979). Imagery and verbal elaboration with retarded children: Effects on learning and memory. In N.R. Ellis (Ed.), Handbook of mental deficiency: Psychological theory and research. Hillsdale, NJ: Lawrence Erlbaum Associates.

Torgesen, J.K. (1981). The relationship between memory and attention in learning disabilities. Exceptional Education Quarterly, 2(3), 51-59.

Torgesen, J.K. (1979). Factors related to poor performance on memory tasks in reading disabled children. Learning Disabilities Quarterly, 2, 17-23.

Torgesen, J.K. (1977a). Memorization processes in reading disabled children. Journal of Educational Psychology, 69, 571-578.

Torgesen, J.K. (1977b). The role of non-specific factors in the task performance of learning disabled children: A theoretical assessment. Journal of Learning Disabilities, 10, 27-34.

Torgesen, J.K., & Goldman, T. (1977). Verbal rehearsal and short-term memory in reading-disabled children. Child Development, 48, 56-60.

Torgesen, J.K., & Houck, D.C. (1980). Processing deficiencies of learning disabled children who perform poorly on the digit-span test. Journal of Educational Psychology, 72, 141-160.

Torgesen, J.K., Murphy, H., & Ivey, C. (1979). The influence of an orienting task on the memory performance of children with reading problems. Joural of Learning Disabilities, 12(6), 43-48.

U.S. Department of Education. (1983). Fifth Annual Report to Congress on the Implementation of Public Law 94-142: The Education for All Handicapped Children Act. Washington, DC: U.S. Government Printing Office.

Vest, C.R. (1983). Marketing and procurement of technology assisted learning systems. Micro Market Examiner, 1(3), 1.

Waugh, N.C., & Norman, D.A. (1965). Primary memory. _Psychological Review_, _72_, 89-104.

Wellman, H., Ritter, R., & Flavell, J. (1975). Deliberate memory behavior in the delayed reactions of very young children. _Developmental Psychology_, _11_, 780-787.

Wong, B., Wong, R., & Foth, D. (1977). Recall and clustering of verbal materials among normal and poor readers. _Bulletin of the Psychonomic Society_, _10_, 375-378.

APPENDIX A

General Program Information

General Program Information

Students will participate in two general types of activity in the program: assessment and instruction at different levels of difficulty. The program is ordered such that the student always receives the assessment section first. If (s)he meets criterion on the particular level of assessment, (s)he automatically progresses to the next level of assessment. If the student fails to reach criterion on assessment, instruction at that level commences on that level. The last instructional loop at each level includes a reassessment. This cycle repeats until the student fails to reach a criterion after three consecutive instruction/assessment sequences or (s)he reaches criterion at the highest level of assessment.

The target task embodied in this program is successful circular recall. Circular recall tasks have been used extensively in direct-instruction cognitive research. While they are somewhat abstract, they permit more precise assessment of cognitive strategy usage. These tasks are related to real-world effectiveness, however, and these similarities will be addressed in pre- and post-test evaluations as well as future modifications to the software package. If a student was told to remember the string L,T,Z,J,R,P,F, (s)he could employ circular recall strategy by recalling R,P,F, and then circling back to remember L,T,Z,J. This would be referred to as a 3/4 circular recall because the student remembered first the last three elements, and then the first four.

In the instructional portion of this program, the circular recall task is disassembled and each of the four components of the effective strategy is individually trained. The student then learns how to combine the component strategies into integrated performance on the target task. The first component, known as "fast finish" training, teaches the student to retain the terminal set of items first by quickly memorizing them in a chunk. The second component, known as "cumulative rehearsal" training, shows the student how to memorize the first (and more difficult) set of elements by retrogressively rehear sing all previous elements in their original order as new ones are revealed. The third component, that of "interpolated delay and self testing", trains the student to hold those items memorized with cumulative rehearsal in memory for the amount of time equivalent to that needed to complete the fast finish on the terminal set. The final component, that of "chaining", teaches the student to incorporate the components into a unified strategy.

This software has been designed to advance through a general hierarchy of difficulty posed by various circular recall requirements. The requirements addressed in this program are in the estimated order from simplest to most difficult:

Program Level	Circular Recall Pattern	Number of Elements
A	2/2	4
B	3/2	5
C	3/3	6
D	2/4	6
E	2/5	7
F	4/4	8

The student begins assessment and instruction at the first level, that of a 2/2 circular recall requirement. There are two basic types of information that will be recorded for each student: <u>accuracy of recall</u> and <u>pause-time pattern</u>. Recall accuracy represents the number of items recalled correctly by the student. Pause-time patterns reflect the amount of time a student waits after seeing an element before displaying the next one. Thus, pause times correspond to the amount of time spent committing the item to memory. Since long strings of elements take longer to rehearse than do short strings, this would be reflected in corresponding differences in pause times. Circular recall requires the student to cumulatively rehearse certain elements; if pause times between elements do not vary, it is an indication that the student is not using the strategy.

Accuracy of recall is reflected in a percentage score derived by dividing the number of correctly recalled trials by the total number of trials and multiplying by 100. In addition to computing the accuracy of the whole string of elements, separate accuracy scores will be determined for the two components of the circular recall (initial items and terminal items). The accuracy criterion varies with the number of elements to be remembered:

Cumulative Rehearsal		Fast Finish	
Number of Elements to be remembered	Criterion Greater than or equal to	Number of Elements to be remembered	Criterion Greater than or equal to
2	2		
3	3	2	2
4	3	3	2
5	3	4	3
6	4		

The pause-time pattern criterion is a predetermined omega-squared value, (see attached) which quantifies the comparison between the student's pause-time pattern and an "ideal" pattern. Pause-time patterns will be computed for all phases of training but the cumulative rehearsal portions of Levels A and B and the fast finish portions of every level, and accuracy scores will be computed for all phases of training.

Pause Time Ideals

Cumulative rehearsal - 1 second per inter-item pause

2 item list: 1 second
3 item list: 1 second, 2 second
4 item list: 1 second, 2 second, 3 second
5 item list: 1 second, 2 second, 3 second, 4 second
6 item list: 1 second, 2 second, 3 second, 4 second, 5 second

Fast Finish: .75 second per inter-item pause

2 item list: .75 second
3 item list: .75 second, .75 second
4 item list: .75 second, .75 second, .75 second

Level	Cum. Rehearsal List Length	Omega^2	Total List Length	Omega^2
A	2	--	4	.979
B	2	--	5	.873
C	3	.999	6	.775
D	4	.979	6	.775
E	5	.873	7	.699
F	4	.979	8	.639

To reach criterion on assessment or any portion of training, the student must meet both accuracy and pause-time criteria: achievement of the accuracy criterion indicates that the student is able to remember the specified number of elements, and attainment of the pause-time criterion indicates that the student is employing the appropriate cognitive strategy. In the assessments, the student receives three trials which are used to compute the performance data. In the instructional portion four components are taught. Performance on a component must reach criterion before the student can proceed to the next component. A block of three trials at the end of each component is used for assessing mastery.

APPENDIX B

Information for Programmer

Information For Programmer

General Information

- The Ready Screen and Recall Screen always remain on for one second unless specifically stated in the program narrative.
- The Recall Screen remains on for periods from .5 to 6 seconds, and recall is signaled by 3 bursts of a high frequency tone that take up .1 second apiece at the end of the Recall Screen's duration (e.g., if duration of screen is .5 second, screen is on in silence for .2 second and is paired with the tones at .3, .4, and .5 second).
- When illuminating letters in the individual boxes, a subject should select the boxes in order from left to right. Any deviation from this pattern should result in the system's ignoring the incorrect selection, and waiting for the correct one. Regardless of the duration of the ensuing time delay, the screen will respond when the correct box is activated. After a box has been illuminated for .5 seconds, it will change color to indicate that it has already been activated.
- When a subject selects individual letters for placement into the boxes, placement position will be predetermined (e.g., the first letter selected automatically goes into the third of five boxes, the second letter selected goes into the fourth box, etc.). The student may not correct errors of placement, but (s)he may use a letter more than once.
- When elements on the screen are to flash, this flashing lasts for .75 seconds.

- The letters to be used for each trial will be selected from the pool of all English consonants. To avoid perceptual confusion, no phonetically similar consonants can occur in the same portion of the list (fast finish, cumulative rehearsal). All phonetically similar letter names will be grouped together, and a randomization subroutine will choose only one letter from each grouping:

  B,C,D,G,P,T,V,Z
  M,N
  S,X,F
  J,K
  H
  L
  Q
  R
  W
  Y

- For the portions using numbers, the numbers will be randomly selected from 0 through 9. No letter or number may appear more than once during any one trial. The order of the letters when displayed beneath the empty boxes should also be randomized.

- All text to be spoken will also be represented orthographically. This written text will appear on the bottom 4 lines of the screen. Previous text should be erased from the screen before new text appears; in other words, successive utterances should not scroll, but should appear independent of one another.

- To use screen 11 (versions A, B, and C) with element strings greater than 2, the boxes will need to be collapsed into the 2 represented on screen 11.

- After completion of every assessment portion and its accompanying video game interlude, screen 16 will appear. The student is required to respond with the joystick/light pen in order to have the program continue. If the student does not respond, screen 16 will remain visible until the teacher performs the escape function.

- The teacher will have the option of discontinuing student interaction at any time by the use of the escape function. The programmer will determine the specific keystroke sequence required for this function.

- The next time a student works with the system following termination of an interaction using the escape function, assessment/instruction will begin at the level at which the student was working when interaction was halted. For the purpose of counting the number of times through assessment/instruction, tabulation will begin at zero as if the child had not previously interacted at the level.

- A student will be allowed a specific amount of time in which to respond. If no response has been made by the end of this time period, it will be prompted by the aural and written cue "Do it now". In all portions of the program except the study times, this interval will be 30 seconds. The interval will be 45 seconds during study times (the times in which the student is illuminating the letters in the boxes in order to study them prior to his/her recall attempt). After the prompt, the student has 30 seconds in which to make a response in all portions of the program. If (s)he responds within that 30-second period, the program continues in the manner specified in the Program Narrative. If the student fails to respond within the second (30-second) time period, (s)he receives the following message:

  You are taking too much time when it's your turn.
  Let's try another one.

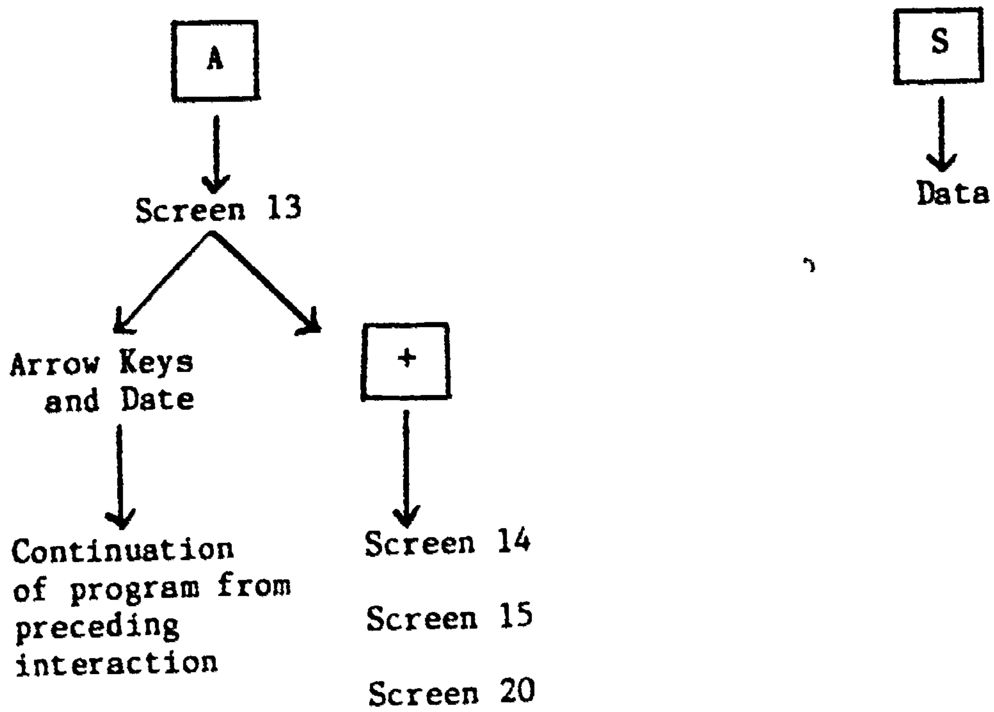
If the student fails to respond in the allotted time on this next problem, receives the prompt, and continues to be unresponsive, his/her interaction with the system will be discontinued, the program will terminate, and his/her performance data file will be closed.

Any time a student receives a prompt during instruction, that trial will not be counted toward his/her progression to the next step of the program. To progress, the student must complete a trial independently (that is, without the use of the prompting procedure described above).

o In all portions of instruction (fast finish, cumulative rehearsal, interpolated delay, and chaining) the final loop consists of six trials. The first three of these will be trials in which the correction procedure may be applied; that is, the student is cycled back to the beginning of that portion at Loop 1. The last three of these trials will serve as the assessment, and the correction procedure will not be used.

Beginning of Program

Always begin with Screen 12



Levels of Difficulty

1. Introduction

2. Input mode familiarization

3. Preassessment training on interaction with screen

   A. 2/2

   B. 3/2

   C. 3/3

   D. 2/4

   E. 2/5

   F. 4/4

Programming Additional Levels

Level A is contained in the program narrative. It includes both the assessment and instructional portions. All other levels follow the same course, the only modifications being the total number of elements in the string, and the circular recall configuration.

The following is a breakdown of these variables according to level.

Level	Total # of Elements	Initial Items (for cum. rehearsal)	Terminal Items (for fast finish)
B	5	2	3
C	6	3	3
D	6	4	2
E	7	5	2
F	8	4	4

Changes in the text/speech will be necessary on words like "second", "both", "middle", etc., and screen 5 will also need to change accordingly.

Data Recording/Analysis

1. Types of information needed

   a. Demographics:

      name
      birthdate
      date of each interaction with computer

   b. Task parameters of each interaction:

      level
      assessment/instruction
      number of times through each instructional component loop

Fast finish:	Loop 1 (complete assistance)
	Loop 2 (voice/graphic assistance)
	Loop 3 (graphic assistance)
	Loop 4 (no assistance)
Cumulative rehearsal:	Loop 1 (complete assistance)
	Loop 2 (voice/graphic assistance)
	Loop 3 (graphic assistance)
	Loop 4 (no assistance)
Interpolated delay:	Loop 1 (1 second delay)
	Loop 2 (2 second delay)
	Loop 3 (3 second delay)
	Loop 4 (4 second delay)
	Loop 5 (5 second delay)
	Loop 6 (6 second delay)
Chaining:	Loop 1 (complete assistance)
	Loop 2 (voice/graphic assistance)
	Loop 3 (graphic assistance)
	Loop 4 (no assistance) - serves as post-instruction assessment

   The first time the student goes through a level (e.g., Level A), the data will be designed as A1. The second time through it will be designated A2, and so on. After three unsuccessful cycles (i.e., not reaching criterion) through the entire assessment/instruction package, his interaction with the program is terminated.

   c. Assessment data:

      number of items correctly recalled for each position in each of 3 assessment trials

      pause time for each position in each of 3 assessment trials

      average cumulative score on recall accuracy of terminal items

      average cumulative score on recall accuracy of initial items

d. Computations to be performed:

omega^2 for each trial
average accuracy across 3 assessment trials

2. Data printouts available

   a. Graph or table of average pause time as a function of position

   b. Graph or table of recall accuracy as a function of position

   c. Table of instructional information: record of cycles through each loop per instructional component

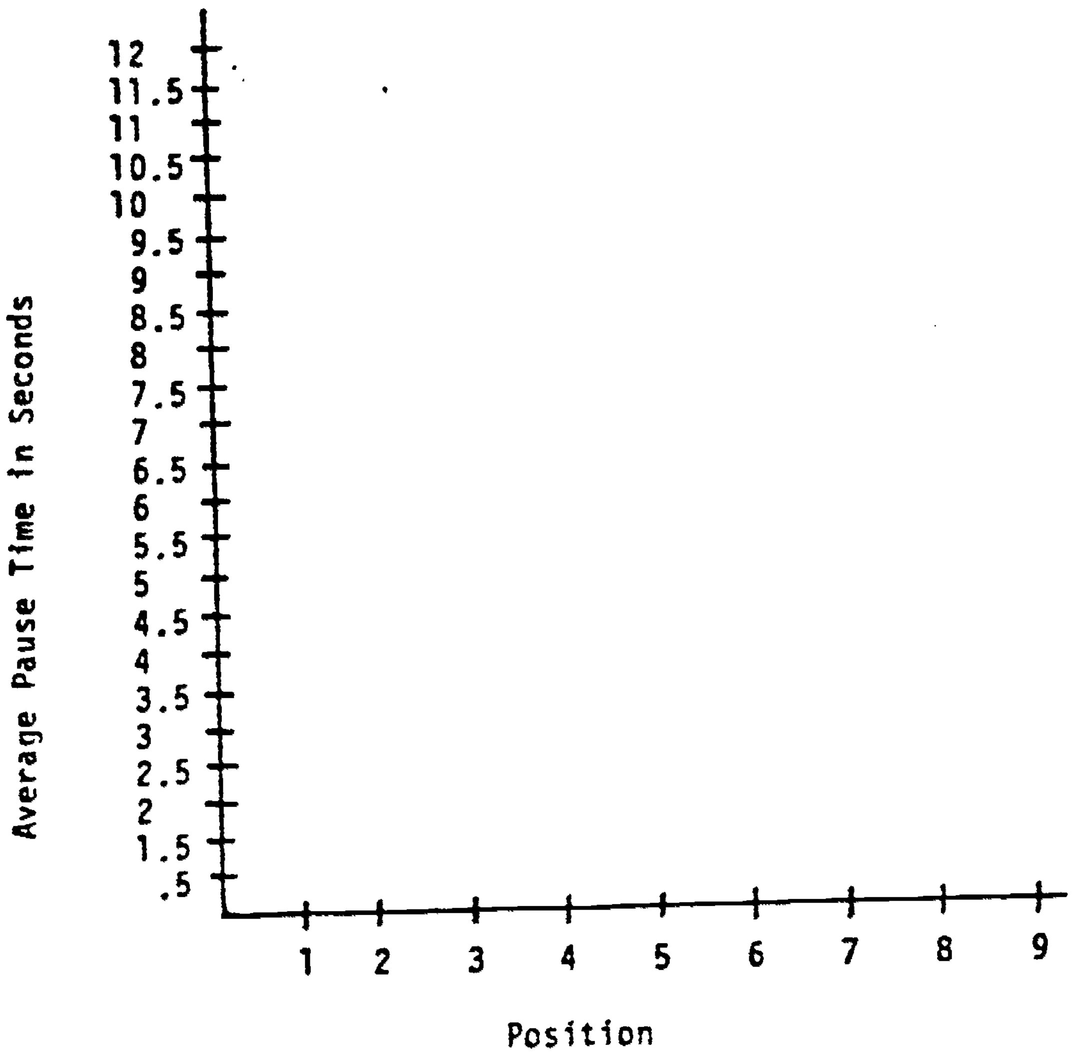
   d. Interpretive remarks

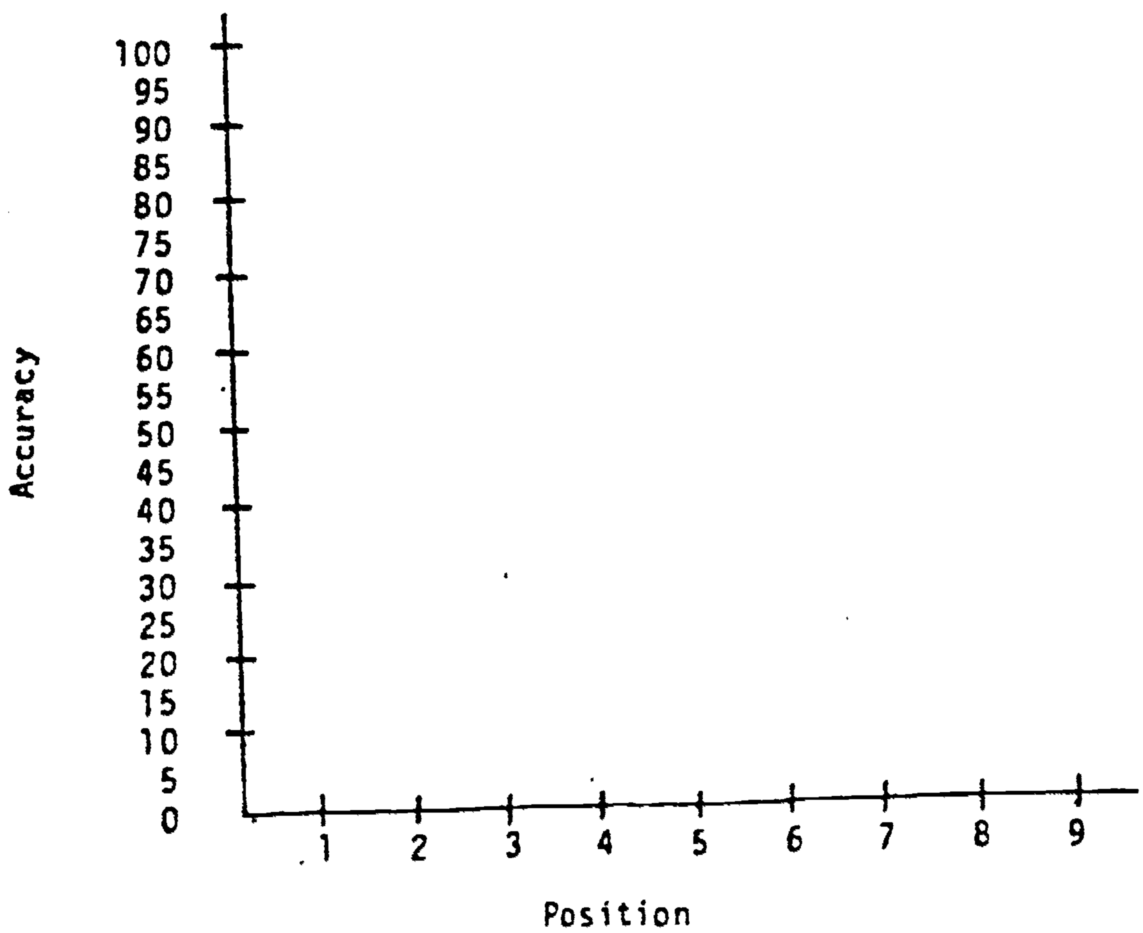
3. Accessing printouts

   a. Daily printout - available after end of daily interaction

   b. Entire data base - available during initial screen interaction

Graphic Printouts





Omega^2 Computation

Omega^2 (ω^2) is the value reflecting the "fit" between an ideal pause-time pattern and the pause-time pattern demonstrated by the subject. An omega^2 value is computed for each of the last three assessment trials. The computational formula is:

$$\text{Omega}^2 = \frac{SS_{sp} - (df_{sp} \times MS_{error})}{SS_t + MS_p + (2)MS_{error}}$$

Formulae for each of the above values is provided next. The following chart should assist in interpretation of these formulae.

SERIAL POSITION

		j = 1	j = 2	j = 3	j = 4	j = 5	j = 6	j = 7	TOTAL
i = 1	SUBJECT								
i = 2	IDEA								
	TOTAL								

i is an index of a specific row

j is an index of a specific column

The student's raw pause-time scores first must be transformed into Z scores to permit comparison with ideal times.

Steps to Calculating Z scores:

$$A = \frac{x_1 + x_2 + x_3 + \ldots x_n}{n}$$

$$B = \sqrt{\frac{(x_1 - A)^2 + (x_2 - A)^2 + \ldots (x_n - A)^2}{A}}$$

$$Z = \frac{x_i - A}{B} + 4$$

x_i = individual score ($x_1, x_2, x_3 \ldots$)

n = number of serial positions

Steps to Calculating the Mean Square for Patterns (MS_p):

$$A = (x_{1,1} + x_{1,2} + x_{1,3} + \ldots x_{1,j})^2 + (x_{2,1} + x_{2,2} + x_{2,3} + \ldots x_{2,j})^2$$

$$B = \frac{A}{n_{sp}}$$

$$C = x_{1,1} + x_{2,1} + x_{1,2} + x_{2,2} + x_{1,3} + \ldots x_{i,j}$$

$$D = C^2$$

$$E = \frac{D}{(2)\,(n_{sp})}$$

x = a score

$x_{i,j}$ = the score in the i-th row and j-th column

n_{sp} = the number of serial positions

n_p = the number of patterns

Mean Square for Patterns = Sums of Squares for Patterns = B-E

Steps to Calculating the Sums of Squares for Serial Postions (SS_{sp}):

$$A = (x_{1,1} + x_{2,1})^2 + (x_{1,2} + x_{2,2})^2 + \ldots(x_{1,j} + x_{2,j})^2$$

$$B = \frac{A}{2}$$

$$C = x_{1,1} + x_{2,1} + x_{1,2} + x_{2,2} + x_{1,3} + x_{2,3} + \ldots x_{i,j}$$

$$D = C^2$$

$$E = \frac{D}{(2)\ (n_{sp})}$$

Sums of Squares for Serial Positions = B-E

x = a score

$x_{i,j}$ = the score in the i-th row and the j-th column

n_{sp} = the number of serial postions

Mean Square for Serial Position = $\dfrac{B - E}{(n_{sp} - 1)}$

Steps to Calculating the Sums of Squares of the Total (SS_t):

$$A = x_{1,1}^2 + x_{2,1}^2 + x_{1,2}^2 + x_{2,2}^2 + x_{1,3}^2 + x_{2,3}^2 + \ldots x_{i,j}^2$$

$$C = x_{1,1} + x_{2,1} + x_{1,2} + x_{2,2} + x_{1,3} + x_{2,3} + \ldots x_{i,j}$$

$$D = C^2$$

$$E = \frac{D}{(2)(n_{sp})}$$

x = a score

$x_{i,j}$ = the score in the i-th row and j-th column

n_{sp} = the number of serial positions

93

Steps to Calculating the Mean Square of the Error (MS_{error}):

Sums of Squares of the Error = $SS_{total} - SS_{sp} - SS_{sp}$

$$\text{Mean Square of the Error} = \frac{\text{Sums of Squares of the Error}}{(n_{sp} - 1)(n_p - 1)}$$

Steps to Calculating Degrees of Freedom for Serial Postions (df_{sp}):

$$df_{sp} = n_{sp} - 1$$

n_{sp} = the number of serial positions

Pre/Post Assessment Disk

The Pre/Post Assessment will be contained on a disk separate from that containing the actual Assessment-and-Instruction program. The data from the Pre/Post Assessment will be contained on another disk in a two-drive system and on the Pre/Post Assessment disk in a one-drive system. The Pre/Post Assessment will be used for evaluative purposes only in the project and will not be included in the final software product.

In the Beta test phase, Pre/Post Assessment will be administered to a set of students, a subset of whom will interact with the Assessment-and-Instruction program (which can be considered the treatment condition in the experimental design). The students who do not receive the Assessment-and-Instruction program can be considered to be in the control condition.

The number of different recall requirements in the Pre/Post Assessment is greater than the number in the Assessment-and-Instruction program. The recall requirements that are not included in the latter (and therefore are not trained) will be used on the Pre/Post Assessment to derive a measure of limited generalization, i.e., a transfer of training to tasks similar to, but not identical with, the training tasks.

The introduction to the task and selection mode in the Pre/Post Assessment sequence will follow exactly the specifications in the Program Narrative for Levels 1, 2, and 3. It then follows the identical procedure set forth in Loop 4 of the "Chaining" component of the instructional portion, which is described on pages 51 and 52 of the Program Narrative. The format for the assessment will remain the same; only the list length and circular recall pattern will vary.

Recall Requirement	Total # of Elements	Initial Item (Cum. Rehearsal)	Terminal Item (Fast Finish)
2/2	4	2	2
2/3	5	3	2
3/3	6	3	3
4/2	6	2	4
2/4	6	4	2
4/3	7	3	4
3/4	7	4	3
2/5	7	5	2
4/4	8	4	4
3/5	8	5	3
4/5	9	5	4
3/6	9	6	3

APPENDIX C

Program Narrative

LEVEL 1

This portion of the program is simply a brief introduction. It will be presented to the student only once.

LEVEL	PAGE	VOICE	NUMBER	ACTIVITY ON SCREEN
1	1	Hello. This is a game to see how good you are at remembering numbers and letters. Before we start playing, I will teach you how to use the computer and what to do. Here we go.	12 Blank	Screen 12 remains illuminated for the duration of the text.

LEVEL 2

This portion of the program is designed to familiarize the student with the input mode s/he will be using. Instructions in the use of each mode - light pen and joystick - are included, but the child will receive only those pertinent to the input mode selected for him/her by the teacher.

LEVEL	PAGE	VOICE	NUMBER	ACTIVITY ON SCREEN
2	1	You will use a light pen when you work with the computer. We will practice with it now so that you get to be good at using it	1	
		This is what I want you to do.	2	Flash far left-hand box.
		See the box that's blinking? Your job is to put the number 3 in that box.		
		This is how you do it: Put the tip of the pen right on the number 3.		(Positive feedback follows correct response immediately. If student fails to respond within 3 seconds or responds incorrectly, corrective feedback is given). When student activates #3, number appears in box that was flashing. Flashing stops.
		+ Good you did it. It went into the box. − Put the pen right on the number 3.		
		Let's try it again.		
		See the blinking box?		Flash middle box.
		Put the number 2 in that box. Touch it with the pen, and it will go into the box.		(Positive feedback follows correct response immediately. If student fails to respond within 3 seconds or responds incorrectly, corrective feedback is given). Number 2 appears in box when activated. Flashing stops.
		+ Great you've got the idea. − Put the pen right on the number 2.		

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
2	2	Now you try one all by yourself. Get the number 1. + Good. You did it. − Put the pen right on the number 1.	2	Flash right-hand box. (Positive feedback follows correct response immediately. If student fails to respond within 3 seconds or responds incorrectly, corrective feedback is given. Number 1 appears in box when activated. Flashing stops.

LEVEL	PAGE	VOICE	NUMBER	ACTIVITY ON SCREEN
2	3	You will use a joystick when you work with the computer. We will practice with it now so that you get to be good at using it.	3	
		This is what I want you to do.	2	Flash left-hand box.
		See the box that's blinking? Your job is to put the number 3 in that box.		
		This is what you should do: Move the joystick so that the light is on the number 3, and then push the joystick button.		Cursor moves according to child's control. When child activates #3, number appears in box that was flashing. Flashing stops.
				(Positive feedback follows correct response immediately. If student fails to respond within 5 seconds or responds incorrectly, corrective feedback is given).
		+ Good. You did it. It went into the box.		
		− Move the light onto the number 3 and press the button.		
		Let's try it again.		
		See the blinking box?		Flash middle box.
		Put the number 2 in that box. Move the light to the number 2, press the button, and it will go into the box.		Number 2 appears in box when activated. Flashing stops. (Positive feedback follows correct response immediately. If student fails to respond within 5 seconds or responds incorrectly, corrective feedback is given).

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
2	4	+ Great. You've got the idea. - Move the light onto the number 2, and press the button.	2	
		Now try one all by yourself.		
		See the blinking box?		Flash right-hand box.
		Get the number 1.		Number 1 appears in box when activated. Flashing stops. (Positive feedback follows correct response immediately. If student fails to respond within 5 seconds or responds incorrectly, corrective feedback is given).
		+ Good. You did it. - Move the light onto the number 2, and press the button.		

LEVEL 3

This portion of the program is intended to familiarize the student with the symbols and routines employed throughout the rest of the program. It takes the student through some interactions that, while simplified, are typical of the ones in which (s)he will participate.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
3	1	This is a game to see how good you are at remembering. I'm going to show you some numbers. I'll show them to you one at a time.	5	Screen shows 3 boxes (only lower ones)
				Left-hand box flashes
		See the flashing box? That's the one to light up first. I'll pretend to be you. Watch what I do.		
		After you light it up, a number will appear in it. Your job is to remember this number.		
				Little hand points to box, number appears for .5 sec then goes off.
		After the number disappears, light up the next box and you will see the next number to remember.		
				Middle box flashes next.
				Little hand points to box, number appears for .5 sec then goes off.
				Right-hand box flashes.
		Now light up the last box.		
				Little hand points to box, number appears for .5 sec then goes off.
		When you light up the boxes this way, always work from the left-hand side to right-hand side.		
				Light up a triangle on left side of screen.
				Move arrow across screen from left to right. Using triangle on left as tip of arrow.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
3	2		5	Shows three lower boxes.
		Now its your turn to light up the boxes.		
				Left-hand box flashes.
		Light up the box.		
				Upon activation, illuminates the numbers in the left hand box for .5 second.
				Middle box flashes
		Light up this one next.		
				Upon activation illuminates the middle box for .5 second.
				Right-hand box flashes.
		Light up the next one.		
				Upon activation, illuminates the number in the right-hand box for .5 second.
		Good. You saw all the numbers.		
		This means the numbers are coming.	7	
			5	Shows three lower boxes.
		Light up the first one.		
				Upon activation, illuminates the numbers in the left-hand box for .5 second.
		Light up the next one.		
				Upon activation, illuminates the numbers in the middle box for .5 second.
		Light up the next one		
				Upon activation, illuminates the numbers in the right-hand box for .5 second.
		Good. You saw all the numbers.		

LEVEL	PAGE	VOICE	NUMBER	ACTIVITY ON SCREEN
3	3	Now you will see how to recall the numbers. I will be telling you rules about which number to recall first.	BLANK	
		You will see this every time I am ready to tell you a new rule. When you hear this (sound 3 short tones, each of .1 second duration), it means that it is time to recall the numbers.	6	Stars flash in alternating pattern.
		I'll show you.	BLANK	
		This means a new rule is coming. Here is the rule:	6	Stars flash in alternating pattern

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
3	4	This time, you should try to recall first the number you saw in this box,	5	
				Shows only 3 lower boxes Flashes middle box.
		and then recall the ones in the other boxes. Since you always work from left to right, you would do this one next,		
				Flashes right-hand box.
		and then, since there aren't any more to the right, go back and do the other one.		
				Flashes left-hand box.
		This means the numbers are coming.	7	
		If you light up the boxes in the wrong order, they will not show a number.		
			5	Screen shows 3 lower boxes
		Watch what I do.		Left-hand box flashes.
		I light up this one first.		Little hand points to box, number appears for .5 second, then goes off.
				Middle box flashes.
		This one next.		Little hand points to box, number appears for .5 second, then goes off.
				Right hand box flashes.
		And this one next.		Little hand points to box, number appears for .5 second, then goes off.
			8	(recall signal)
		I saw all the numbers. Now I will remember them.		

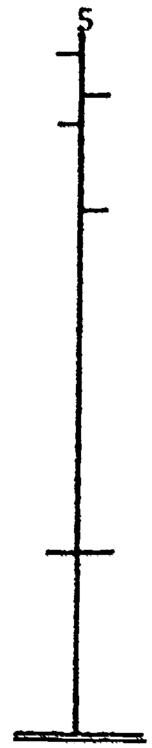
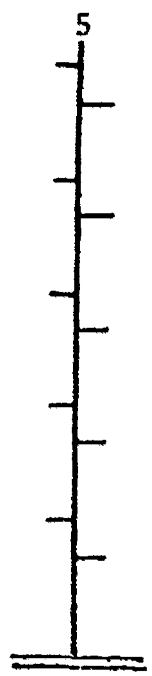
LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
3	5		5	Shows 3 lower boxes, with numbers below them.
		The rule said to remember this one first.		
				Little hand goes to middle box, then to number that belongs in middle box.
		This number goes there.		
		This number goes in the next one.		Little hand goes to number that belongs in right-hand box.
		This number goes in the other box.		Little hand goes to number that belongs in left-hand box.
		Let's see if I got it right.		
				Another row of boxes appears above those already on the screen.
				Draws box outline around upper and lower middle selections. Illuminates number in upper middle box, which is the same as that in lower middle box.
		Good. These are the same.		Colors in area in box outline.
		Let's look at the next one.		
				Draws box outline around upper and lower right-hand selections. Illuminates number in upper right-hand box, which is the same as that in the lower right-hand box.
		These are the same.		Colors in area in box outline.
		Let's look at the next one.		

LEVEL	PAGE	VOICE	NUMBER	ACTIVITY ON SCREEN
3	6		5	Draws box outline around upper and lower left-hand selections. Illuminates number in upper left-hand box, which is the same as that in the lower left-hand box.
		These are the same, too.		Colors in area in box outline.
		Good. I got them all.		
				3 boxes collapse into 2.
			11	Trucks move across screen.
			Blank	
		Now its your turn to recall the numbers.		
			6	Stars flash in alternating pattern.
	A	Recall this one first.	5	Flashes middle box.
	A	Then this one,		Flashes right-hand box.
	A	and then this one.		Flashes left-hand box.

LEVEL	PAGE		VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
3	6a	B	Put your pen on the one you are going to recall first	5	Shows three lower boxes
			+ Good. That's right.		
			- No. Recall this one first.		Flash middle box (repeat from B until student performs correctly).
		C	Put your pen on the one you will recall next.		
			+ Good. That's right.		
			- No. Recall this one next.		Flash right-hand box (repeat from A until student performs correctly).
		D	Put your pen on the one you will recall next.		
			+ Good. That's right.		
			- No. Recall this one next.		Flash left-hand box (repeat from A until student performs correctly).

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
3	7	The numbers are coming.	7	Move arrow across screen.
			5	Shows three lower boxes.
				Left-hand box flashes.
		Light up the box.____		
				Upon activation, illuminates the number on the left-hand box for .5 second.
		Remember the number you saw. When you're ready, light up this one.		
				Middle box flashes.
				Upon activation, illuminates the number in the middle box for .5 second.
		When you're ready, light up the next one.		
				Right-hand box flashes.
				Upon activation, illuminates the number in the right-hand box for .5 second.
		O.K. You saw all the numbers. When you hear the beeps, it will be time to recall the numbers by putting them where they belong.		
			8	(recall signal)

LEVEL	PAGE	VOICE	NUMBER	ACTIVITY ON SCREEN
3	8		5	Shows 3 lower boxes, with numbers underneath.
		Remember the rule? Do this one first.		Flashes middle box.
		What number was in this box?		Continues to flash middle box until a number is activated.
		Get it. This will put it in the box.		
				Number appears in box.
		+ Yes, that's right.		
		– No, that's not the right one. Try again.		Erases number in middle box. Flashes middle box until another number is activated. This number appears in box. Repeat until student activates correct number.
		What number was in this box?		
		Put it in the box.		Flashes right-hand box.
				Continues to flash middle box until a number is activated. When activated, the number appears in box.
		+ Yes, that's right.		
		– No, that's not the right one. Try again.		Erases number in middle box. Flashes middle box until another number is activated. This number appears in box. Repeat until student activates correct number.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
3	9	What number was in this box?		
		Put it in the box.		Flashes left-hand box.
				Continues to flash left-hand box until a number is activated. When activated, the number appears in box.
		+ Yes, that's right.		
		– No, that's not right. Try again.		Erases number in middle box. Flashes middle box until another number is activated. This number appears in box. Repeat until student activates correct number.
		Let's see how you did.	Blank	
		This is how I did mine.		Illuminates upper 3 boxes above existing boxes.
		The rule said to do this one first.		Hand points to middle-box.
		I got the one that goes here.		Hand goes to number belonging in middle-box - it appears in box.
		Moving to the right.		Hand goes to right-hand box.
		I got the one that goes in here.		Hand goes to number belonging to right-hand box - it appears in box.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
3	10	Going back to the left.	5	
				Hand goes to left-hand box.
		I got the one that goes in here.		
				Hand goes to number belonging in left-hand box - it appears in box.
		Did you do it that way?		
		Let's see if we did it the same.		
				Draws box outline around upper & lower left-hand selections.
		+ Good. You did just like I did.		Colors in area in box outline.
		– These don't match		Dissolves box outline.
				Draws box outline around upper & lower middle selections.
		Are these the same?		
		+ Good. Yours is just like mine.		Colors in area in box outline.
		– No. They're not the same.		Dissolves box outline.
				Draws box outline around upper & lower right-hand selection.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
3	11	Are these the same?	5	
		+ Good. Yours is just like mine.		Colors in area in box outline.
		− No. They're not the same.		Dissolves box outline.
*		O.K. We're going to try it again.	Blank	
			6	Stars flash in alternating pattern.
		This time the rule is to recall.	5	Illuminates three lower boxes. Flash left-hand box.
		This one first.		
		Remember that.		
		The numbers are coming.	7	
				Move arrow across screen.
		Remember to go left-to-right. Light up the box.	5	When left-hand box is activated, illuminates number in box for .5 second.
		When you are ready, light up the next one.		When middle box is activated, illuminates number in box for .5 second.
		Light up the next one.		When right-hand box is activated, illuminates number in box for .5 second.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
			8.5	(recall screen)
3	12	Now its your turn to put the numbers in the boxes. Remember which one to put in first.		
				Number chosen by student goes into left-hand box.
		Put in the next one.		
				Number chosen by student goes into middle box.
		Put in the next one.		
				Number chosen by student goes into right-hand box.
		Let's see if yours is the same as mine.		
				Illuminates upper 3 boxes above existing boxes.
				Draws box outline around upper & lower left-hand selections.
		+ Good. Yours is just like mine.		Colors in area in box outline.
		− These are not the same.		Dissolves box outline.
				Draws box outline around upper & lower middle selections.
		+ You did it.		Colors in area in box outline.
		− They don't match		Dissolves box outline.
				Draws box outline around upper & lower right-hand selections.

LEVEL	PAGE	VOICE	NUMBER	ACTIVITY ON SCREEN
3	13	+ Good they match.	5	Colors in area in box outline.
		– These do not match.		Dissolves box outline.
				If all three areas are colored in, goes to screen 11.
				(Repeat from * until student performs two consecutive series with 100% accuracy.)

LEVEL A

This is the level at which the students begin assessment and training on circular recall strategy use. Thus strategy targeted for training at this level is the "2/2", which involves remembering the last 2 of 4 digits first using a "fast finish" technique, and then remembering the first 2 digits using a "cumulative rehearsal" technique. The students receive training on each of these techniques separately, and then learn to chain them together.

LEVEL	PAGE	VOICE	NUMBER	ACTIVITY ON SCREEN
A	1	I am going to show you some letters.	Blank	
			5	Shows 4 lower boxes
		Your job is to remember this many letters.		
		Here is the recall rule:	6	Flashes stars in alternating pattern.
			5	Shows 4 lower boxes Flashes B 3
		Recall first the letter that appears in this box.		
		Which one are you going to recall first? Get it.		
		Yes, that's the right one		
		No, it's this one.		Flashes the correct box.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	2	Let's begin	7	Moves arrow across screen.
			5	Shows 4 lower boxes.
		Light up the letters in the boxes.		
				When B_1, activated, LN_1, appears in it for .5 Sec.
				When B_2, activated, LN_2, appears in it for .5 Sec.
				When B_3, activated, LN_3, appears in it for .5 Sec.
				When B_4, activated, LN_4, appears in it for .5 Sec.
				Shows 4 empty boxes. Letters appear below in random order in a row. (recall signal)
			8	
		Put the letters in the right boxes. Remember the rule.	5	
				First letter selected goes in B_3
				Second letter seletec goes in B_4
				Third letter selected goes in B_1
				Fourth letter selected goes in B_2
		Let's see how you did. Here is the right answer.		Shows 4 upper boxes above existing ones. Letters appear in boxes in correct order all at once. (are not put on one-at-a-time.)
		These are the ones that you recalled correctly.		Boxes in area around all correctly matched upper & lower selections, and fills them in with color.
		Repeat this page 5 more times for a total of 6 assessment trials.		

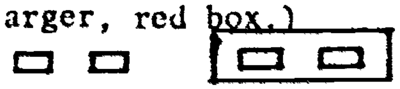
LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	3	Following final trial comes brief interlude of video game.		

Performance:

Meets criterion on last 3 trials --- longer interlude of game, move on to next level of assessment

less than criterion --- brief game interlude, move into training at this level.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	4	Your job is to remember this many letters.	5.	
				Shows screen with lower 4 boxes.
		I'm going to show you some ways to study when you have to recall the last two letters then the first two letters.		
		The first way is to remember the letters in a little group.		
				Distinguishes between two clusters of boxes by drawing a blue box around the first two and a red box around the last two. 
		I am going to show you how to study the last letters.		

LEVEL	PAGE	VOICE	SCREEN NUMBER 7	ACTIVITY ON SCREEN
A	5		7	Moves arrow across screen
			5	(Although all 4 remain on screen, now working with 2-box group, only this group is highlighted with a larger, red box.) 
		See these 2 boxes?		
				Moves hand to left-hand box of the 2-box group.
		I'm lighting up the first one.		
				Letter appears in the left-hand box for .5 second.
		It's a <u>lettername</u>, (LN). I say the letter and go on right away to the next one.		
				Moves hand to right-hand box.
		Now I'm lighting up the next one.		
				Letter appears in right-hand box for .5 second.
		Its a <u>(LN)</u>. I say <u>(LN)</u>.		

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	6		8	(recall signal)
		This tells me to recall all the letters I saw	5	Displays the 2 letters in random order below the boxes.
				With all 4 boxes on screen and last 2 highlighted.
				Flashes left-hand box of the 2-box group.
		Which one went here?		
		It was a (LN). I will put the (LN) in the box.		
				Little hand goes to (LN); letter appears in flashing left-hand box and flashing stops.
				Flashes right-hand box.
		This one is next.		
				Little hand goes to letter for right-hand box; it appears in right-hand box and flashing stops.
		Let's see if I did it right.		

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	7		5	2 upper boxes appear above lower ones.
		Here is the way it is supposed to be.		Little hand points to top row. Puts a letter in left-hand box. then right-hand box.
		Here is what I did.		Little hand points to bottom row.
		Let's see if they are the same.		Draws box outline around upper and lower left-hand selections.
		Good. They match.		Colors in area in box outline between the boxes and the out-lining borders. Draws box outline around upper and lower right-hand selections.
		These match, too.		Colors in area in box outline.
			11	Trucks drive off screen.
		Now I want you to help me.	BLANK	

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	8		5	With all 4 boxes on screen, the groups separated and last 2 boxes highlighted:
		Here are 2 boxes.		
				Flashes 2 boxes momentarily.
		You are going to light up the letters in the boxes.		
				Flashes left-hand box of 2-box group.
		Light up this one.		
				Upon activation of box, letter appears in left-hand box for .5 second.
		(It's a (LN). Say out loud (LN).		
				(pauses 2 seconds to allow student to respond)
		Now light up this one.		
				Flashes right-hand box.
				Upon activation of box, letter appears in middle box for .5 second.
		It's a (LN). Say out loud (LN).		
				(pauses 2 seconds to allow student to respond)
			8	(recall signal)
			5	With all 4 boxes on the screen and last two highlighted: The 2 letters that had been in the boxes appear below in random order.
		Put the right letter in this box.		Flashes left-hand box. Upon activation of a letter, that letter appears in left-hand box.
		Put the right letter in this box.		Flashes right-hand box. Upon activation of a letter, that letter appears in right-hand box.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	9	Let's see if you got it right	5	
		Here's the right answer.		Upper row of 2 boxes appears. Correct letters are filled in: left-hand, right-hand.
		Let's see if they are the same.		Draws box outline around upper and lower left-hand selections.
		$\boxed{+}$ Good. They match.		Colors in area in box outline.
		$\boxed{-}$ No. They don't match		Dissolves box outline.
				Draws box outline around upper and lower right hand selections.
		$\boxed{+}$ Good. These match.		Colors in area in box outline.
		$\boxed{-}$ They don't match.		Dissolves box outline.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	10	Great. You got them all.	5	If all correct goes to screen 11 and moves trucks across screen.
		Let's do some more.		
		Let's try it again.		If incorrect, repeat from page 7.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	11		7	Move arrow across screen.
			5	With all 4 boxes on the screen with last 2 highlighted:
		Light up the letters in the last 2 boxes. Try to move through these quickly.		
				Flashes left-hand box of the second 2-box group.
				Upon activation of box, letter appears in left-hand box for .5 second.
				Flashes right-hand box.
				Upon activation of box, letter appears in right-hand box for .5 second.
		O.K. You saw all the letters.		
			8	(recall signal)
			5	All 4 boxes on screen, last 2 highlighted. Displays letters in random order below boxes.
		Put the letters in the right boxes.		
				Flashes left-hand box of the second 2-box group.
				Upon activation of letter, it appears in left-hand box.
				Flash right-hand box.
				Upon activation of letter, it appears in right-hand box.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	12	Repeat Comparison Procedure 1 (Page 9).		

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	13	Good, you got them all.	5	If all correct, go to screen 11 and move trucks across screen. Go to next page.
		Let's try again.		If incorrect, repeat from page 11.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	14		7	Moves arrow across screen.
			5	With all 4 boxes on the screen, and last 2 highlighted:
		Light up the last 2 letters in the boxes.		
				Upon activation of left-hand box, letter appears in it (activation of any other box is ignored) for .5 second.
				Upon activation of right-hand box, letter appears in it for .5 second.
			8	(recall signal)
			5	With all 4 boxes on the screen and last 2 highlighted:
				Display letters in random order below boxes.
		Put the letters in the right boxes...		
				As student selects each letter, it goes into a box:
				Box #3 - box into which 1st letter selected appears Box #4 - box into which 2nd letter selected appears

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	15	(Repeat comparison procedure 1 (Page 9).)		

Instruction - Fast Finish

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	16	Good. You got them all. Let's do some more.	———	If all correct: Go to screen 11 and move trucks across screen. Repeat page 14 five more times.
		Let's try again.	———	If incorrect: Repeat page 11.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	17	(If after 6 trials at page 14, student is performing at or above the specified accuracy criterion for the particular list length across the last three trials, he moves on to the next page.)		

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	18		5	
				Shows 4 boxes, but nothing is highlighted.
		You just learned a way to remember these letters.		
				Highlights last 2 boxes with a box around them.
		Now you are going to learn a way to remember these first two letters.		
				Eliminates highlighting of last 2. Highlights first 2 in same manner.
		So that you can remember them when you have to remember the last two and then the first two, you will need to practice letters by saying them, first out loud and then in your head.		
		You are going to see the letters one at a time like before. Watch me.		

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	19		7	Move arrow across screen.
			5	Shows boxes, highlighting first 2.
				Flashes left-hand box.
		When this letter comes on, I'm going to say it once.		
				Little hand goes to flashing box; letter appears in box for .5 second.
		$\underline{LN_1}$ (Voice says LN aloud.)		
				Flashes right-hand box.
		When this letter comes on, I will say the first letter, then the second one, Until I am sure I know them.		
				Little hand goes to flashing box; letter appears in box for .5 second.
		$\underline{LN_1}$ $\underline{LN_2}$, $\underline{LN_1}$ $\underline{LN_2}$		
		(Voice says both letters aloud, twice.)		

INSTRUCTION LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	20	I'm practicing the letters to help me recall them.	BLANK	
		$\underline{LN}_1$ $\underline{LN}_2$, $\underline{LN}_1$ $\underline{LN}_2$		
			$\frac{8}{5}$	(recall signal)
				Shows 4 boxes, highlighting first 2.
				Letters appear below.
				Display letters in random order below boxes.
				Hand goes to (LN_1); letter appears in LH box.
				Hand goes to (LN_2); letter appears in RH box.

Instruction - Cumulative rehearsal

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	21		5	
		Let's see if I got them right.		
		Here's the way it is supposed to be.		Upper row of 2 boxes appears.
				Little hand points to the top row. letters are filled in:left-hand, right hand
		Here is what I did.		
				Little hand points to bottom row.
		Let's see if they are the same.		
				Draws box outline around upper & lower left-hand selections.
		Good,they match.		
				Colors in area in box outline.
				Draws box outline around upper & lower right-hand selections.
		These match, too.		
				Colors in area in box outline.
		I got them all right.	-11-	Moves trucks across screen.
		Now I want you to help me.	Blank	

Instruction - Cumulative Rehearsal

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	22		7	Moves arrow across screen.
			5	Shows 4 boxes, highlights first two.
				Flashes left-hand box.
		See the flashing box? Light it up.		
				When activated, flashing stops and letter LN_1 appears for .5 second. (pause 1 second to allow student time to repeat)
		Out loud, say $\underline{LN_1}$.		
				Flash right-hand box.
		Light up the next box.		
				When activated, flashing stops and letter LN_2 appears for .5 second. (pause 2.5 seconds to allow student time to repeat)
		Out loud, say LN_1, LN_2; LN_1, LN_2.		
			8	(recall signal)
			5	Shows 4 boxes, highlighting first two. Letters (LN1 & LN2) appear below in random order.
		Now it's your turn to put the letters in the boxes.		
				Flashes left-hand box.
		What goes here?		
				When activated, letter appears in left-hand box.
				Flashes right-hand box.
		What goes here?		
				When activated, letter appears in right-hand box.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	23		5	
		Let's see if you got them right.		
		Here's the right answer.		Upper row of 2 boxes appears. Letters are filled in: left-hand, right-hand
		Let's see if they are the same.		
				Draws box outline around upper and lower left-hand selections.
	Comparison Procedure 2	+ Good. They match		Colors in area in box outline.
		− No. They don't match.		Dissolves box outline.
				Draws box outline around upper and lower right-hand selections.
		+ They match.		Colors in area in box outline.
		− They don't match.		Dissolves box outline.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	24	Good. You got both of them.	———	If all correct, goes to screen 11 and moves trucks across screen.
		Let's do some more.	———	go to next page.
		Let's try again.	———	If incorrect, Repeat from page 22.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	25		7	Moves arrow across screen.
			5	With all 5 boxes on the screen, and first 2 highlighted.
		Light up the letters in the boxes.		
		Remember to say them over and over in your head.		
				Flashes left-hand box.
				When activated, flashing stops and letter LN_1 appears in it for .5 second. (pause 1 second)
				Flashes right-hand box.
				When activated, flashing stops and letter LN_2 appears in it for .5 second. (pause 2.5 second)
			8	(recall signal)
		You saw both of the letters - Put them in the right boxes.	5	Shows 5 boxes, highlighting first two. Letters appear below in random order.
				Flashes left-hand box.
				When activated, letter appears in left-hand box.
				Flashes right-hand box. When activated, letter appears in right-hand box.

Instruction - Cumulative Rehearsal

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	26	Repeat comparison procedure 2 page 23.		

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	27	Great. You got them both.	———	If both correct:
		Let's do some more.	———	Go to screen 11 and moves trucks across screen. Go to next page.
		Let's try again.	———	If incorrect: Repeat from page 25.

Instruction - Cumulative Rehearsal

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	28		7	Moves arrow across screen.
			5	With all 4 boxes on the screen, and first 2 highlighted:
		Light up the letters in the boxes.		
				When left-hand box is activated, letter LN_1 appears in it for .5 second.
				When right-hand box is activated, letter LN_2 appears in it for .5 second.
			8	(recall signal)
		Put the letters in the right boxes.	5	Shows 4 boxes, highlighting first two. letters appear below in random order.
				First letter activated appears in left-hand box.
				Second letter activated appears in right-hand box.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	29	Repeat comparison procedure 2 page 23.		

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	30	Great. You got them both.	———	If all correct: Goes to screen 11 and moves trucks across screen.
		Let's do some more.	———	Repeat from page 28 five more times.
		Let's try again.	———	If incorrect: Repeat from page 25.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	32		7	Moves arrow across screen.
			5	With all 4 boxes on the screen and the first 2 highlighted:
		Light up the letters in the boxes. Don't move on to a new letter until you are <u>sure</u> you can remember the other ones. You can practice them as many times as you want to be sure you know them.		
		Light up the letters in the boxes.		
				When left-hand box is activated, LN_1 appears in it for .5 second.
				When right-hand box is activated, LN_2 appears in it for .5 second.
			8	(recall signal - question mark remains on screen for 1 second; 3 beeps enter after .7 second has elapsed.)
			5	Shows 4 boxes, highlighting first 2 letters appear below in random order.
		Put the letters in the right boxes.		
				First letter activated appears in left-hand box.
				Second letter activated appears in right-hand box.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	33	Repeat Comparison Procedure 2 page 23.		

Instruction - Interpolated delay

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	34	$\boxed{+}$		
		Great, You got them both. See, you can really remember those letters, even if you have to wait before recalling them.	———	If all correct: Go to screen 11 and move trucks across screen.
		Let's do some more.	———	Go to next page.
		$\boxed{-}$		
		Let's try again.	———	Repeat from page 32.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	35		7	Moves arrow across screen
			3	With all 4 boxes on the screen and first 2 highlighted:
		I'm going to make you work a little harder. This time you are going to have to remember the letters for a longer time before you can recall them. Practice the letters over and over while you are waiting to put them in the right boxes. This will help you remember them.		
		Light up the letters in the boxes.		
				When left-hand box is activated, LN_1 appears in it for .5 second.
				When right-hand box is activated, LN_2 appears in it for .5 second.
		Practice these until its time to put them in the boxes.		
			8	(recall signal - question mark remains on screen for 2 seconds; 3 beeps after 1.7 seconds have elapsed.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	36		5	Shows 4 boxes, highlighting first 2. Letters appear below in random order.
		Put the letters in the right boxes.		
				First letter activated appears in left-hand box.
				Second letter activated appears in right-hand box.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	37	Repeat Comparison Procedure 2, page 23.		

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	38	+		
		Great. You got them both.	———	If all correct: Go to screen 11 and move frogs across screen.
		Let's do some more.	———	(Go to next page.)
		-		
		Let's try it again.	———	Repeat from page 35.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	39		7	Moves arrow across screen.
			5	With all 4 boxes on screen and first 2 highlighted.
		Remember to practice the letters until you're sure you know them.		
		Light up the letters in the boxes.		
				When left-hand box is activated, LN_1, appears in it for .5 second.
				When right-hand box is activated, LN_2, appears in it for .5 second.
			8	(recall screen - question mark remains on screen for 3 seconds; 3 beeps after 2.7 seconds have elapsed.)
			5	Shows 4 boxes, highlighting first 2. Letters appear below in random order.
		Put the letters in the right boxes.		
				First letter activated appears in left-hand box.
				Second letter activated appears in right-hand box.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	40	Repeat Comparison Procedure 2, page 23.		

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	41	[+]		
		Great. You got them both.	——	If all correct: Go to screen 11 and move mice across screen.
		Let's do some more	——	Repeat pages 39 thru 41 with: A. 4- second delay during Recall Screen: question mark up for 4 seconds; 3 beeps after 3.7 seconds. If "A" correct, repeat pages 39-41 with B. 5-second delay during Recall Screen: question mark up for 5 seconds; 3 beeps after 4.7 seconds. If "B" correct, repeat pages 39-41 with C. 6-second delay during Recall Screen: question mark up for 6 seconds; 3 beeps after 5.7 seconds. If "C" correct, go to next page.
		[-]		
		Let's try it again. Keep saying the letters over and over in your head to hel you recall them.	——	Repeat from page 39 with appropriate delay interval.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	42	Repeat "C" (page 41) 5 more times.		

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	43	If, after 6 trials at "C" (page 41) student is performing on last 3 trials at or above the specified level for accuracy and omega^2 for a given list length, he moves on to next page.		

LEVEL	PAGE	Instruction - Chaining VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	44	Now you are going to use all the things that you've learned at once so that you can recall all these letters.	5	Shows 4 boxes, not highlighted in any way.
		I'm going to see if you can remember all these letters at one time.		Flashes all 4 boxes.
			6	Stars flash in alternating pattern.
		This is the rule.		
			5	Shows boxes, not separated.
		Start recalling with this one first, then this one, and this one, and that one.		Flashes boxes in appropriate circular recall sequence.
		I'll show you what I mean.		
		I'll pretend to be you. Watch how I study the letters so that I can recall the last ones and then the first ones.		

Instruction - Chaining

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	45		7	Arrow moves across screen
		I'm going to light up the letters in the boxes. I'm going to practice the first two over and over in my head, and then move through the last two quickly.		
			5	Shows 4 boxes (to be referred to in this narrative as B1 B2 B3 B4 for explanatory purposes)
				Little hand goes to B1. LN_1 appears for .5 second.
		LN_1		
				Little hand goes to B_2. LN_2 appears for .5 second.
		LN_1, LN_2, LN_1, LN_2. I'm practicing these to make sure I know them.		
				Little hand goes to B_3, LN_3 appears for .5 second.
		LN_3		
				Little hand goes to B_4, LN_4 appears for .5 second.
		LN_4		
			8	(recall signal)
			5	Shows 4 boxes Letters appear below in random order.
		The rule told me to recall this one first.		Little hand goes to B_3.
		What letter was there?		
				Little hand goes to LN_3. It appears in B_3.
				Little hand goes to LN_4. It appears in B_4.
		Now I'll go back and recall the ones I practiced.		
				Little hand goes to LN_1. It appears in B_1.
				Little hand goes to LN_2. It appears in B_2.
		Let's see if I did it right.		

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	46	Here is the way it is supposed to be.	5	Upper row of 4 boxes appears. Little hand points to the top row. Letters are filled in this order: (1 letter every .2 second)
				B_3, B_4, B_1, B_2
		Here is what I did.		Little hand points to bottom row.
				Draws box outline around upper and lower B_3 selections.
		Good they match.		
				Colors in area in box outline.
				Draws box outline around upper and lower B_4 selections.
		Good they match.		
				Colors in area in box outline.
				Draws box outline around upper and lower B_1 selections.
		Good they match.		
				Colors in area in box outline.
				Draws box outline around upper and lower B_2 selections.
		These match, too.		
				Colors in areas in box outline.
		I got them all right.	11	Moves trucks across screen.
		Now I want you to help me.	Blank	

Instruction - Chaining

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	47	Remember the rule:	6	Stars flash in alternating pattern.
		When you are recalling where the letters go, do this one first.	5	Shows 4 boxes; flash B_3.
		Which one are you going to recall first? Get it!		Flashing stops.
		$\boxed{+}$ Yes, that's the one		
		$\boxed{-}$ No, its this one (Flash B_3) (Repeat procedure from level 3 page 6a with the appropriate recall pattern)	7	Arrow moves across screen.
		You are going to light up the letters in the boxes. Remember to say the first two over and over in your head, and then move through the last two quickly.	5	Shows 4 boxes.
				Flash B_1.
		See the flashing box? Get it!		When B_1 activated, flashing stops and LN_1 appears in it for .5 second.
		LN_1 .		
				(Pause 1 second to allow student time to repeat.)
				Flashes B_2.
				When B_2 activated, flashing stops and LN_2 appears in it for .5 second.
		LN_1, LN_2; LN_1, LN_2.		
				(Pause 3 second to allow student time to repeat.)
		Repeat these until you are sure you know them. When you're sure you know them, then get the next one.		Flashes B_3.
				When B_3 activated, flashing stops and LN_3 appears in it for .5 second.
		LN_3.		
				Flashes B_4.
				When B_4 activated, flashing stops and LN_4 appears in it for .5 second.
		LN_4.		

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	48		8	(recall signal)
			5	Shows 4 boxes Letters appear below in random order.
		Now it's your turn to put the letters in the boxes.		
		Which one did the rule say to recall first?		
		Get it!		
		+ Good		
		- No. Recall this one first		Flashes B_3.
		What letter goes here?		
				When activated, letter appears in B_3.
				Flashes B_4.
		What letter goes here?		
				When activated, letter appears in B_4.
				Flashes B_1.
		Here?		
				When activated, letter appears in B_1.
				Flashes B_2.
		Here?		
				When activated, letter appears in B_2.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	49	Let's see if you got it right.	5	
		Here's the right answer.		Upper row of 4 boxes appears. Letters filled in in this order:
				B_3, B_4, B_1, B_2,
		Let's see if they match.		Draws box outline around upper and lower B_3 selections.
		+ Good they match.		Colors in area in box outline.
		− No. They don't match.		Dissolves box outline.
	Comparison Procedure 3			Draws box outline around upper and lower B_4 selections.
		+ They match		Colors in area in box outline.
		− They don't match.		Dissolves box outline.
				Draws box outline around upper and lower B_1 selections.
		+ They match		Colors in area in box outline.
		− They don't match.		Dissolves box outline.
				Draws box outline around upper and lower B_2 selections.
		+ Good. They match.		Colors in area in box outline.
		− They don't match.		Dissolves box outline.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	50	Good. You got them all.	———	If all correct:
		Let's do some more.	———	Goes to screen 11 and moves trucks across screen.
		Let's try again.	———	If incorrect:
				Repeat from page 41.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	51		6	Flashes stars in alternating fashion.
		Remember the rule: Recall this one first.	5	Shows 4 boxes; flash B_3 for 1 second.
		Get the one you will recall first. + Good - No. You should recall this one first		Flashes B_1.
			7	Moves arrow across screen.
			5	Shows 4 boxes.
		Light up the letters in the boxes. Remember to say the first two over and over in your head until you're sure you know them, and then to move through the last two quickly.		Flashes B_1.
				When B_1 activated, flashing stops and LN_1 appears in it for .5 second.
		Remember to say these over and over in your head.		(pause 1 second)
				Flashes B_2.
				When B_2 activated, flashing stops and LN_2 appears in it for .5 second.
		Go through these next 2 quickly.		Flashes B_3.
				When B_3 activated, flashing stops and LN_3 appears in it for .5 second.
				Flashes B_4.
				When B_4 activated, flashing stops and LN_4 appears in it for .5 second.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	52		8	(recall screen)
			5	Show 4 boxes Letters appear below in random order.
		Put the letters in the right boxes. Remember the rule.		
				Flashes B_3.
				When activated, letter appears in B_3.
				Flashes B_4.
				When activated, letter appears in B_4.
				Flashes B_1.
				When activated, letter appears in B_1.
				Flashes B_2.
				When activated, letter appears in B_2.

LEVEL	FRAME	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	53	Repeat Comparison Procedure 3, Page 43.		

Instruction- Chaining

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	54	Good you got them all.	———	If all correct.
		Let's do some more.	———	Goes to screen 11 and moves trucks across screen.
		Let's try again.	———	If incorrect.
				Repeat from page 45.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	55		7	Moves arrow across screen.
			5	Shows 4 boxes.
		Light up the letters in the boxes.		
				When B_1 activated, LN_1 appears in it for .5 second.
				When B_2 activated, LN_2 appears in it for .5 second.
				When B_3 activated, LN_3 appears in it for .5 second.
				When B_4 activated, LN_4 appears in it for .5 second.
			8	(recall screen)
			5	Shows 4 boxes. Letters appear below in random order.
		Put the letters in the right boxes. Remember the rule.		
				First letter selected appears in B_3.
				Second letter selected appears in B_4.
				Third letter selected appears in B_1.
				Fourth letter selected appears in B_2.

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	56	Repeat Comparison Procedure 3, Page 43.		

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
A	57	Great, you got them all.	——	If all correct: Goes to screen 11 and moves trucks across screen. Repeat from page 49 five more times.
		Let's try again.	——	If incorrect: Repeat from Page 45.

LEVEL	PAGE	VOICE Instruction - Chaining	SCREEN NUMBER	ACTIVITY ON SCREEN
A	58	If after 6 trials at page 49 student is performing at or above the specified accuracy and omega^2 criteria for the particular list lengths during the last 3 of the 6 trials, s/he receives the assessment for the next circular recall requirement.		
		Before the beginning of instruction on the third level of circular recall, the student sees and hears the following:		
		You are doing a great job. When you had to remember this many letters		Shows 4 boxes
		You recalled these first		draws box around last 2
		and then went back and recalled these.		draws box around first 2 and eliminates box around last 2
		and when you had to remember this many letters		Shows 5 boxes
		you recalled these first		draws box around last 3
		and then went back and recalled these.		draws box around first 2 and eliminates box around last 3
		You practiced some of the letters over and over, and remembered some in a little group		draws box around last 3
		Now you'll get a chance to do this with more letters. Keep up the good work.		

Instructions for Video Game

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
		You will make the universe safe for mankind if you make contact with all the blue stars.	———	Show several blue stars.
		You make contact every time you put your light on a blue star.	———	Show little hand going to blue star among red and yellow stars. When hand contacts star, it turns white, whistles, and then disappears.
		You get a point for each blue star you contact.	———	Show number 1 in score box.
		Sometimes the stars will be shooting stars that move, and then they'll be worth 5 points if you make contact with them.	———	Show blue shooting stars.
		The evil alien ship is roaming the universe trying to capture blue stars.	———	Show alien ship crossing screen full of stars, grabbing stars in its path.
		If it does, the only way to get them back is to blast the alien ship with a missile as it moves across the screen.	———	
		You can fire missiles by putting your light in the middle of the launch pad.	———	Show little hand contacting center of launch pad, firing a missile that hits the alien ship.
		If you blast the alien ship when it is carrying captured stars, the stars will be released and you will score a point for each one.	———	Show alien ship blowing up and point total increasing in score box. The stars that were inside the ship remain on the screen for 1 second, turn white, and disappear.
		If you contact all the blue stars and blast the alien ship you will get to play for an extra minute.	———	

Instructions for Video Game

LEVEL	PAGE	VOICE	SCREEN NUMBER	ACTIVITY ON SCREEN
		You will make the universe safe for mankind if you make contact with all the blue stars.	——	Show several blue stars.
		You make contact every time you put your light on a blue star and push the button.	——	Show little hand going to blue star among red and yellow stars. When hand contacts star, it turns white, whistles, and then disappears.
		You get a point for each blue star you contact.	——	Show number 1 in score box.
		Sometimes the stars will be shooting stars that move, and then they'll be worth 5 points if you make contact with them.	——	Show blue shooting stars.
		The evil alien ship is roaming the universe trying to capture blue stars.	——	Show alien ship crossing screen full of stars, grabbing stars in its path.
		If it does, the only way to get them back is to blast the alien ship with a missile as it moves across the screen.	——	
		You can fire missiles by putting your light in the middle of the launch pad and push the button.	——	Show little hand contacting center of launch pad, firing a missile that hits the alien ship.
		If you blast the alien ship when it is carrying captured stars, the stars will be released and you will score a point for each one.	——	Show alien ship blowing up and point total increasing in score box. The stars that were inside the ship remain on the screen for 1 second, turn white, and disappear.
		If you contact all the blue stars and blast the alien ship you will get to play for an extra [illegible].	——	

Screen 12 - August

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

MAIN OPTION SCREEN

SELECT AN OPTION, THEN PRESS $\boxed{\text{RETURN}}$

$\boxed{\text{I}}$ TO RETURN TO STUDENT IDENTIFICATION SCREEN

$\boxed{\text{V}}$ TO VIEW INDIVIDUAL STUDENT PERFORMANCE RECORDS

$\boxed{\text{D}}$ TO DELETE INDIVIDUAL STUDENT PERFORMANCE RECORDS

$\boxed{\text{T}}$ TO BEGIN TRAINING OR ASSESSMENT

$\boxed{\text{Q}}$ TO QUIT

TITLE ______________ PROGRAMMER ______________ PAGE _____ OF _____

Screen 13 - August

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

STUDENT IDENTIFICATION

SELECT AN OPTION, THEN PRESS RETURN

R TO REVIEW LIST OF STUDENTS ON FILE

This allows you to display, print, and/or erase the student identification list. WARNING: erasing the ID list destroys all performance records.

A TO ADD A NEW STUDENT TO THE STUDENT IDENTIFICATION FILE

S TO SELECT A STUDENT ALREADY ON FILE

Q TO QUIT

TITLE ______ PROGRAMMER ______ PAGE ____ OF ____

Screen 13A, -August

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

STUDENTS ON FILE

Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname

SELECT AN OPTION, THEN PRESS RETURN

L TO SEE THE REST OF THE STUDENT FILE

I TO RETURN TO IDENTIFICATION SCREEN

TITLE ____________ PROGRAMMER ____________ PAGE ____ OF ____

Screen 13A2- August

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

STUDENTS ON FILE

Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname
Firstname Lastname	Firstname Lastname

SELECT AN OPTION, THEN PRESS RETURN

P TO GET PRINTOUT OF ENTIRE FILE LIST

E TO ERASE ENTIRE FILE LIST

I TO RETURN TO IDENTIFICATION SCREEN

TITLE ________ PROGRAMMER ________ PAGE ____ OF ____

Screen 14 - August

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

ENTER NEW INFORMATION BELOW

LAST NAME: ____________ (PRESS RETURN)
(Enter up to 10 characters)

FIRST NAME: __________ (PRESS RETURN)
(Enter up to 9 characters)

DATE OF BIRTH: MM/DD/YY (PRESS RETURN)

SELECT AN OPTION, THEN PRESS RETURN

SEX: M MALE

F FEMALE

STUDENTS WILL USE: L LIGHTPEN

J JOYSTICK

TITLE 271 PROGRAMMER ______ PAGE ____ OF 272

Screen 15 - August

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

ENTER NEW INFORMATION (continued)

NUMBER OF DISK DRIVES: 1 ONE

2 TWO

VIDEOGAME SPEED: S SLOW

M MEDIUM

F FAST

OUTPUT: V VOICE

T TEXT

B BOTH

TITLE ______ PROGRAMMER ______ PAGE ___ OF ___

Screen 20 - August

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

PLEASE REVIEW THE INFORMATION PROVIDED

LAST NAME: ________
FIRST NAME: ________
SEX: ________
INPUT: ________
DISK DRIVES: ________
GAME SPEED: ________
OUTPUT: ________

SELECT AN OPTION, THEN PRESS $\boxed{\text{RETURN}}$

$\boxed{\text{Y}}$ INFORMATION IS CORRECT

$\boxed{\text{N}}$ ANY INFORMATION IS INCORRECT

Correct errors by typing the correct response over the existing one. Press $\boxed{\text{RETURN}}$ to move to the next item.

TITLE ______ PROGRAMMER ______ PAGE ___ OF ___

Screen 30 - August

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

STUDENT SELECTION

TYPE STUDENT NAME AS IT APPEARS ON FILE

FIRST (RETURN)

LAST (RETURN)

The closest match will appear below.

TYPE I TO SCROLL UP THE LIST

TYPE J TO SCROLL DOWN THE LIST

NAME

SELECT AN OPTION, THEN PRESS RETURN

S TO SELECT THE NAME IN THE WINDOW

A TO ADD A NEW NAME TO THE FILE

TITLE 277 PROGRAMMER PAGE ____ OF 278

Screen 40 - [illegible]

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

INDIVIDUAL STUDENT PERFORMANCE RECORDS

RECORDS ARE AVAILABLE IN THREE FORMS
You may choose to see any or all forms, but please select only one at a time.

SELECT AN OPTION, THEN PRESS RETURN

1 GRAPH

2 TABLE

3 WRITTEN INTERPRETATION

PERFORMANCE RECORDS ARE AVAILABLE TO SUMMARIZE ALL WORK TO DATE OR THE WORK DURING THE MOST RECENT SESSION.

4 ALL WORK TO DATE

5 MOST RECENT SESSION

TITLE ______ PROGRAMMER ______ PAGE ____ OF ____

Screen 50 - August

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

STUDENT PERFORMANCE RECORDS

DATE(S):

SELECT AN OPTION, THEN PRESS RETURN

P PRINT INFORMATION ON SCREEN
V VIEW ANOTHER FORM OF INSTRUCTION
M RETURN TO MAIN OPTION SCREEN
Q QUIT

TITLE 281 PROGRAMMER PAGE OF 282

BPC • 225 S. WESTERN AVENUE, LOS ANGELES, CA 90004 AVAILABLE AT COMPUTER STORES NATIONWIDE

Warning/Verification Screen B

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

*** IMPORTANT ***

This selection indicates you have chosen to delete ______________'s performance records. This means that you will destroy all data for ______________ now in memory.

ARE YOU SURE YOU WANT TO DO THIS?

[Y] for YES
(PRESS [RETURN])
[N] for NO

ARE YOU ABSOLUTELY SURE YOU WANT TO DISCARD THIS STUDENT FILE?

[Y] for YES
(PRESS [RETURN])
[N] for NO

283 284

TITLE ______________ PROGRAMMER ______________ PAGE ____ OF ____

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

*** IMPORTANT ***

This selection indicates you have chosen to erase all student files. This means that you will destroy all data now in memory.

ARE YOU SURE YOU WANT TO DO THIS?

[Y] for YES

(PRESS [RETURN])

[N] for NO

ARE YOU ABSOLUTELY SURE YOU WANT TO DISCARD ALL STUDENT FILES?

[Y] for YES

(PRESS [RETURN])

[N] for NO

TITLE 285 PROGRAMMER ______ PAGE ____ OF 286

Date screen

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

TODAY'S DATE

Please enter today's date using the following format.

FORMAT: MM/DD/YY

EXAMPLE: Enter March 15, 1985 as 03/15/85

TODAY'S DATE: MM/DD/YY

PRESS RETURN TO ACCEPT ENTRY

TITLE ______ PROGRAMMER ______ PAGE ___ OF ___

APPENDIX D

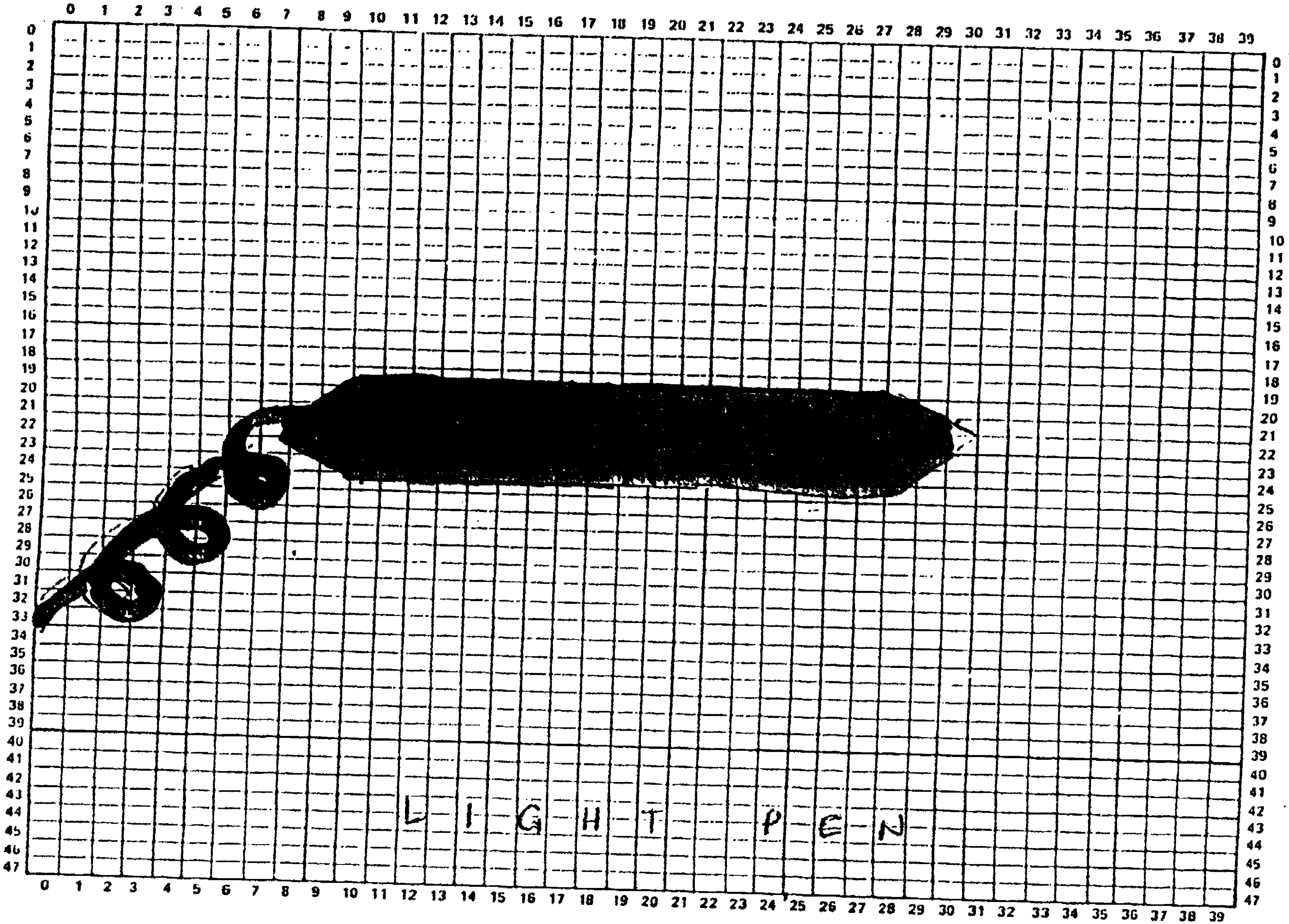
Screen Layouts

Screen Layouts

Screen Number	
12	Title Screen - Parameters (assessment/training, student records)
13	Parameters - Student name/date (previous interactant)
14	Parameters - Student name/date (new interactant)
15	Parameters - Selection mode, game speed, etc.
20	Parameter review
1	Light pen
3	Joystick
2	Familiarization with selection mode drill
9	Two-digit span with correction boxes
5A	Digit span (up to 9 characters)
5B	Digit span with correction boxes
6	Rule screen
7	Ready screen
8	Recall screen
11A	Reward screen - trucks
11B	Reward screen - frogs
11C	Reward screen - mice
21	Video game
16	Continue interaction

screen 1

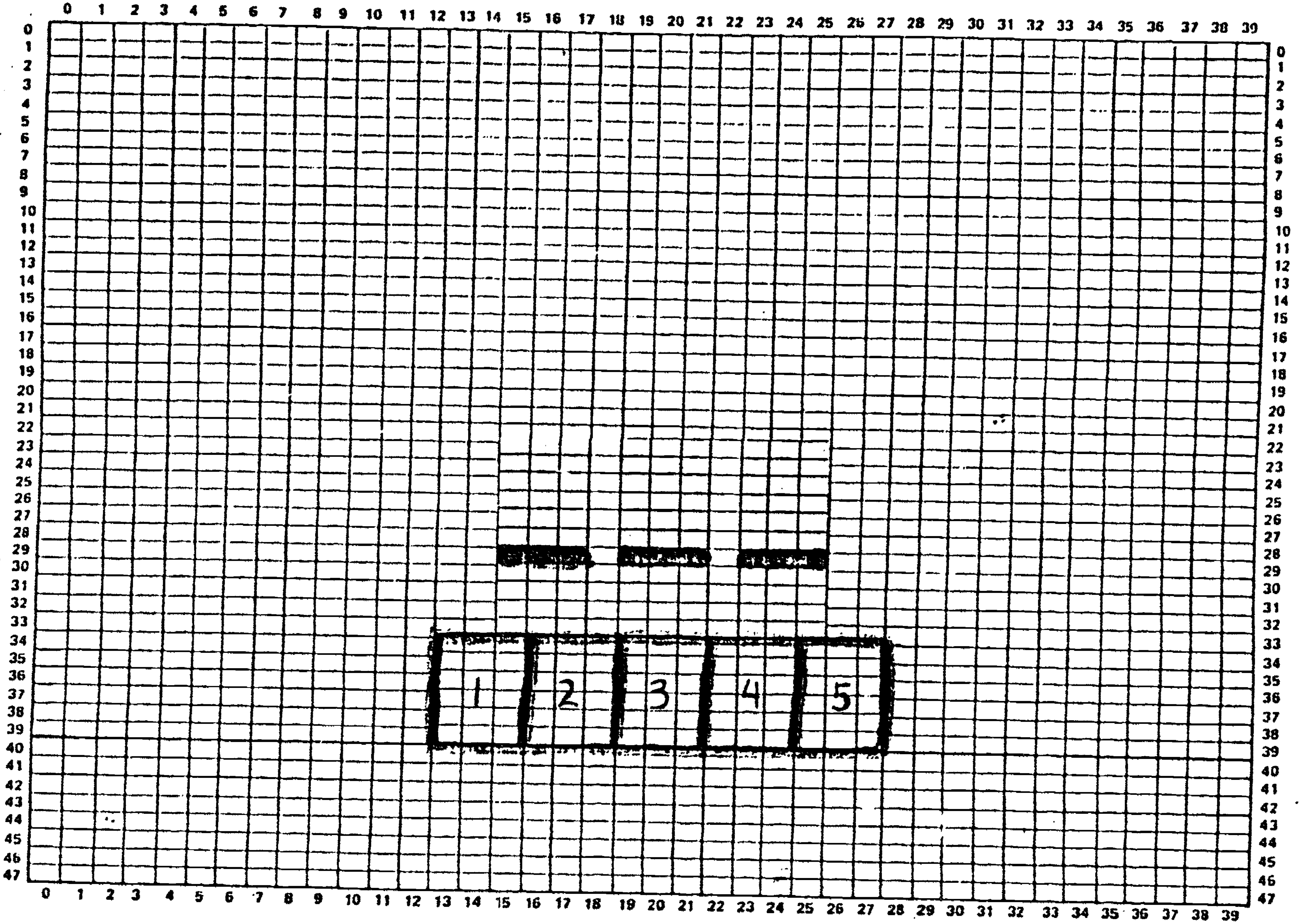
APPLE VIDEO DISPLAY WORK SHEET (LO-RES)



TITLE ______________________ PROGRAMMER ______________________ PAGE _____ OF _____

Screen 2

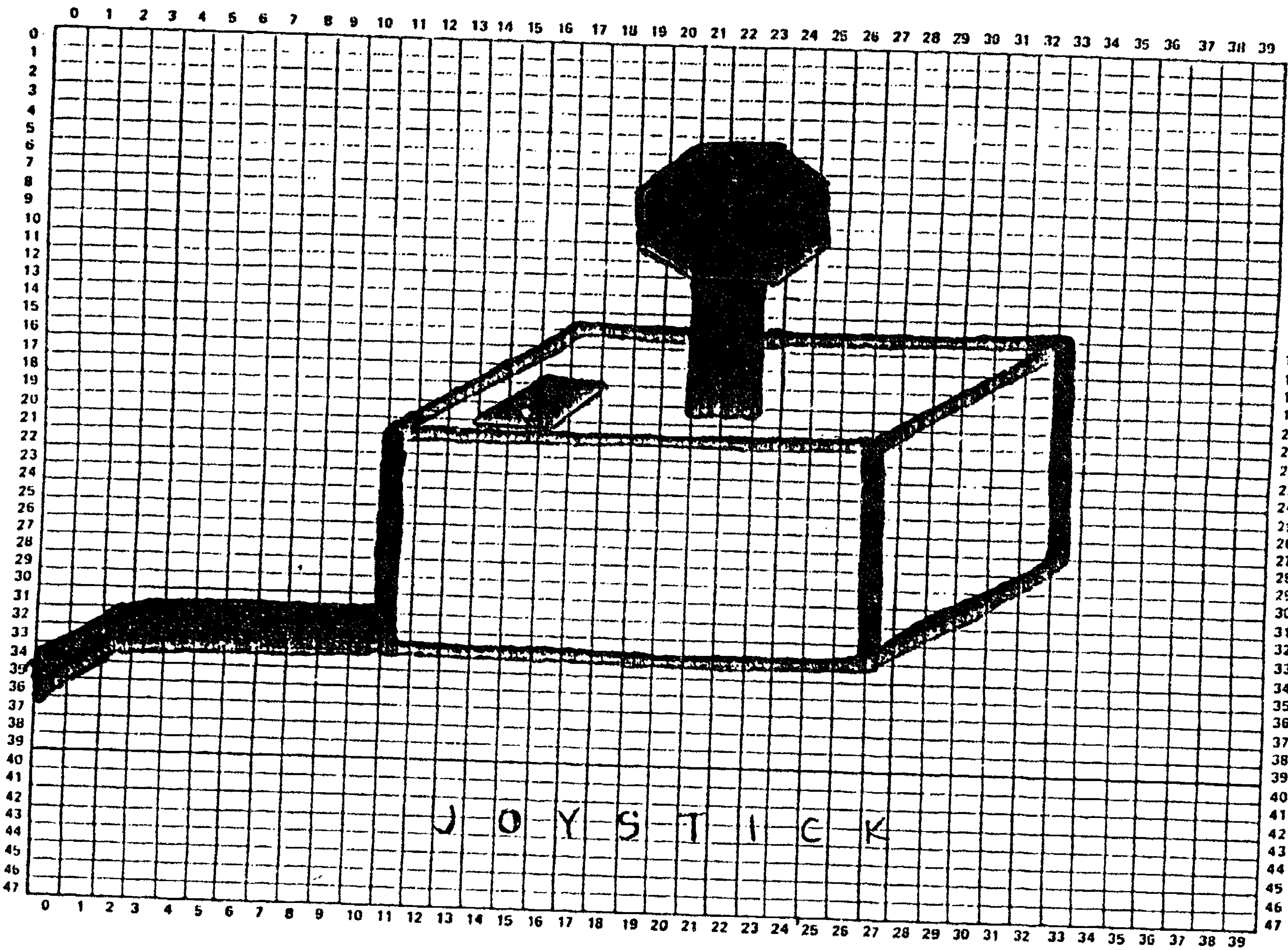
APPLE VIDEO DISPLAY WORK SHEET (LO-RES)



TITLE ______________________ PROGRAMMER ______________________ PAGE ____ OF ____

Screen 3

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

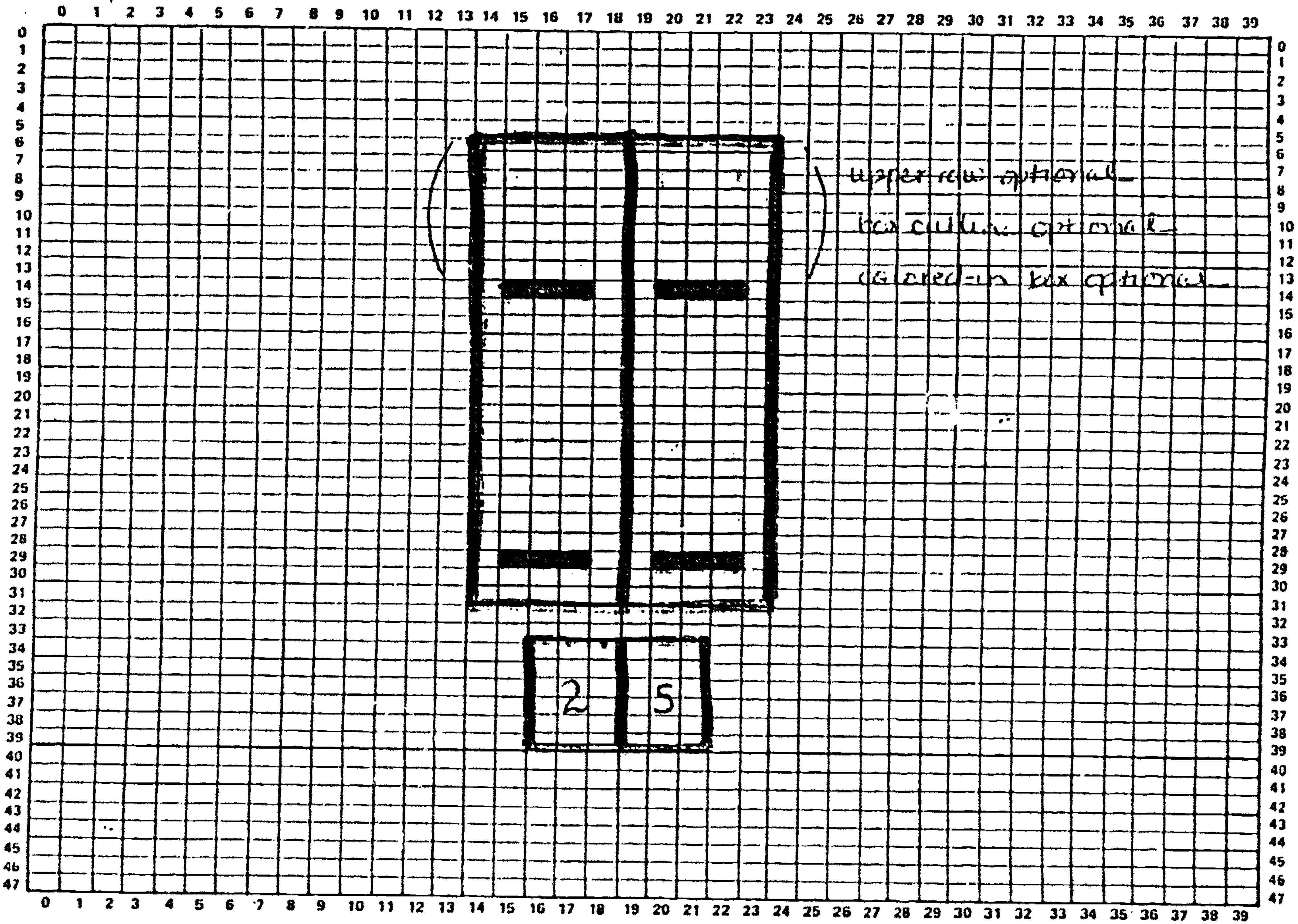


TITLE ______ PROGRAMMER ______ PAGE ____ OF ____

BPC • 225 S. WESTERN AVENUE, LOS ANGELES, CA 90004 AVAILABLE AT COMPUTER STORES NATIONWIDE

Screen 9

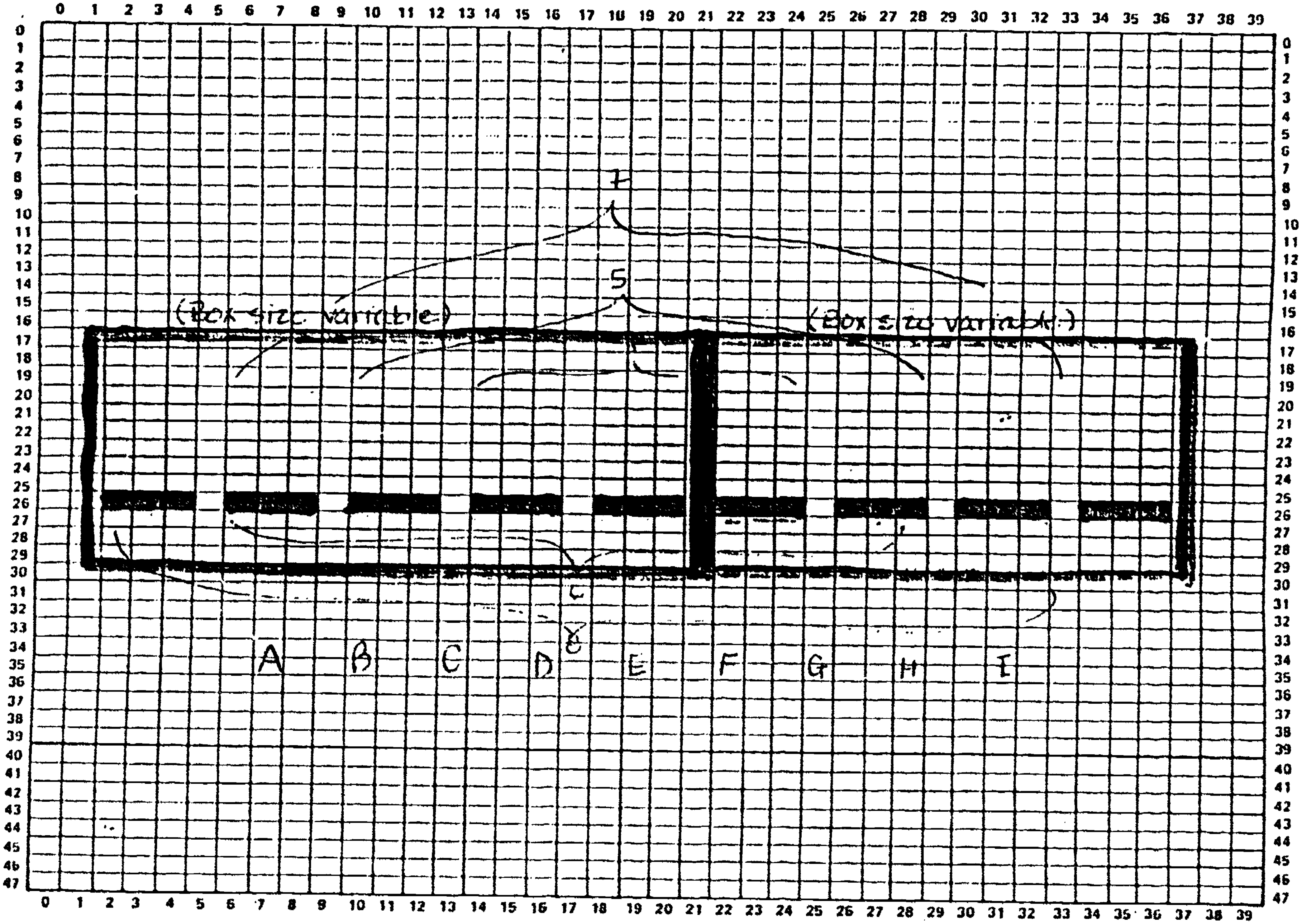
APPLE VIDEO DISPLAY WORK SHEET (LO-RES)



TITLE ______ PROGRAMMER ______ PAGE ____ OF ____

Screen 5A

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)



TITLE ______ PROGRAMMER ______ PAGE ___ OF ___

• BPC • 225 S. WESTERN AVENUE, LOS ANGELES, CA 90004 AVAILABLE AT COMPUTER STORES NATIONWIDE

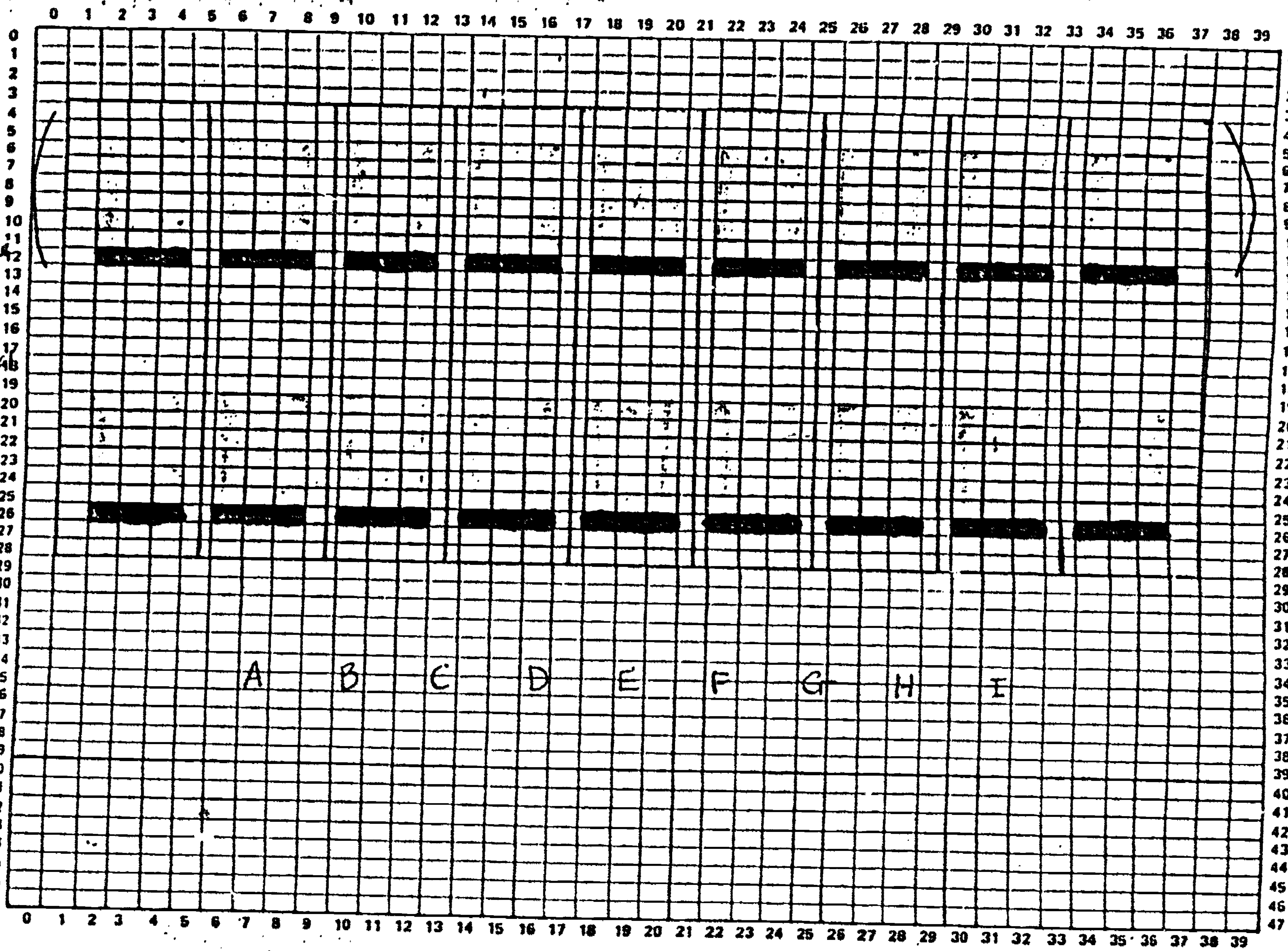
Screen 50

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

upper row optional

box outline optional

colored-in box optional

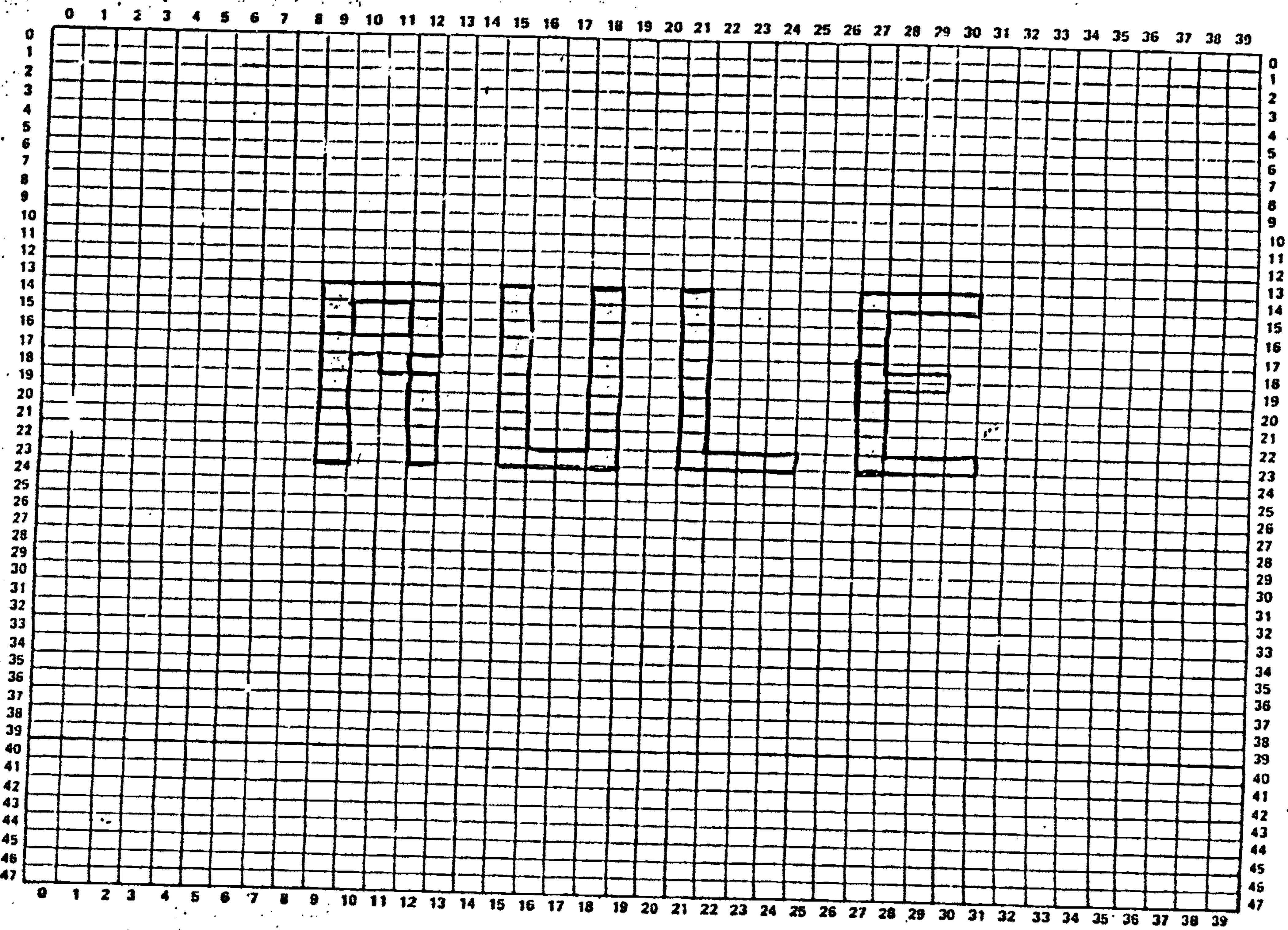


TITLE ______________________ PROGRAMMER ______________________ PAGE ____ OF ____

BPC • 225 S. WESTERN AVENUE, LOS ANGELES, CA 90004 AVAILABLE AT COMPUTER STORES NATIONWIDE

screen 6

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

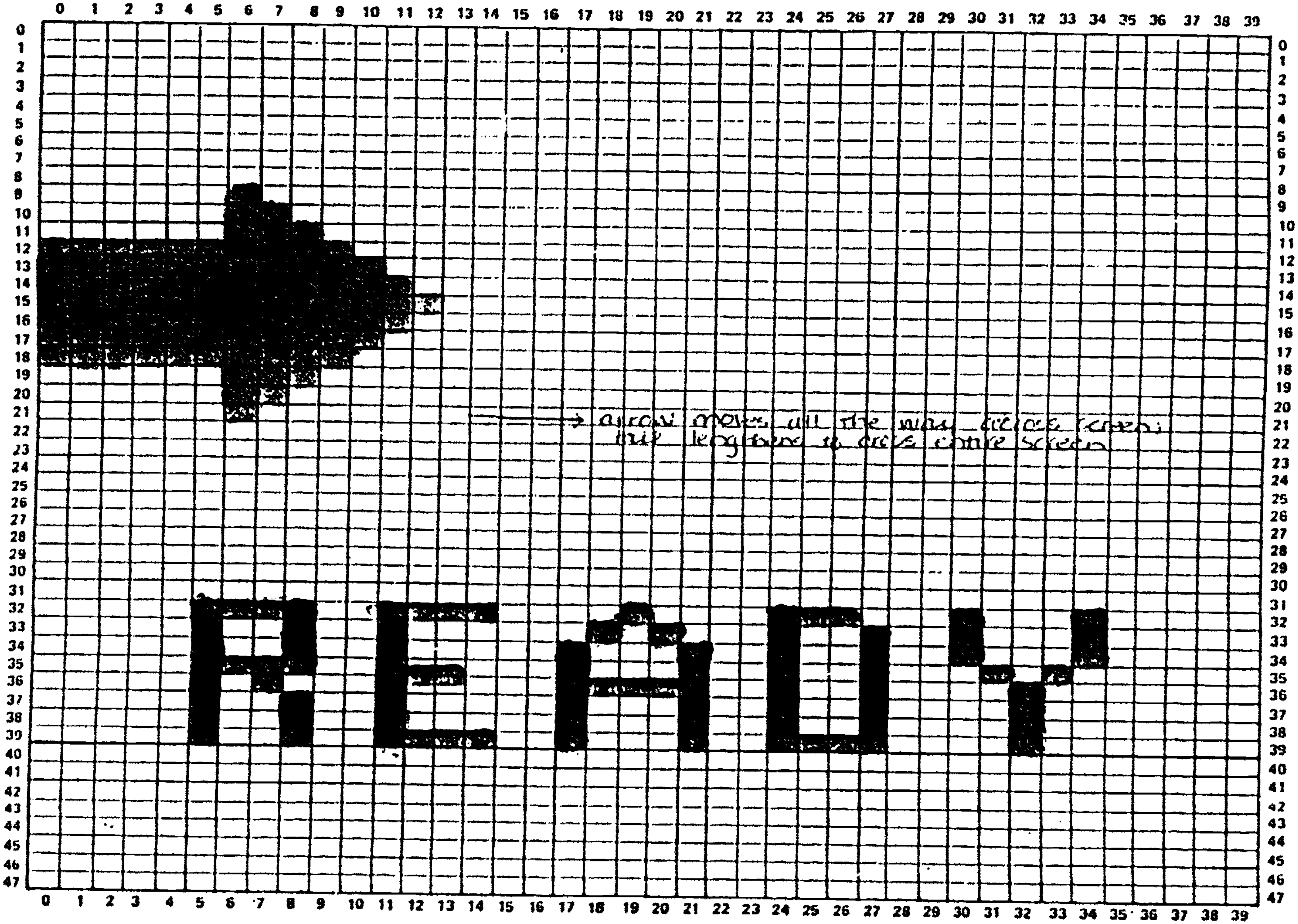


TITLE ______________________ PROGRAMMER ______________________ PAGE_____ OF_____

BPC • 225 S. WESTERN AVENUE, LOS ANGELES, CA 90004 AVAILABLE AT COMPUTER STORES NATIONWIDE

Screen 1

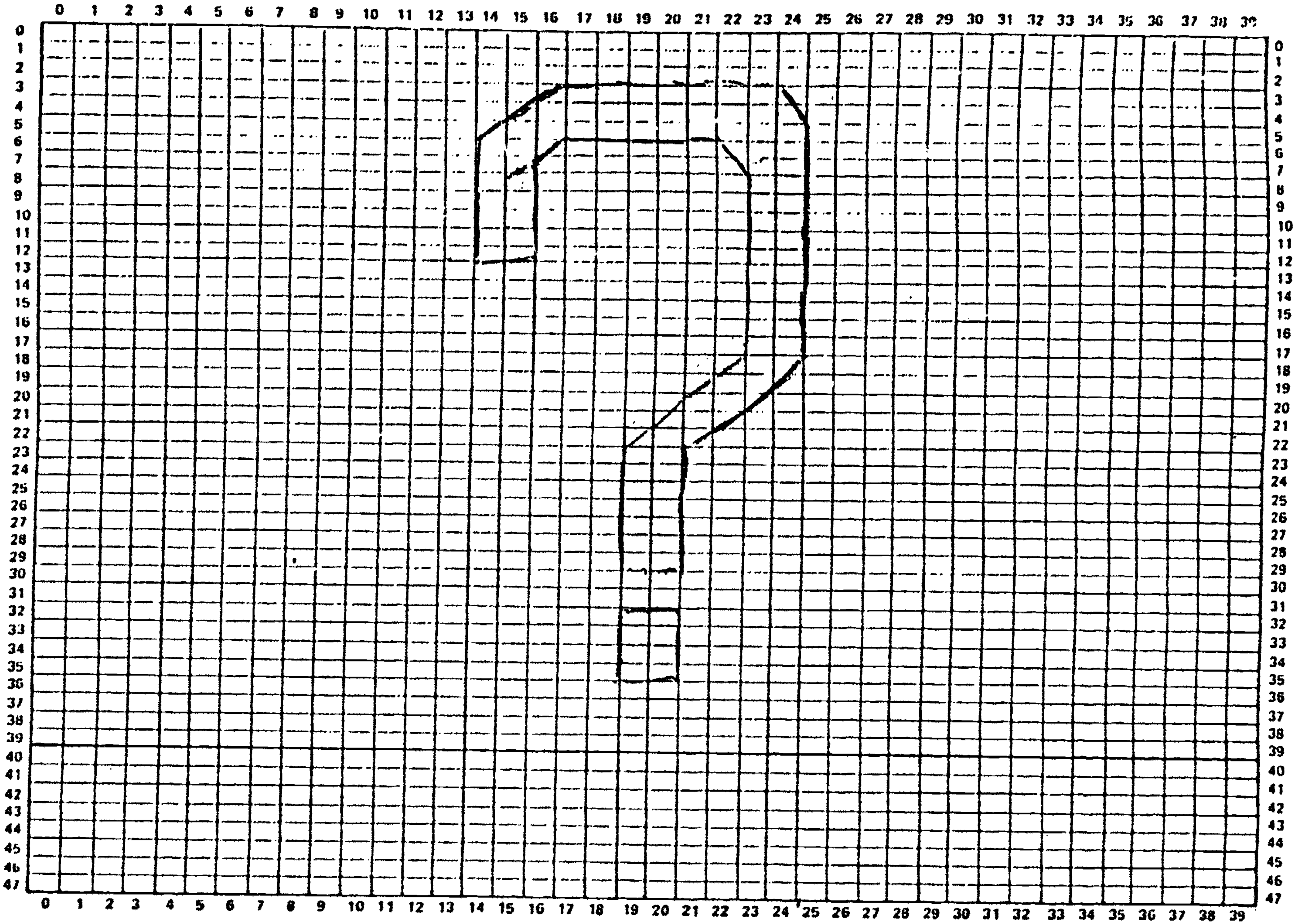
APPLE VIDEO DISPLAY WORK SHEET (LO-RES)



TITLE ______ PROGRAMMER ______ PAGE ____ OF ____ 3' 6

Screen 6

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

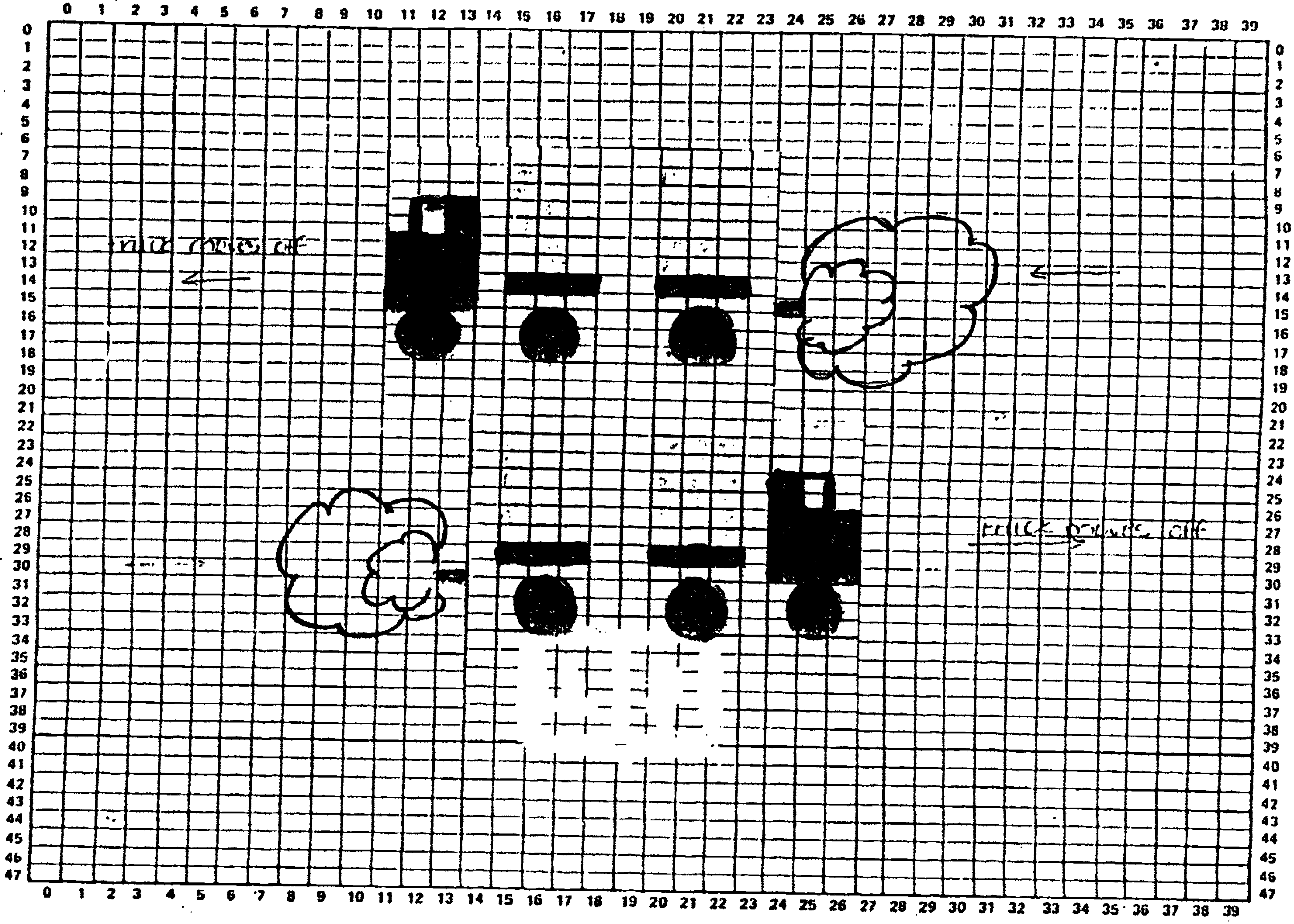


TITLE ____________________ PROGRAMMER ____________________ PAGE ____ OF ____

• BPC • 225 S. WESTERN AVENUE, LOS ANGELES, CA 90004 AVAILABLE AT COMPUTER STORES NATIONWIDE

11 A

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

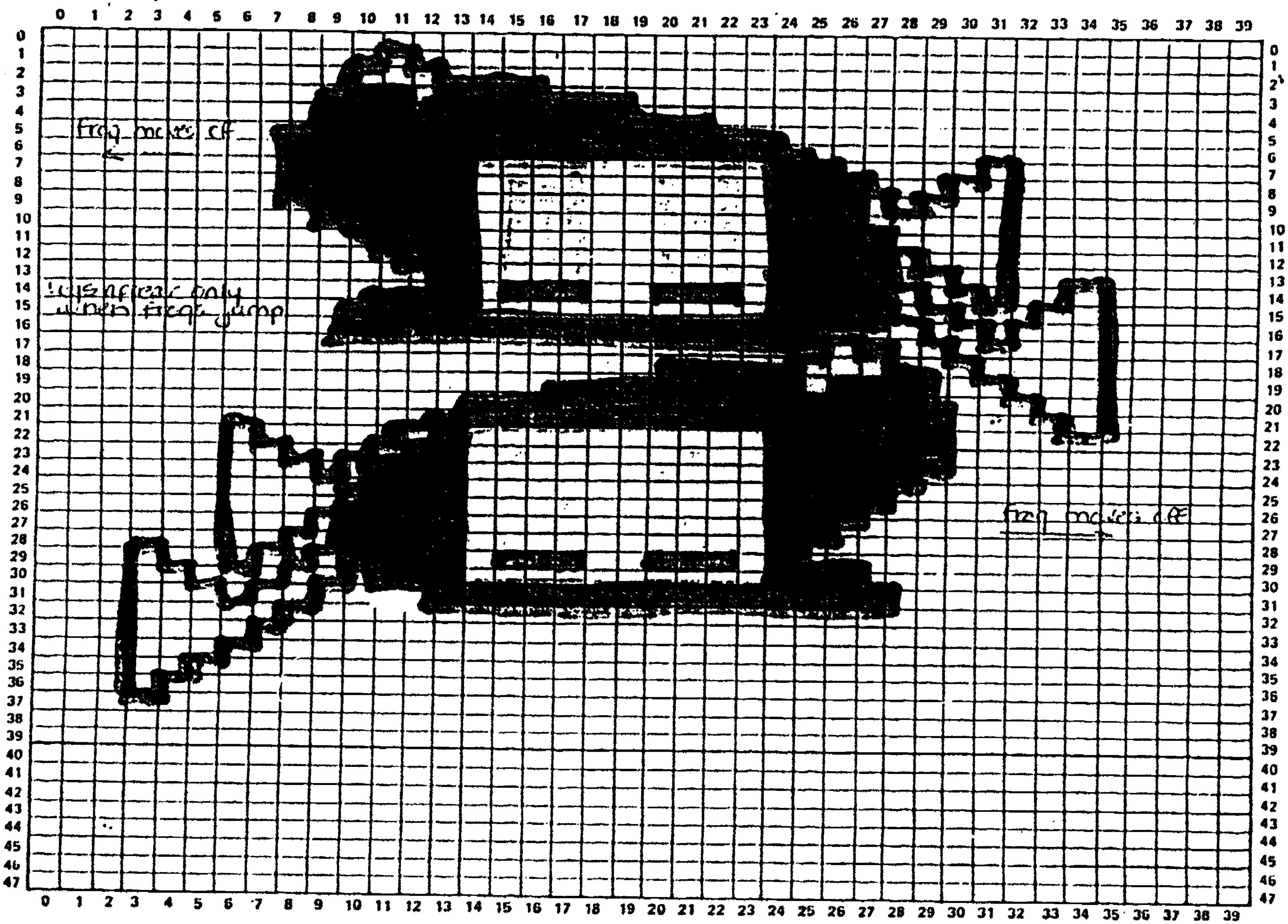


TITLE ____________ PROGRAMMER ____________ PAGE _____ OF _____ 3 0

...5 WESTERN AVENUE, LOS ANGELES, CA 90004 AVAILABLE AT COMPUTER STORES NATIONWIDE

Screen 11B

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

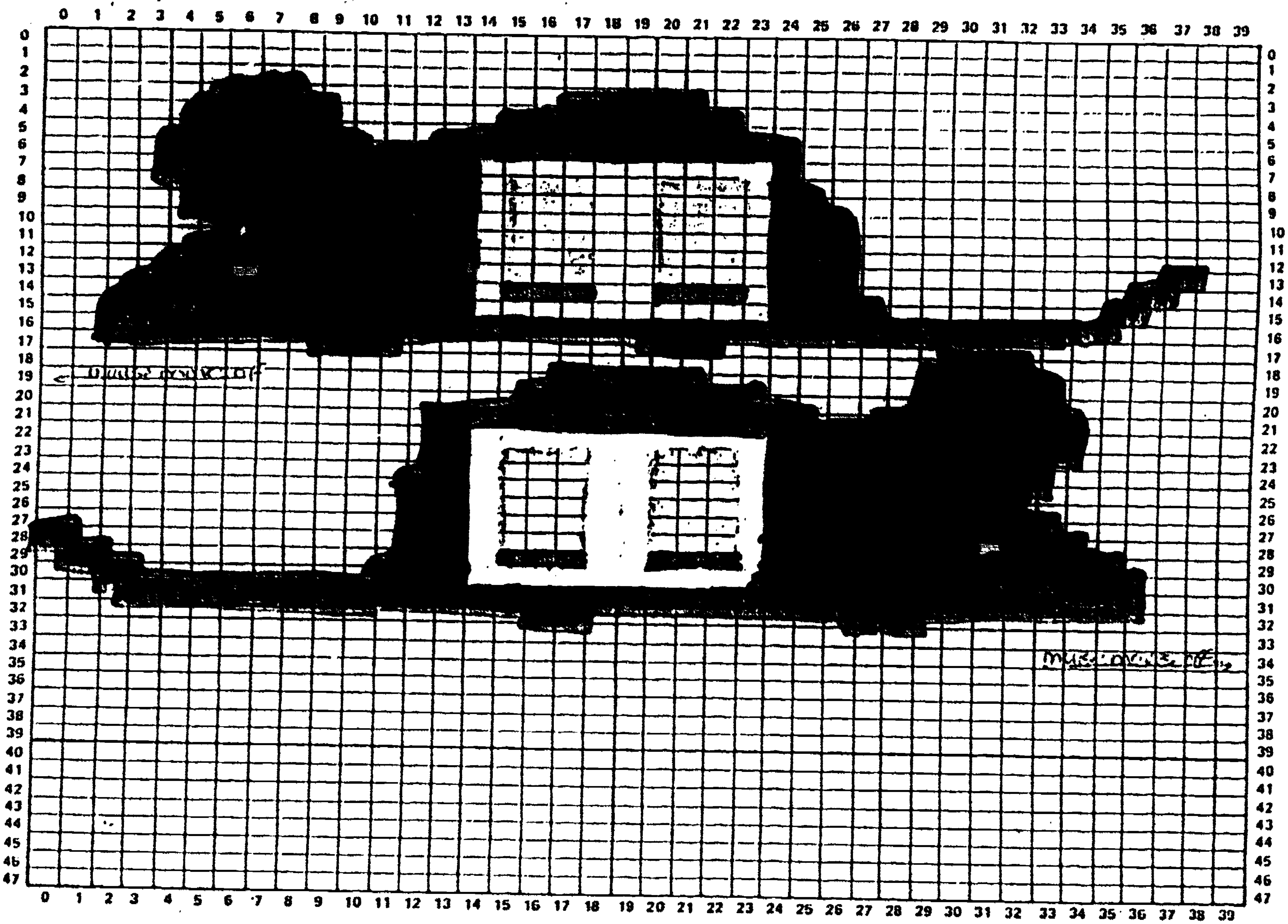


TITLE ______________ PROGRAMMER ______________ PAGE ____ OF ____

BPC • 225 S. WESTERN AVENUE, LOS ANGELES, CA 90004 AVAILABLE AT COMPUTER STORES NATIONWIDE

Screen 11 C

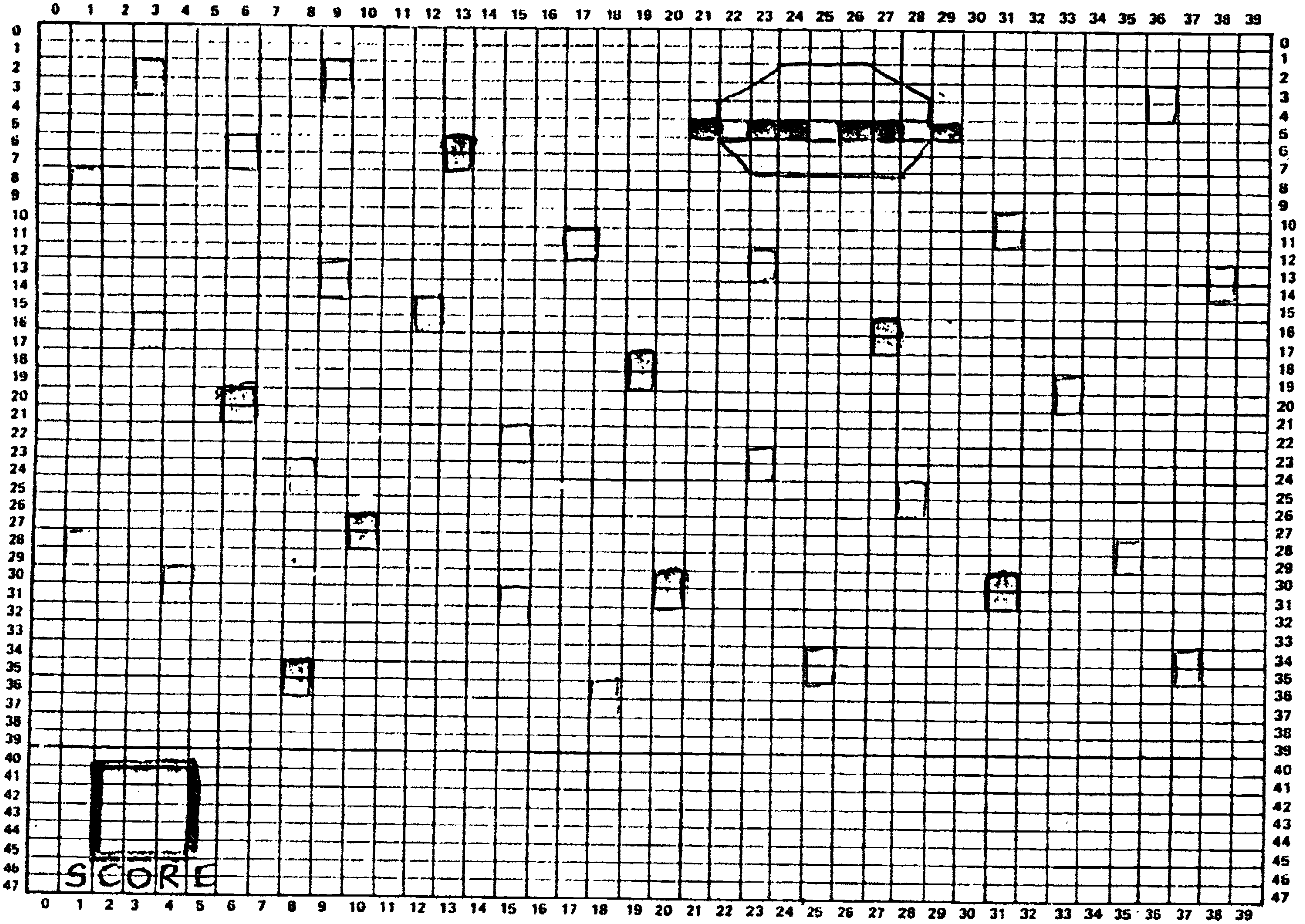
APPLE VIDEO DISPLAY WORK SHEET (LO-RES)



TITLE ______________________ PROGRAMMER ______________________ PAGE _____ OF _____

Game screen

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)



TITLE ________________ PROGRAMMER ________________ PAGE ____ OF ____

Screen 16

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

GET READY.

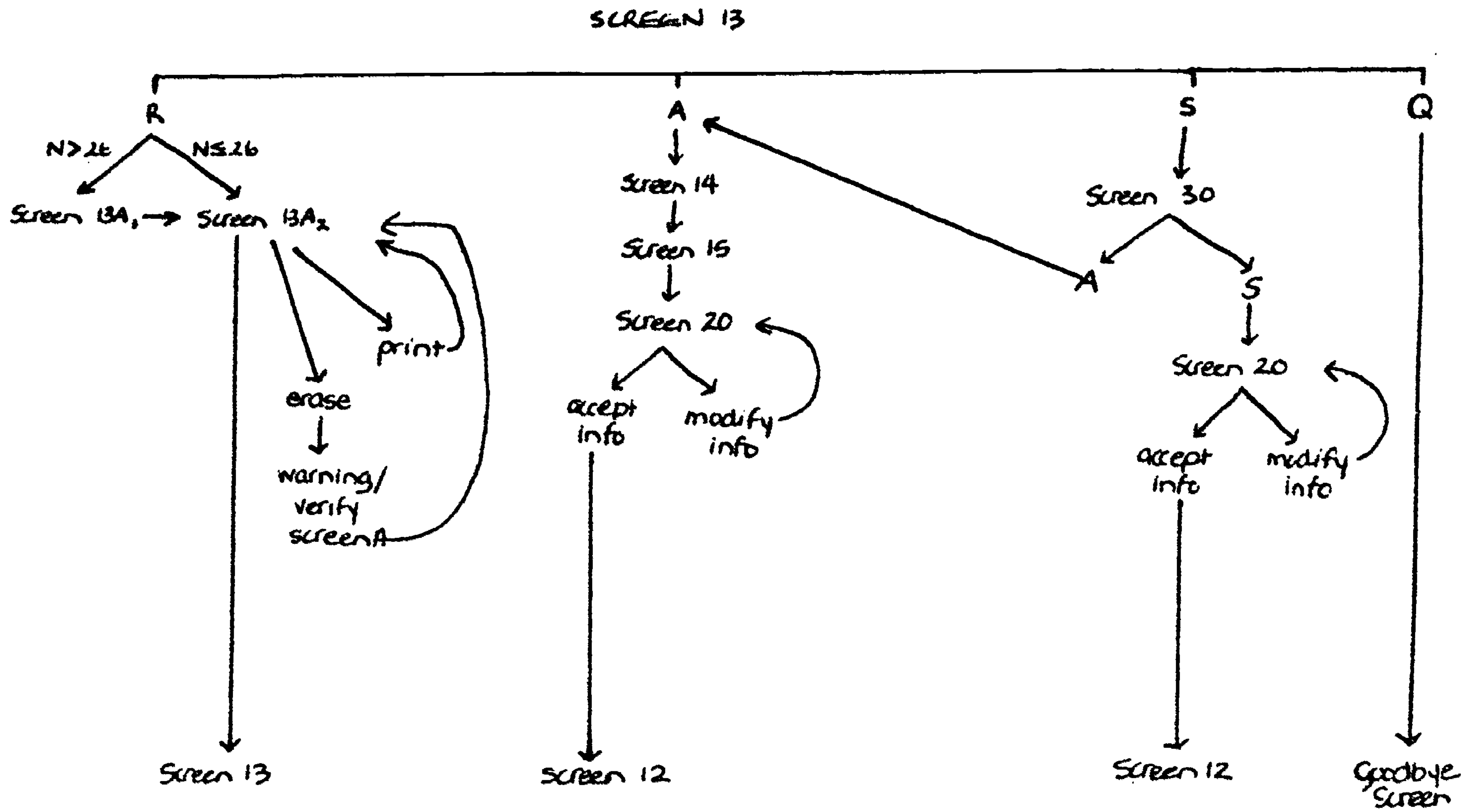
WE ARE GOING TO DO SOME MORE.

PUSH THE JOYSTICK BOTTON TO BEGIN.

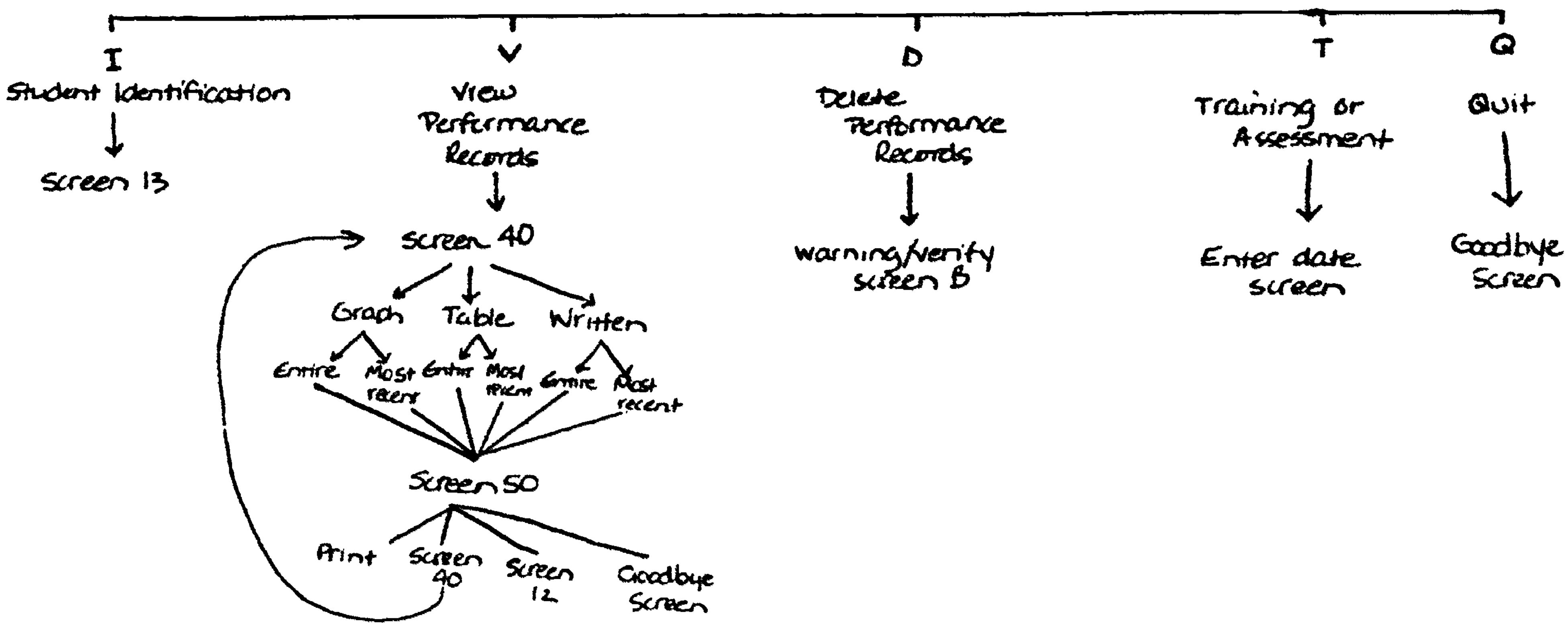
[or]

PUT THE PEN ON THE SQUARE TO BEGIN.

TITLE ______ PROGRAMMER ______ PAGE ____ OF ____


SCREEN 13
R
A
S
Q
N>26
N≤26
Screen 13A1
Screen 13A2
print
erase
warning/
verify
screen A
Screen 13
Screen 14
Screen 15
Screen 20
accept
info
modify
info
Screen 12
Screen 30
A
S
Screen 20
accept
info
modify
info
Screen 12
Goodbye
Screen

SCREEN 12


I
Student Identification
Screen 13
V
View Performance Records
Screen 40
Graph
Table
Written
Entire
Most recent
Entire
Most recent
Entire
Most recent
Screen 50
Print
Screen 40
Screen 12
Goodbye Screen
D
Delete Performance Records
Warning/Verify screen B
T
Training or Assessment
Enter date screen
Q
Quit
Goodbye Screen

Screen 14

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

ENTER NEW INFORMATION BELOW

LAST NAME:________________ (PRESS RETURN)

FIRST NAME:________________ (PRESS RETURN)

DATE OF BIRTH: MM/DD/YY (PRESS RETURN)

SEX: MALE (PRESS RETURN)
FEMALE

TODAY'S DATE: MM/DD/YY (PRESS RETURN)

TITLE 3 3 PROGRAMMER ________ PAGE ____ OF ____

BPC • 275 S. WESTERN AVENUE, LOS ANGELES, CA 90004 AVAILABLE AT COMPUTER STORES NATIONWIDE

Screen 12

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

ASSESSMENT AND TRAINING

OF MEMORY STRATEGIES

PLEASE SELECT AN OPTION:

$\boxed{\text{A}}$SSESSMENT AND TRAINING

OR

$\boxed{\text{S}}$TUDENT RECORDS

TYPE IN HIGHLIGHTED SELECTION (A, S)

THEN PRESS $\boxed{\text{RETURN}}$

TITLE ______ PROGRAMMER ______ PAGE ____ OF ____

3.6

Screen 13

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

STUDENT NAME

PRESS $\boxed{+}$ TO ENTER NEW NAME

OR

ENTER FIRST AND LAST NAMES

OF STUDENT ALREADY ON FILE

FIRST NAME: ______________ (PRESS RETURN)

LAST NAME: ______________ (PRESS RETURN)

TODAY'S DATE: MM/DD/YY (PRESS RETURN)

TITLE ______________ PROGRAMMER ______________ PAGE ____ OF ____

Screen 15

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

PARAMETERS

STUDENT WILL USE: $\boxed{L}$IGHTPEN (PRESS $\boxed{RETURN}$)

$\boxed{J}$OYSTICK

NUMBER OF DISK DRIVES: $\boxed{1}$ (PRESS $\boxed{RETURN}$)

$\boxed{2}$

VIDEO GAME SPEED: $\boxed{S}$LOW

$\boxed{M}$EDIUM (PRESS $\boxed{RETURN}$)

$\boxed{F}$AST

VOICE: $\boxed{Y}$ES (PRESS $\boxed{RETURN}$)

$\boxed{N}$O

PRINTED TEXT: $\boxed{Y}$ES (PRESS $\boxed{RETURN}$)

$\boxed{N}$O

TITLE 329 PROGRAMMER PAGE OF

Screen 20

APPLE VIDEO DISPLAY WORK SHEET (LO-RES)

PLEASE REVIEW THE INFORMATION PROVIDED.

NAME: ____________________

DATE OF BIRTH: ____________________

SEX: __________

TODAY'S DATE: ____________________

STUDENT WILL USE: ____________________

NUMBER OF DISK DRIVES: ____

VIDEO GAME SPEED: ____________________

VOICE: __________

WRITTEN TEXT: __________

IF THIS IS CORRECT, TYPE **Y** THEN **RETURN**

IF ANY PORTION IS INCORRECT, TYPE **N** THEN **RETURN** TO RE-ENTER INFORMATION

TITLE ____________ PROGRAMMER ____________ PAGE _____ OF _____

APPENDIX E

Video Game

Video Game

The setting for the game is intergalactic space; the object is to make contact with stars, with points accumulated for each star contacted. An evil alien ship, Stargrabber II, is continually moving through space capturing stars in its path.

Interludes of the game will follow each assessment portion. If the student meets criterion for the assessment, (s)h_ will receive a longer interlude, i.e., three minutes, than the one routinely provided, i.e., two minutes. The game interlude after the very first assessment will be approximately four minutes long to provide time to explain the rules of the game.

There are four levels of difficulty. The screen for each level contains 16 target stars and 16 meteors that serve as distractors. The basic screen color is black. Target stars are blue; distractors are red and yellow. At levels three and four a portion of the target stars will be shooting stars (i.e., they will move).

Level 1

The stars and meteors are stationary (i.e., they remain in the same position) throughout the interlude. Their locations on the screen are approximately evenly dispersed. The objects are illuminated for 5 seconds, then disappear for 5 seconds. The alien ship does not appear during Level 1.

Level 2

The stars and meteors are again stationary throughout the interlude. The stationary objects are illuminated for one second, then disappear for two seconds. The alien ship appears during Level 2, and remains on the screen for the amount of time necessary for it to cross the screen from left to right. The ship will travel at a rate of one inch per second.

Level 3

Twelve distractor meteors and twelve target stars remain stationary throughout the interlude; four stars and four meteors move. The stationary objects are illuminated for one second, then disappear for two seconds. The moving objects travel across the screen at a rate of two inches per second and, like the other objects, remain illuminated for only one second. They reappear two seconds later, and this reappearance may occur at any location on the screen. The movement pattern is identical to that previously described. The alien ship appears during Level 3, and remains on the screen for the amount of time necessary for it to cross the screen from left to right. The ship will travel at one inch per second.

Level 4

Eight stars and eight meteors remain stationary; the remaining ones move. Stationary objects continue to flash at a rate of one second on, two seconds off. The moving objects travel at a rate of four inches per second, remaining illuminated for one second. They reappear two seconds later, and this reappearance may occur at any location on the screen. The movement pattern is identical to that previously described. The alien ship appears during Level 4, remaining on the screen for the amount of time necessary for it to cross the screen from left to right. The ship will travel at two inches per second.

Rules of the Game

The object of the game is to collect points by contacting stars. Stationary stars are worth one point; shooting stars are worth five points. When a star has been contacted, it turns white, whistles, and then disappears from the screen. Nothing reappears in its place. The alien ship appers in Levels 2, 3, and 4, and crosses the screen in a straight line near the top; any stars in the path of the alien ship are captured by the ship and appear inside the ship for the remainder of its journey across the screen. The ship itself is worth ten points. The player has an opportunity to release the captured stars and add their point values to his/her total by blasting the alien ship before it leaves the screen. This can be accomplished by hitting the ship with a missile fired from one of three missile launch pads located at the bottom of the screen. Any stars in a missile's path are destroyed. There are an unlimited number of missiles available.

The player gets an extension of play if time remains, no more point-carrying elements remain on the screen, and the alien ship has been blasted. In the event that all but the last requirement are fulfilled, the alien ship will continue to periodically traverse the screen at varying heights until it is blasted or time runs out.

An extension of play provides the player with a new screen and play proceeds as it did at the interlude's commencement.

The playing time remaining and points accumulated always appear at the bottom of the screen. Time durations of the stars, meteors, and ship specified above relate to both response modes described below.

Durations and playing time will be adjusted in either response mode to make them of the same approximate difficulty level, as determined in pilot testing.

Mode

Light pen - player touches stars with light pen to contact them and activates missile pad by touching circle in middle of pad.

Joystick - player directs cursor to star/missile pad with joystick and presses button for contact activation.

APPENDIX F

Software Evaluation Form

Software Evaluation

We are requesting that you review this software design for its suitability, useability, and marketability, even if you do not believe it is appropriate for your product line.

Suitability

What do you perceive as the primary purpose of this software?

Is the program as written appropriate for this purpose?

What changes can be made to make the program more appropriate?

Other comments on suitability?

Useability

What aspects of the software contribute to its useability?

What aspects of the software detract from its useability?

How can these detractions be offset or changed?

Is the software appropriate for the target population?

Marketability

Whom do you regard as potential consumers of this software?

What is the commercial marketability of this software?

What changes do you recommend to improve the software's marketability?

What do you estimate to be the annual volume of sales of this software, with strong promotion and endorsement by ARC/US?

Other comments on marketability?

Please Rate the Software on this Continuum

	Low					High
Suitability	1	2	3	4	5	6
Useability	1	2	3	4	5	6
Marketability	1	2	3	4	5	6

We welcome any additional comments. Space is provided below.

Thank you very much for your assistance in the evaluation of this design.

APPENDIX G

Informed Consent Letter

Dear

Your child is eligible for participation in a research project involving the field testing of new educational computer software. This software was developed through the cooperative efforts of special educators, cognitive psychologist and computer experts, and is intended to assess specific memory difficulties and assist in their remediation.

The software is designed to test the child on a number of related memory tasks, determine the child's level of functioning, and teach strategies that (s)he can use to be more successful. The children's performance on related tasks will be evaluated before and after their experience with his software to see if they are using the strategies they learned from the computer in other activities.

There is virtually no discomfort or risk involved with this research. Since training that involves repetition can get boring, this software includes lots of things that will make learning fun, such as lots of colorful pictures and action on the screen. The computer will be able to talk to your child, and will also build in brief periods of video game playing. This should not only prevent the potential problem with boredom, but should actually motivate the children to perform better.

The children who participate can reasonably be expected to benefit from the program in several ways. First, they should experience success on those memory tasks trained directly, and they may also be able to use the strategies taught to them in other situations. In addition, their interactions with the computer and the training assistant should be satisfying and enjoyable.

This research is being conducted by the Association for Retarded Citizens of the United States (ARC/US) for the U.S. Department of Education. We encourage your interest and participation, and will be glad to answer any questions you might have.

Participation in this research is voluntary, and if you choose not to have your child participate there will be no penalty or loss of privileges for your child. You may also discontinue your child's participation at any time.

Results of this project will be shared with other parents and teachers through presentations and publications in appropriate journals; however, no written or oral accounts of this research will mention your child or any other child by name.

Project staff members will be most happy to share the research results with you. We thank you in advance for your consideration of this matter, and we look forward to including your child in this project.

Sincerely,

Al Cavalier, Ph.D.
Director, Bioengineering Program
Department of Research and Program Services

AC/co

APPENDIX B

Addendum to Final Design Report:
Commercial Publishers' Feedback

U.S. DEPARTMENT OF EDUCATION

OFFICE OF SPECIAL EDUCATION PROGRAMS

CONTRACT NO. 300-84-0156

TECHNOLOGY TO ENHANCE SPECIAL EDUCATION: REMEDIATION OF PROBLEMS IN LOGICAL THINKING AND MEMORY

Evaluation of the Program Specifications by Commercial Software Companies

PROJECT STAFF:

AL CAVALIER, PH.D.

BETH MINEO, PH.D.

CINDY OLIVER

THE BIOENGINEERING PROGRAM

DEPARTMENT OF RESEARCH AND PROGRAM SERVICES

arc National Headquarters 2501 Avenue J Arlington, Texas 76006 (817) 640-0204

Association for Retarded Citizens of the United States

Representatives of three educational software companies evaluated the ARC's computer-based instructional system in terms of its suitability, useability, and marketability, using the form included in Appendix F of the Final Design Report. Their evaluation and comments are summarized below.

All evaluators perceived the training of memory/recall skills as the primary purpose of the software and agreed that the program as written is appropriate for that purpose.

All evaluators stated that the software's useability is enhanced by the fact that it is based on solid research and a well-developed theory in cognitive psychology, unlike 99% of all other special education software, and by the integration of extensive voice output and the inclusion of a powerful assessment capability.

The only factor noted as potentially detracting from its useability was the amount of repetition. Two evaluators noted that it will be important to assess the motivational power of the program since a fair amount of repetition is incorporated into the design, and there is always a fine line between insuring learning and losing attention through repetition. One evaluator suggested that the repetition could be offset by couching it in a fantasy or game. After reviewing this suggestion with the Program Design Consultant, project staff concluded that the integrity of the basic paradigm that has evolved from the cognitive theory and supporting research would be compromised if such features were incorporated. Another evaluator predicted that adults could be considered to be potential users if some of the aspects aimed at the juvenile population were modified. For instance, the use of animated graphics as reinforcement could be replaced by a scorecard on which percent correct is displayed. Project staff and the programmers are looking at these options at the present time.

Evaluators identified public and private institutions, schools, families, and user groups as consumers of the software. They saw potential for its use in both early learning and special education school programs. They characterized favorably the commerical marketability of the software, with comments ranging from "good" to "excellent". The evaluators offered some suggestions for changes that would improve the software's marketability; these included (a) shifting responsiblity for explaining the operation of the hardware from the program to the teacher, (b) creating an MS-DOS version, and (c) incorporating an authoring system to allow for customized vocabulary. Evaluators were reluctant to project sales volume estimates, although one stated that "sales in the thousands per year are realistic". This person also suggested that if the product and marketing were aimed at the consumer level usage, the numbers could be increased by one or two orders of magnitude.

On a scale from 1 to 6, with six being the most favorable rating, the evaluators gave the software an average rating of 5.3 for suitability, 5.7 for useability, and 6 for marketability.

Educational Software Companies that Evaluated the Software Package:

Laureate Learning Systems
One Mill Street
Burlington, VT 05401

Life Science Associates
One Fenimore Road
Bayport, New York 11705

Jostens Learning Systems
600 West University Drive
Arlington Heights, IL 60004

APPENDIX C

User's Manual

Introduction

Rationale

Memory is vital to learning. In fact, many times what appears as an inability to learn may in fact be a problem in remembering. Memory processes are complex, involving perception, encoding, storage, and retrieval of information. Research has shown that there are actually two types of memory: short-term memory (STM) and long-term memory (LTM). Short-term memory is limited in capacity and relatively brief in duration, i.e., approximately 30 seconds. Success in dialing a telephone number that one has just looked up in the telephone book but the failure to recall it 30 minutes later is an example of the use of STM. Long-term memory, on the other hand, is considered to be of unlimited capacity and of permanent duration. Recalling the telephone number of one's childhood home is an example of LTM.

To be able to remember something, a person needs to know whether it should be stored in STM or LTM, and how it can be transferred from STM to LTM. These factors determine what type of voluntary rehearsal or encoding strategies an efficient learner employs to store the necessary information in LTM for later retrieval and use.

The software developed by the ARC is not curriculum-specific but instead focuses on some of the fundamental cognitive skills that underlie learning and performance across every content area. The software is structured around a memory task that is frequently used in assessment and instructional applications and that requires many of the same cognitive strategies for successful performance that underlie efficient information processing across a wide variety of situations (Latham, 1978).

The instructional package developed in this project is based upon the <u>ordered recall</u> task. In the ordered recall task, a student is requested to recall in the order presented a list of items that s/he has seen only once. The items are serially-presented, with only one item exposed at a time. The student is asked to first recall the subset of the last items presented (the terminal items) and then circle back and recall the subset of the items that were presented first (the initial items). This aspect of the task is called "circular recall" (Butterfield, Siladi, & Belmont, 1980). If a student was told to remember the string L,T,Z,J,R,P,F, s/he could employ a circular recall strategy by recalling R,P,F, and then circling back to remember L,T,Z,J. This would be referred to as 3/4 circular recall because the student remembered first the last three elements and then the first four. In the ordered recall task, each item is displayed for a fixed period of time (e.g., 0.5 seconds), but the student controls the pace of the presentation, i.e., the timing of the presentation of the next item. Task difficulty and memory load can be varied by changing the number of items in the to-be-recalled list and the type of items to be recalled (e.g., letters, numbers, words).

Extensive research has shown that performance on the terminal items reflects a student's STM abilities, and the task permits precise manipulation of variables that pertain specifically to STM limitations and instructional strategies. Similarly, performance on the initial items reflects a student's LTM abilities and is sensitive to a number of manipulations directly related to strategic cognitive activity (Belmont & Butterfield, 1969, 1971a, 1971b; Brown & Barclay, 1976).

One of the most revealing measures of strategic cognitive activity in this task is the length of time the student <u>pauses</u> after the presentation of each item in the list. Non-retarded, non-learning-disabled students generally exhibit high recall accuracy when their pauses steadily increase across the initial items, followed by very brief pausing over the terminal items. The increased pausing over the initial items reflects silent cumulative rehearsal of the growing list after each new item is presented since these are the most difficult items to remember.

<u>Program Overview</u>

Students will participate in two general types of activity in the program: <u>assessment</u> and <u>instruction</u> at different levels of difficulty. The program is ordered such that the student always receives the assessment section first. If s/he meets a passing criterion on the particular level of assessment, s/he automatically progresses to the next level of assessment. If the student fails to reach criterion on assessment, instruction commences at that level. The last instructional loop at each level includes a reassessment. This cycle repeats until the student fails to reach a criterion after three consecutive instruction/reassessment sequences or s/he reaches criterion at the highest level of assessment.

In the instructional mode, students receive training on effective cognitive strategies on the recall task. In the early instruction trials, the computer models correct performance. Subsequently, assistance is systematically withdrawn until the student is performing the task independently. The cycle of assessment and instruction continues until the student has attained his/her maximum level of performance or completed the highest level of instruction contained in the program.

In the instructional portion of the program, the circular recall task is disassembled and each of the four components of the effective strategy is individually trained. The student then learns how to combine the component strategies into integrated performance on the target task. The first component, known as "fast finish" training, teaches the student to retain the terminal set of items first by quickly memorizing them in a chunk. The second component, known as "cumulative rehearsal" training, shows the student how to memorize the first (and more difficult) set of elements by retrogressively rehearsing all previous elements in their original order as new ones are revealed. The third component, that of "interpolated delay and self testing", trains the student to hold those items memorized with cumulative rehearsal in memory for the amount of time equivalent to that needed to complete the fast finish on the terminal set. The final component, that of "chaining", teaches the student to incorporate the components into a unified strategy.

This software has been designed to advance through a general hierarchy of difficulty posed by different circular recall requirements. The requirements addressed in this program are in the estimated order from easiest to most difficult:

Program Level	Circular Recall Pattern	Number of Elements
A	2/2	4
B	3/2	5
C	3/3	6
D	2/4	6
E	2/5	7
F	4/4	8

The student begins assessment and instruction at the first level, that of a 2/2 circular recall requirement. There are two basic types of information that will be recorded on each student: accuracy of recall and pause-time pattern. Recall accuracy represents the number of items recalled correctly on a trial by the student. Pause-time patterns reflect the amount of time a student waits after seeing each item before displaying the next one. Thus, pause time corresponds to the amount of time spent committing an item to memory. Since long strings of items take longer to rehearse than do short strings, this would be reflected in corresponding differences in pause times. Circular recall requires the student to cumulatively rehearse certain elements; if pause times between elements do not vary, it is an indication that the student is not using the strategy. To reach criterion on assessment or any portion of training, the student must meet only an accuracy criterion, which indicates that the student is able to remember the specified number of items. Pause-time information, which indicates that the student is employing the appropriate cognitive strategy, is collected but not used to determine whether or not the student moves on to the next level. In the assessments, the student receives three trials that are used to compute the performance data. In the instructional portion, four strategy components are taught. Performance on each component must reach criterion before the student can proceed to the next component. A block of three trials at the end of each component is used for assessing mastery.

Features

This system has a number of features that make it attractive from an educational and technological standpoint. The software was designed by professionals who understand young people and the ways in which they learn. It makes use of the computer's many abilities in order to make the interactions between student and system interesting, motivating, and educational.

- The student can elect to respond with either a light pen or a joystick.
- The program automatically begins instruction at the level at which the student first has difficulty.
- The student receives playing time on an intergalactic video game as a reward for hard work.
- Multimodal output (graphics, voice, text) provides important redundancy for the student.
- The system is based on a training strategy proven effective in numerous laboratory research studies.
- Data is automatically gathered and analyzed by the computer as the student interacts with the system.
- Information is provided teachers and parents via graphs, tables, and written interpretations of the results.

USER GUIDELINES

Using the Software for the First Time

Begin by familiarizing yourself with the software package. The package that has been provided to you consists of four floppy disks. These are the system disks, and contain the "workings" of the package. You will need to prepare a data disk; this disk will contain all of the student data.

Preparing Data Disks

Your first task is to initialize the data disk to prepare it to receive student data. With the computer turned off, place Disk #1 of the system disk set in Drive #1, and place a disk formatted for your particular computer in Drive #2. Turn the machine on and wait a short period while the program is loaded in. When loading is complete, you will see a menu of choices that looks like this:

(Screen of Main Option Menu)

To prepare the data disk, select the choice designated as "E". This selection is also used when you want to erase an entire disk of its accumulated data. You will see a message asking you to verify that you really intended to select this option because of the potential for destruction of any data files existing on the disk. Answer "yes" to both queries, and the system will take several seconds to clear the disk of any old data and prepare it to receive new data. Following completion of this operation, the system will return a prompt that looks like this:

A>

Should you wish to continue beyond the initialization of the data disk, you will need to re-boot the system. You may do this by typing **OVCTT** or by pressing the **CONTROL**, **ALT**, and **DEL** keys simultaneously.

Program Options

Having done this, you will again see the Main Option Menu. The options available at this point are:

I This allows you to proceed to another menu presenting student information options.

V This takes you to the portion of the program that permits you to view table, and graphic representations of student performance.

D This allows you to delete individual student files.

E This erases an entire disk or initializes a new disk.

T This permits a student to enter the assessment and training portions of the package.

Q This closes all files and permits an orderly exit from the program.

These options will be explained in greater detail on the following pages.

Student Identification

The software provides a sequence of assessments and training blocks to students, and also allows teachers and parents to review student progress. **VERY IMPORTANT**: To be able to access these capabilities, a student name first must be provided. The Student Identification Screen permits:

1. The selection of a student already on file;
2. The addition of a new student name into the file; or
3. The review of a list of all students on file.

(Screen 13)

Typing **R** allows you to review the list of students on file.

Typing **A** allows you to add a new student to the file.

The information needed for a new student includes name, birthdate, sex, and parameters of the students interaction with the system. These include the input mode (whether the student will use a lightpen or joystick), the desired videogame difficulty level, and the output mode (voice, text, or both). This individual information needs to be entered only once unless you desire to change some aspect of it later. After this initial input, this tailored information is called up automatically each time the student's name is selected at the beginning of the program.

(Screen 14)

(Screen 15)

After this information has been entered, you will have an opportunity to review it.

(Screen 20)

If the information is correct, type **Y**. If any of the information is incorrect, type **N**. The cursor will return to the first line of information. If the last name is correct, press **RETURN** and the cursor will move to the next line. If the last name needs to be corrected, type the correct version over the incorrect one then press **RETURN** to accept the new version. Continue in this manner until all information presented is correct.

(Screen 13)

Typing **S** allows you to select a student already on file.

(Screen 30)

Every student who has interacted with the software previously is contained in the file unless his specific information was deleted. The Student Selection screen contains a window through which all the names on file can be scrolled. When you type in the name of the student you wish to select, the software automatically matches it up with the most similar name on file, and that name appears in the scrolling window. When you correctly type in the student's name, the window's function is not necessary. If you should misspell the student's name, the closest match will appear in the window, and that match will most likely be the name you intended to type. In most programs, a misspelling will prevent the computer from locating the desired file. On the other hand, this software compensates for spelling errors, and can actually save time. For example, to access the records of "Clayton Oliver", you would only need to type O and the system would automatically display a student name beginning with O. Even if there is more than one O entry, the **I** and **M** keys can be used to scroll up and down the list to locate "Oliver".

When the desired name appears in the window, type **S** to select that student. You will then have a chance to review the information entered previously about the student. Any changes to the parameters, such as increasing the videogame difficulty or changing the input mode from lightpen to joystick, can be made at this point. **NOTE:** The option of changing parameters is beneficial in that it allows you to tailor the software to the abilities of each student, yet be advised that changing parameters may alter the task to some degree. If research reliability is desired, the parameters should not be altered once they have been set.

If you cannot locate the desired name in the scrolling window, it means that the student is not on file. In that case, you will need to type **A** to add the student's name to the file.

(Screen 13)

Typing **M** allows you to return to the Main Option Menu.

Selecting A Program Option

After selecting a student, you will need to indicate whether you want to review the student's performance record, have the student participate in training/assessment, or select another student.

<u>Student Identification</u>

(Screen 12)

Typing **I** will return you to the Identification Screen, allowing you to select another student or view the list of all students on file.

<u>Viewing Performance Records</u>

(Screen 12)

Typing **V** will allow you to select the form in which you would like to view student performance records.

(Screen 40)

You first get to select the form in which performance records will be presented. Typing **G** will result in records being presented in graph form; typing **T** will give you records in table form. Typing **I** will result in the presentation of an interpretation of student performance. These interpretations clarify the relationship of various aspects of the task and provide teachers and parents with a verbal account of student behavior on the task.

If you select the **G** option, you will next be asked whether you would like a graph of pause times or accuracy, and if you would like a printed copy of your selection.

(Screen 50)

You will then be required to select the circular recall level that you desire to be graphed by using the selection window. This allows you to scroll through the available options using the I and M keys. When your desired selection appears in the window, type S and your selection will be registered. Printouts are available for performance on the last three trials of each assessment, and you will be asked to specify which trial's data you wish to view.

(Screen 51)

(Screen 40)

Selecting the T option provides you with a representation of student data in table form. You will be asked to specify whether you wish to view a table containing data from the level most recently completed (select L) or for all levels completed (select A). You will also be given the option for a printed copy of the table you select.

(Screen 52)

Selecting the I option provides you with a narrative interpretation of student performance for a particular level. Again you will be given the option for a printed copy of the interpretation.

(Screen 53)

(Screen 40)

You may cycle through these options as many times as you wish. When viewing/printing of records is complete, type M to return to the Main Option Menu.

<u>Deleting Individual Student Files</u>

(Screen 12)

Typing D will delete the records of the student whose name you selected in the previous step.

As opposed to the function described next that permits the entire data base to be erased, this function allows the deletion of a single student's file. This may be the option of choice when extra space is needed in the data base or the student has completed the entire instructional sequence and his data has been printed.

The computer will issue a warning when this option is selected since deletion of a file is a permanent alteration and the information cannot be recovered. You will be asked to verify that the selection is the one you intended.

(Warning/Verification Screen B)

Erasing All Files/Initializing New Disk

(Screen 12)	Typing E erases all the files on the data disk. This option is also used to initialize a new data disk.

Typing E erases all the data stored for each student on file. It should only be used when it is necessary to free the entire disk for a large amount of new data for new students, or when you desire to initialize a disk that had not been used with the program previously. Due to the severity of the consequences resulting from the selection of E, the computer will doublecheck that your selection of E was intentional by asking you to verify the selection.

(Warning/Verification Screen A)

Selecting Training/Assessment Option

(Screen 12)	Typing T begins the training/assessment portion of the program.

You will first be asked to enter the date (month, then day, then last two digits of the year) then press RETURN-. Use zeroes to fill in if the month or day is a single digit (e.g., March 5, 1986 would be entered as 03/05/86). If you type in a nonsense response for one of the items, the software will not advance to the next item, but will ask you to type another response.

(Date Screen)

Following the input of the date, the program will ask you to select the difficulty level at which the student should begin using the selection window.

(Screen 20)

The software automatically marks where the student's previous interaction ended; to start up at this point, simply type **S** to select "Current Level". You may choose to bring a student into the system at any level; however, **THE ONLY WAY TO GUARANTEE THAT ALL DATA WILL BE AVAILABLE FOR PRINTOUTS IS TO PROGRESS THROUGH THE SYSTEM IN AN ORDERLY FASHION.** The best way to insure this is to always select the "Current Level" option at this point. After selecting a starting point that will bring the student into the training portion of the software, you will see a message to change the disk in Drive #1.

Quitting the System

(Screen 12)

Typing **Q** allows you to quit the programand places you back in the MS-DOS operating system.

A Student's First Interaction with the Software

The student's first exposure to the software is intended to familiarize him/her with the input mode selected for use (light pen or joystick) and the manner in which to respond to the various instructions given by the computer.

(Screen 1) (Screen 3)

1. The student is shown a picture of the input mode s/he will use.

(Screen 2)

2. The student learns how to manipulate items on the screen using the light pen/joystick.

(Screen 5)

3. Correct responding to the conventions of the program is demonstrated for the student. This assistance is systematically withdrawn until the student is interacting independently with the program.

The student is instructed to:

a. "Light up" the numbers in the boxes by activating each box with the joystick/light pen.

b. Always work from left to right when lighting up the boxes. If the boxes are activated in the wrong order, they will not light up.

(Screen 6)

c. Pay attention when the word "Rule" and the flashing stars appear on the screen because they signify the impending announcement of a new recall rule.

(Screen 8)

d. Recall the numbers s/he saw when three short tones are sounded.

e. Place numbers in the boxes in accordance with the recall rule. Placement position is predetermined by the recall strategy being assessed; for instance, if the strategy calls for the student to recall first the item in the third box, the first letter selected would automatically go into the third box. The student may not correct errors of placement, but s/he may use a letter more than once.

As mentioned above, the student's first exposure to the workings of the software is through the computer's modeling of a correct interaction. The cursor operates in concert with vocal and graphic cues (spoken directions, color changes, and flashing boxes) to direct student attention to the relevant aspects of the presentation. This assistance is gradually withdrawn: first, the cursor is removed, leaving the flashing boxes and vocal cues; second, the vocal cues are removed, leaving only the flashing boxes; finally, all cues are removed, This hierarchy of levels of assistance is employed throughout the program.

Assessment

Once the student has been familiarized with the workings of the software, the program moves on to the assessment portion. In this section the student gets six opportunities to employ a designated circular recall strategy to remember a list of alphabet letters. No assistance is provided to the student during the assessment portion of the program because the purpose of the assessment is to determine whether the student can independently employ a designated circular recall strategy.

If the student performs at or above a predetermined criterion of recall accuracy, s/he progresses to an assessment at the next-highest level of circular recall. If s/he fails to meet criterion, instruction in the use of that particular circular recall strategy is begun. After assessment at each level, the student receives a few minutes of playing time on an intergalactic video game. The student's first interaction with the game begins with a presentation of the rules; this description is not repeated after this point.

Assessment involves:

illumination of the boxes from left to right by the student,

study of the letter in each box for whatever amount of time the student decides to study and,

replacement of the letters into the boxes according to the designated recall strategy.

If, while studying a letter, a student takes too long, the computer will beep at him/her after 25 seconds to prompt him/her to illuminate the next letter. During recall of the letters, the computer will beep at the student if s/he delays longer than 45 seconds before selecting his/her next response.

Instruction

Instruction is begun at that level when the student fails to meet criterion during the assessment. Instruction breaks the circular recall into its component parts -- recalling the initial items and recalling the terminal items -- and teaches each separately. Then the student is taught to coordinate these strategies.

(Screen)

The first component of the instructional sequence involves training the student to pace quickly through the terminal items.

(Screen)

The <u>second</u> component has the student cumulatively rehearsing the growing list of initial items as each one is presented. Integrated into this component is the practice of self-checking in which the student mentally tests himself/herself to be certain of his/her accurate retrieval of the subset prior to exposing the next item in the list.

(Screen)

The <u>third</u> component is the introduction of a delay between the last item seen by the student and the beginning of his/her recall attempt. This delay is to insure that in practice the initial items are successfully recalled from LTM only, and also to enhance the student's understanding of the necessity for active rehearsal of the initial items.

(Screen)

The <u>fourth</u> component instructs the student to put all of these cognitive strategies together and provides practice on the smooth coordination of the strategies.

Practice on each of these four components is broken down into the four levels of assistance discussed earlier. If a student has difficulty at a level in which little assistance is provided, the computer will revert down to the simplest level and work back up the hierarchy of difficulty. A student is allowed to cycle back through the hierarchy three times; if s/he fails to be successful at completing a particular component (e.g., the terminal portion of a 3/2 circular recall) after three cycles through the hierarchy, interaction with the computer is terminated.

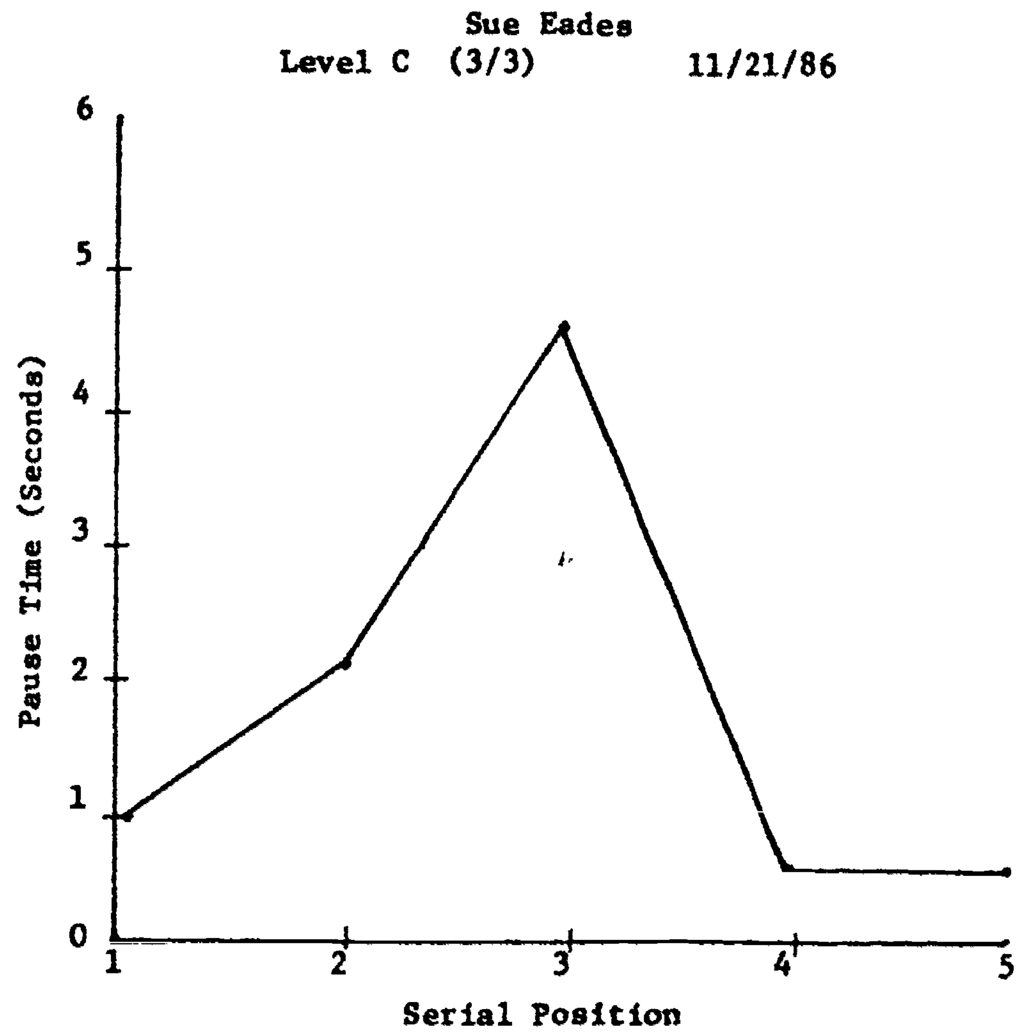
Additionally, the student's interaction with the system can be terminated at any time by pressing the **CONTROL** and **BREAK** keys simultaneously. Then, the next time the student works with the program, assessment/instruction begins at the level at which the student was working when interaction was halted.

<u>Presentation of Results</u>

Two types of data are being recorded as the student interacts with the software. The first type is a record of <u>pause times</u>, revealing the amount of time the student spent studying each letter before illuminating the next one. It is through a comparison of a student's pause time pattern with one known to be ideal for a particular circular recall requirement that allows the computer to decide if the student is using the recall strategy correctly. The second type of data being computed is a recall accuracy measure (that is, the number of items the student is correctly sequencing during recall).

The computer tallies the number of times the student cycles through each level of assistance (modeling, voice/graphic cues, graphic cues, no assistance) for each instructional component (fast finish, cumulative rehearsal, etc.). This allows the teacher to see what portions of the program are giving the student the most difficulty. This information is available in tabular form.

Graphs and tables are also available to depict average pause time as a function of position of the item in the list (corresponding to the number of items in circular recall pattern) and accuracy as it relates to each position in the list.



363

Pause Time Table (1/10 Seconds)

John Brown 10/31/86

LEVEL	SERIAL POSITION 1	2	3	4	5	OMEGA
A	8	14	5			0.977
A	7	8	4			0.724
A	11	23	4			0.965
B	5	33	4	5		0.985
B	5	15	5	4		0.986
B	4	19	4	4		0.982
C	5	14	33	5	5	0.969
C	13	17	37	5	6	0.956
C	11	19	43	4	5	0.973

Finally, the computer will supply a written interpretation of student performance. This information will summarize the student's performance, clarify its significance, and assist the teacher in relating it to tasks beyond the instructional program.

Video Game Interlude

Interludes of the game follow each assessment portion, including those following instruction. If the student meets criterion for an assessment, s/he will receive a longer interlude than the one routinely provided. The game interlude that follows the very first assessment will be longer to provide ample time to explain the rules of the game.

The setting of the game is intergalactic space; the object is to make contact with stars, with points accumulated for each star contacted, only the blue stars are point-carrying stars; purple stars serve as distractors. An enemy ship is continually competing against the player since it moves across the screen capturing stars in its path.

There are four levels of difficulty. At Level 1, the stars are stationary throughout the interlude. The stars stay illuminated for five seconds, then disappear for five seconds. The enemy ship does not appear during Level 1. In Level 2, the stars are again stationary, but they are illuminated for only one second, then disappear for two seconds. The enemy ship appears during Level 2.

In Level 3, 75% of the stars are stationary, and the remaining 25% move across the screen. All stars remain illuminated for one second, and disappear for two seconds. The enemy ship appears in Level 3. In Level 4, only half of the stars are stationary; the others move across the screen. Illumination times are identical to those in Level 3. The enemy ship appears in Level 4.

If Levels 1-4 are selected for a student, the game will always play at that level until you change the selection. If Level A is selected, the game will begin at Level 1 and then will automatically increase levels as the student advances through the levels of the memory task.

<u>Rules of the Game</u>. The object of the game is to collect points by contacting stars. Stationary stars are worth one point; shooting stars are worth five points. When a star has been contacted, it chirps and then disappears from the screen. Nothing reappears in its place. The enemy ship appears in Level 2 and beyond. It crosses the screen in a straight line near the top, and any stars in the path of the enemy ship are captured by the ship and appear as a trail behind the ship for the remainder of its journey across the screen. The ship itself is worth ten points. The player has an opportunity to release the captured stars and add their point values to his/her total by blasting the enemy ship before it leaves the screen. This can be accomplished by hitting the ship with a rocket fired from one of three rocket bases located at the bottom of the screen. There are an unlimited number of rockets available, although a second rocket can't be fired until the first one has cleared the top of the screen.

The playing time remaining (in minutes) and the points accumulated always appear at the bottom of the screen. The player gets an extension of playing time for successful performance. An extension of playing time provides the player with a new screen and play proceeds as it did at the videogame interlude's commencement.

<u>Light Pen Mode</u>. The player touches stars with the light pen to contact them and activates the rockets by touching the rocket bases.

<u>Joystick Mode</u>. The player directs the cursor to a star or rocket base with the joystick and presses the joystick button for contact activation.

APPENDIX D

Field Test Plan

U.S. DEPARTMENT OF EDUCATION

OFFICE OF SPECIAL EDUCATION PROGRAMS

CONTRACT NO. 300-84-0156

TECHNOLOGY TO ENHANCE SPECIAL EDUCATION: REMEDIATION OF PROBLEMS IN LOGICAL THINKING AND MEMORY

FIELD TEST PLAN

PROJECT STAFF:

AL CAVALIER, PH.D.

BETH MINEO, PH.D.

SUE EADES

THE BIOENGINEERING PROGRAM

DEPARTMENT OF RESEARCH AND PROGRAM SERVICES

ASSOCIATION FOR RETARDED CITIZENS OF THE UNITED STATES

TABLE OF CONTENTS

Main Research Questions

a) Does the computer-based package yield data similar to those derived from previous laboratory and classroom research on memory and metamemory processes?
b) Is the assessment of process deficiencies valid?
c) What specific memory problems in the targeted populations are uncovered by this package?
d) Are the instructional techniques on effective cognitive processes effective?
e) To what degree are the diagnosed short-term and long-term memory problems remdiated by the computer-based package?
f) Will students generalize the use of the trained strategies to instances on which they have received no training?

Rationale

Educational and cognitive research has shown that memory process deficiencies are pervasive in persons with mental retardation or learning disabilities. Cognitive assessment and remediation procedures that have been too cumbersome and time consuming for classroom use have been translated by the ARC/US into a computer-based instructional package. This research permits examination of the package's validity and instructional potency, as well as a comparison of the remediation effects among groups of students with varying degrees of learning handicaps. It also provides a means by which valuable feedback from students and classroom teachers can be obtained.

Student Populations

Approximately 60 students will participate (20 nonhandicapped students, 20 students with mental retardation, and 20 students with learning disabilities). The nonhandicapped students are those functioning adequately in regular education classrooms. Learning disabled students and students with mental retardation have been identified as such in accordance with school district evaluation and placement procedures. Learning disabled students have average or above average intellectual functioning (as measured by the WISC-R) but demonstrate significant delays in one or more academic areas. Students with mental retardation demonstrate significantly delayed functioning in academic areas commensurate with their overall academic functioning (as measured by the WISC-R) and adaptive behavior. All students will be between 12 and 14 years of age.

Measurement Instruments

Four measurement instruments will be used. The first is a separate computer-based assessment of memory competencies that will be used as the pre- and post-test for all research subjects. The second is the assessment and remediation package incorporated into the system software. The system allows for the automatic recording and analysis of data. The third instrument is a questionnaire for participating teachers that will be used in conjunction with the direct student research. The fourth instrument is a questionnaire for participating students. Copies of the questionnaires may be found in Appendix A and Appendix B.

General Procedures

a) Six complete computer systems with all necessary peripherals will be placed in the evaluation sites for the duration of the field tests.

b) Two sessions will be conducted for a sample of the teachers whose students have served as subjects. The first will be a short inservice program intended to acquaint them with the theoretical basis for the software and the way in which the assessment and instruction is being presented to the students. They will have an opportunity to interact with the instructional system during this session. The second session will be a follow-up to the completion of data collection. In this session the preliminary research findings will be discussed, and teachers will be asked to evaluate the software's appropriateness in relation to the particular type of student with whom they are involved.

c) Students will complete a questionnaire regarding their previous experience with computers and their opinions regarding the most effective and enjoyable uses of computers in the schools. Following completion of participation in the study, a brief interview will be conducted with each student in which they will be asked to comment on the software's ability to teach a new skill, hold student attention, and motivate improved performance.

d) A pre-and post-assessment of memory processes in all participants will take place at the beginning and end of the study. This assessment will be computer-based, so all participants will undergo a brief familiarization with the computer before assessment begins. Each assessment should take approximately one hour.

e) Participants receiving the intervention will interact daily with the computer until they have completed the instruction or attained their maximum level of performance. The daily sessions will last for approximately 30 minutes unless the student chooses to terminate the session early. Students will interact independently with the computer, and a trained research assistant will be available for assistance should it be needed.

f) Data on the educational and psychometric measures pertaining to the students' diagnostic classifications were obtained from school records.

g) Parental consent letters detailing the study were sent to all parents of potential subjects. A copy of this letter may be found in Appendix C.

Research Design

<u>Procedure: Pre- and Post-Test</u>. All subjects will receive training to familiarize them with the computer, after which a computer-based pre-test will be administered. The subjects will receive 6 trials for each of 9 different circular recall requirements (i.e., the order in which recall is to occur) in which they will be shown items in a list and be asked to recall the list. The last three of these trials will be used in the data compilation. During this pre-test, no subjects will receive memory-strategy training. This procedure will be the same for the post-test that follows the training sessions.

<u>Intervention</u>. Half of the subjects in each subject classification will receive the computer-assisted instruction in the use of effective memory strategies. The other half will receive no intervention between pre-and post-tests. A total of 5 circular recall requirements will be addressed in training.

<u>Experimental Design and Data Analysis</u>. There will be several factors, or independent variables, addressed in the experimental design and data analysis: subject classification (learning disabilities, mental retardation, non-handicapped), instructional level (intervention, control), test (pre, post), serial position in a list, and circular recall requirement. The dependent measures are a measure of memory strategy use (as relected by the omega statistic) and an accuracy measure (as reflected by percent correct recall).

Three types of group analyses will be conducted: The first is to determine the relationship between the measures of memory processes and recall accuracy. Theoretically, if a subject revises his processing to match that taught in the instructional package, his recall accuracy should improve. This correlation will be computed for every circular recall requirement.

The second type is an aggregate analysis of variance of the pre- and post-test measures of recall accuracy and memory processing as a function of subject classification and instructional level. Again, these will be computed for every circular recall requirement.

It may happen that a subject's recall accuracy would be satisfactory without his/her use of the memory strategies that were instructed. To determine this, a third type of analysis that looks at serial position in relation to recall accuracy and memory processing would be necessary. This will entail a four-way analysis of variance for subject classification, pre-/post-test, instructional level, and serial position (the specific number of serial positions is dependent on circular recall requirement). This analysis of variance will be computed for every circular recall requirement.

Depending on the nature of the accumulated data, it may be appropriate to do some analyses of individual subject's data in terms of the relationship between specific circular recall requirements and memory strategy use and/or recall accuracy data.

Since instruction will be conducted on only half of the circular recall requirements assessed in the pre- and post-tests, performance on the remaining circular recall requirements will serve as an index of generalization of strategy use to different but similar tasks. It would be premature at this juncture to assess generalization to dissimilar ones. If generalization is evident, future research efforts should explore the extent to which it occurs and the conditions that optimize its occurrence.

Personnel

The research will be conducted by the Assistant Project Director and three research assistants in conjunction with the subjects' special education teachers. They will conduct daily sessions in classrooms of various schools in the Dallas Independent School District. The teachers at the various sites will be invited to return comments on the forms provided.

Project Summary

The following represents a summary of the project as it was communicated to representatives of the Dallas Independent School District; on the basis of this summary and a research proposal, permission was granted by the district for the research to be conducted in classrooms of the district's middle schools:

The most common informal observation about children and youth with learning disabilities and mental retardation is that they do not "learn" as quickly or thoroughly as their non-handicapped peers. Over the past 15 years of research, these observations have been well substantiated. However, a large volume of investigations indicates that these learning problems are primarily caused not by deficiencies in learning ability per se, but by deficiencies in the person's memory which underlies learning (Belmont & Butterfield, 1969; Detterman, 1979; Ellis, 1970). Guided by a well-supported theory of memory processes, several researchers have succeeded in their attempts to improve the memory processes of persons with learning difficulties.

This project incorporates one of the best and most frequently used memory-assessment tasks along with training on the most effective memory strategy for that task into a computer-based instructional system for assessing and assisting in remediating basic memory-process deficiencies. The computer-based system increases the potential for learning since it incorporates many of the features found to enhance retention by leading cognitive psychologists and special educators.

Significance of the Problem

Initially, the poor memory of persons with learning problems was attributed to immutable defects in their neurological system (Ellis, 1963). As research techniques and theories become more refined, however, the precision in the understanding of memory deficiencies steadily increased. The most important influence in this movement was the development of sophisticated theories of memory based on computer information-processing models of mental functioning in non-handicapped persons (Atkinson & Shiffrin, 1968; Waugh & Norman, 1965). When translated from the field of theoretical cognitive psychology to the field of mental retardation (Ellis, 1970), the theories prescribed a whole new way of conceptualizing the mental activity of persons with mental retardation and pointed the way to a number of possible causes for their memory deficiencies.

In the new conceptualization, memory is held to be comprised of two components, short-term memory (STM) and long-term memory (LTM). Short-term memory is limited in capacity and relatively brief in duration, i.e., approximately 30 seconds. Success in dialing a telephone number that a person has just looked up in a telephone book, but failure to recall it 30 minutes later is an example of the use of STM. Long-term memory, on the other hand, is considered to be of unlimited capacity and of permanent duration (Waugh & Norman, 1965). Recalling the name of a favorite dog from childhood is an example of LTM.

The important tast of transferring needed information from STM to LTM is primarily a function of active mental processing of that information. There are a number of voluntary rehearsal or encoding strategies that an efficient learner can employ to store the necessary information in LTM for later retrieval and use (Atkinson & Shiffrin, 1968, 1971). The more a person uses a cognitive strategy, the less mental effort it requires and the more automatic it becomes (Shiffrin & Schneider, 1977; Sternberg & Wagner, 1982).

With the consensus that memory process deficiencies represented a critical problem for persons with learning difficulties, interest became very intense in determining the degree to which they could be remediated. Extensive research attention turned towards developing an array of effective instructional techniques to impart to deficient information processors the rehearsal and metacognitive strategies of efficient information processors. The basic assumption underlying this research, and the work of this project, was that if basic process deficiencies exist and remain uncorrected, they compound higher-level areas of functioning and frustrate instructional efforts. As a result of this new research, increasingly sophisticated techniques to identify the specific process deficiencies and then to remediate these deficiencies have been emerging (Belmont & Butterfield, 1977; Bray, 1979; Brown, 1978; Campione & Brown, 1977; Glidden, 1979; Hagen & Stanovich, 1977; Kramer & Engle, 1981). However, these techniques are very labor-intensive and have not made their way into classroom applications to any large degree.

The conclusions that are drawn from the information presented above are that: (a) significant and pervasive problem in memory exists in the lives of persons with learning disabilities or mental retardationa, (b) these problems are the result of deficiencies in basic memory and metacognitive processes, (c) assessment techniques are available to identify the specific process deficiencies, and (d) instructional techiques are available to begin to remediate those deficiencies.

The Computer-Based Instructional System

The system developed by the ARC is not curriculum-specific but instead focuses on some of the fundamental cognitive skills which underlie learning and performance across every content area. The software is structured around a memory task which is frequently used in assessment and instructional applications and which requires many of the same cognitive strategies for successful performance that underlie efficient information processing across a wide variety of situations (Latham, 1978).

The instructional system being developed in this project is based upon the ordered recall task. In the ordered recall task, a student is requested to recall in the order presented a list of items that s/he has seen only once. The items are serially-presented, with only one item exposed at one time. The student is asked to first recall the subset of the last items presented (the terminal items) and then circle back and recall the subset of the items which were presented first (the initial items). This aspect of the task is called "circular recall" (Butterfield, Siladi, & Belmont, 1980). In the ordered recall task, each item is displayed for a fixed period of time (e.g., 0.5 seconds), but the student controls the pace of the presentation, i.e., the timing of the presentation of the next item. Task difficulty and memory load requirements can be varied by changing the number of items in the to-be-recalled list and the type of items to be recalled (e.g., letters, numbers, words).

Extensive research has shown that performance on the terminal items reflects a student's STM abilities, and the task permits precise manipulation of variables which pertain specifically to STM limitations and instructional strategies. Similarly, performance on the initial items reflects a student's LTM abilities and is sensitive to a number of manipulations directly related to strategic cognitive activity (Belmont & Butterfield, 1969, 1971a, 1971b; Brown & Barclay, 1976).

One of the most revealing measures of strategic cognitive activity in this task is the length of time the student pauses after the presentation of each item in the list. Non-retarded, non-learning-disabled students generally exhibit high recall accuracy when their pauses steadily increase across the initial items, followed by very brief pausing over the terminal items.

The computer-based instructional system under development combines proven training techniques with the unique capabilities of the microcomputer. This computer-based system breaks the circular recall memory strategy into its component parts, trains each separately, and then chains the components together for smooth operation. The students will work with a number of different list lengths and circular recall requirements; these variations on the same general task should increase the students' understanding of the basic strategy because the students are able to witness and participate in its application in a number of situations. The system also provides the student with additional practice in those areas in which s/he is experiencing difficulty.

Appendix A

Memory-Process Questionnaire for Teachers: Part One

The Association for Retarded Citizens of the United States appreciates your involvement in the field-testing of this software package on cognitive memory processes. Our goal is to develop an effective assessment and training tool that will be of assistance to teachers. Your comments are vital in helping us attain this goal; our refinements to this package will be guided by your feedback. Thank you for your assistance.

Inservice Program

Was a presentation of this type of inservice program useful? Please comment in the space below on how beneficial it was to your understanding and use of the software package.

Please rate the following aspects of the inservice program on how beneficial they were to your understanding and use of the software package.

	Not Beneficial	Somewhat Beneficial	Beneficial	Very Beneficial
Background Information	______	______	______	______
Demonstration	______	______	______	______
Hands-On Experience	______	______	______	______

Memory-Process Questionnaire for Teachers: Part Two

Now that your students have interacted with the software, we would like you to consider a few more questions. Thank you for your assistance in the field-testing of this software package.

Please complete the following:

My students have ___ mental retardation,

___ learning disabilities,

___ no diagnosed learning problems,

and range in age from ________ to ________.

Software Package

Does this software package address important skills/needs?

Does it address skills relevant to your classroom activities?

Is the software design sufficiently interesting to hold students' attention?

Describe the nature of the students for whom you feel this software would be most appropriate.

Please rate the following aspects of the software on the rating scale provided.

	Inadequate 1	2	3	4	5	Excellent 6
Familiarization exercises	___	___	___	___	___	___
Progression of task difficulty	___	___	___	___	___	___
Pace of interaction	___	___	___	___	___	___
Graphics quality	___	___	___	___	___	___
Voice quality	___	___	___	___	___	___
Motivational quality of video game	___	___	___	___	___	___
Ease of record keeping	___	___	___	___	___	___
Use of Voice for Instruction/ Prompting	___	___	___	___	___	___
Usefulness of Data Presentation	___	___	___	___	___	___

Does this software package provide you with useful information? Please comment.

Does the information provided add to your knowledge about your students' memory skills? Please comment.

Was the level of instruction appropriate for your students? Please comment.

Would you purchase and/or recommend the purchase of software such as this for use with special-needs students in your school?

Please describe any suggested changes or recommendations to increase the utility of this software package?

Please feel free to provide any additional comments.

Appendix B

Name	Date

1. Do you have a computer at home?

   If "yes", which brand?

2. Have you used any computers that your school owns?

3. If "yes", on how many days have you used them (approximately) this school year?

4. Would you like to use the computers more often at school?

5. In what ways, if any, can a computer help you learn?

6. In what ways, if any, can a teacher help you learn something better than a computer can?

Appendix C

November 17, 1986

Dear Parent/Guardian:

Your child is eligible for participation in a research project, involving the field testing of new educational computer software. This software was developed through the cooperative efforts of special educators, cognitive psychologists and computer experts, and is intended to assess specific memory difficulties and assist in their remediation.

The software is designed to test the child on a number of related memory tasks, determine the child's level of functioning, and teach strategies that (s)he can use to be more successful. The children's performance on related tasks will be evaluated before and after their experience with this software to see if they are using the strategies they learned from the computer in other activities.

There is virtually no discomfort or risk involved with this research. Since training that involves repetition can get boring, this software includes lots of things that will make learning fun, such as lots of colorful pictures and action on the screen. The computer will be able to talk to your child, and will also build in brief periods of video game playing. This should not only prevent the potential problem with boredom, but should actually motivate the children to perform better.

Eligible children will be divided into two groups. The "control" group will not receive the computer-based instruction, but will be involved in pre- and post-testing. The other group of children who participate can reasonably be expected to benefit from the program in several ways. First, they should experience success on those memory tasks trained directly, and they may also be able to use the strategies taught to them in other situations. In addition, their interactions with the computer and the training assistant should be satisfying and enjoyable.

This research is being conducted by the Association for Retarded Citizens of the United States (ARC/US) for the U.S. Department of Education. It is being conducted with the approval and cooperation of the Dallas Independent School District (DISD) and with the sanction of the Department of Research, Evaluation and Audit and the Department of Special Education. We encourage your interest and participation, and will be glad to answer any questions you might have. Questions may be directed to Beth Mineo of ARC/US at (817) 640-0204, or to Don Hawkins of DISD at (214) 490-8701.

Dallas Independent
School District

Linus Wright
General Superintendent

3700 Ross Avenue
Dallas, Texas 75204

We are asking several non-handicapped students from Fred F. Florence School to participate in a "community/school service" capacity. Twenty students will be selected from those whose parent/guardian agree. Participation will require 30 minutes before school, starting at either 7:30 or 8:00, for 10 to 15 school days. If you agree to allow your child to possibly take part in this project, please sign the attached Parental Consent Form and have your child return it to his/her first period teacher immediately. If your child is selected for participation, you will be notified regarding the time your child needs to be at school and the duration of his or her participation.

Participation in this research is voluntary, and if you choose not to have your child participate there will be no penalty or loss of privileges for your child. You may also discontinue your child's participation at any time.

Results of this project will be shared with other parents and teachers through presentations and publications in appropriate journals; however, no written or oral accounts of this research will mention your child or any other child by name.

Project staff members will be most happy to share the research results with you. We thank you in advance for your consideration of this matter, and we look forward to including your child in this project.

Sincerely,

George W. Reid

George Reid
Assistant Superintendent
Secondary Instruction

Arturo Luis Gutierrez
Assistant Superintendent
Instructional Support

Enclosure

Dear Parent/Guardian:

Your child is eligible for participation in a research project involving the field testing of new educational computer software. This software was developed through the cooperative efforts of special educators, cognitive psychologists and computer experts, and is intended to assess specific memory difficulties and assist in their remediation.

The software is designed to test the child on a number of related memory tasks, determine the child's level of functioning, and teach strategies that (s)he can use to be more successful. The children's performance on related tasks will be evaluated before and after their experience with this software to see if they are using the strategies they learned from the computer in other activities.

There is virtually no discomfort or risk involved with this research. Since training that involves repetition can get boring, this software includes lots of things that will make learning fun, such as lots of colorful pictures and action on the screen. The computer will be able to talk to your child, and will also build in brief periods of video game playing. This should not only prevent the potential problem with boredom, but should actually motivate the children to perform better.

Eligible children will be divided into two groups. The "control" group will not receive the computer-based instruction, but will be involved in pre- and post-testing. The other group of children who participate can reasonably be expected to benefit from the program in several ways. First, they should experience success on those memory tasks trained directly, and they may also be able to use the strategies taught to them in other situations. In addition, their interactions with the computer and the training assistant should be satisfying and enjoyable.

This research is being conducted by the Association for Retarded Citizens of the United States (ARC/US) for the U. S. Department of Education. It is being conducted with the approval and cooperation of the Dallas Independent School District (DISD) and with the sanction of the Department of Research, Evaluation and Audit and the Department of Special Education. We encourage your interest and participation, and will be glad to answer any questions you might have. Questions may be directed to Beth Mineo of ARC/US at (817) 640-0204, or to Don Hawkins of DISD at (214) 526-0991.

If you agree to allow your child to take part in this project, please sign the attached Parental Consent form and have your child return it to his/her homeroom teacher by January 24, 1986.

Participation in this research is voluntary, and if you choose not to have your child participate there will be no penalty or loss of privileges for your child. You may also discontinue your child's participation at any time.

Results of this project will be shared with other parents and teachers through presentations and publications in appropriate journals; however, no written or oral accounts of this research will mention your child or any other child by name.

Project staff members will be most happy to share the research results with you. We thank you in advance for your consideration of this matter, and we look forward to including your child in this project.

Sincerely,

Ruth Turner

Ruth Turner, Ed.D.
Administrator, Special Education

Approved:

George W. Reid

George Reid
Assistant Superintendent
Secondary Instruction

Arturo Luis Gutierrez
Assistant Superintendent
Instructional Support

Enclosure

PARENTAL CONSENT FORM

My child, ______________________________, has my permission to participate in the computer-based instructional project to be conducted by ARC/US and DISD during the second semester of the 1985-86 school year.

______________________________ ______________________

Parent or Guardian Signature Date

Please have your child to return this signed form to his or her homeroom teacher by January 24, 1986. Thank you.

APPENDIX E

Final Marketing Plan

Overview

The Bioengineering staff of the ARC of the United States have developed a computer-based instructional package that can be used to assess and assist in remediating problems in memory and metacognition in children and youth with mental retardation and other developmental disabilities. This innovative package, using a memory task used previously only in laboratory applications, focuses on fundamental cognitive skills that are crucial to learning and performance.

Applications

Although the user population for the cognitive software is extensive, primary usage is anticipated in school systems, specifically special education classrooms. The software will enable a teacher in a special education classroom to assess whether a student has significant memory process deficiencies, identify the nature of the deficiencies, and provide the student with individualized instruction that will help him/her improve memory skills.

Ultimately, the software will provide teacher and, possibly, parents, a means by which they can begin to remediate serious and pervasive learning problems encountered so often in children and youth with mental retardation and learning disabilities.

A sample of potential publishers also see applications for the package as a research tool in university settings that are involved in cognitive research.

Finally, the software is designed for use on either Apple IIe or IBM-compatible, MS-DOS computers. Since these are the most widely used types of computers in public school systems, this can be considered an additional benefit and selling point.

The Target Population

The software was field tested with middle school-aged students who were nonhandicapped, those with learning disabilities, and those with mental retardation because project staff and consultants believed it to have applicability to the majority of students in the schools. Field-test results support this contention; gains were made by all subject groups that received training. Project staff and a sample of potential publishers agree that modifications to the original package that tailor it for specifically use by particular subgroups of consumers would greatly increase the size of the potential user population as well as increase the pedagogical power of the instructional package.

Channels of Distribution

Having completed the development, field testing, and refinement stages, the instructional package of software and documentation is ready to fulfill its primary purpose, which is assisting teachers in the assessment and remediation of memory deficiencies. Vital to the attainment of this end is the identification of appropriate channels of distribution for the product.

The most preferred distributor would be a software publisher with an established reputation and a wide distribution network in the education market. Since there are literally hundreds of software companies, the pool of relevant potential marketers would be comprised of those offering a product line consistent with the offering of the ARC project. Three types of product lines potentially offer this compatibility: those with regular educational software, those with software designed for special needs populations, and those with software designed specifically around cognitive tasks. These three product lines are not mutually exclusive; in fact, from our compilation of information on software publishers, there are a few companies promoting software appropriate to all three categories. Several companies were contacted regarding their interest in marketing the software. The results of these activities is discussed later in this report.

The responsibility of the ARC in the distribution process is six-fold. The first responsiblity is the technical specification of the capabilities of the instructional package. The second is the identification of the target population. The third is to establish and document the need for software of this type among the target population. The fourth responsiblity is to highlight the unique features of the system. The fifth is to identify, based on the target population and needs assesements, marketing strategies that a software publisher could employ to increase networks of information dissemination. The final responsibility entails dissemination of information regarding the package's capabilities and its availability from the eventual marketer.

Through the field testing process, the ARC obtained the information necessary to meet the first of these outlined responsiblities. The tests involved 60 students: 20 had mental retardation, 20 were learning disabled, and 20 were nonhandicapped. The instructional package was evaluated in terms of its validity (that is, the consistency of its results with those of the laboratory tasks upon which it is based and its ability to differentiate among subject groups) and its instructional value (that is, its ability to assist in the remediation process). The research indicated that the subject groups did bring different abilities to the task and the software detected these differences, and it also demonstrated that the software was effective in improving performance on the targeted memory task across all subject groups.

The research also permitted a determination to be made regarding the breadth of effective application of the package across the populations in need. Although the appropriateness of this software for the population of nonhandicapped students was not the primary focus of the development project, nonhandicapped students were included in the subject population to provide a base of comparison for the handicapped subjects. Field test results indicate that the software was very effective in improving the skills of the nonhandicapped subjects. Further, these students found the software to be motivating and enjoyable. Thus, the original projections of market size have been greatly increased.

Project staff initially conducted an extensive review of the cognitive psychology and special education literatures to identify the characteristics and nature of the populations to benefit from the aid. It was estimated that 55% of the students enrolled in special education classrooms in this country could benefit from this instructional package. This figure represents a market of close to two and one-half million students. Add to this the several million students in regular education classrooms, and the potential market increases tremendously. In addition, further modifications to the software package would make it potentially useful for other purposes and with other populations.

The ARC's extensive literature review also assisted in confirming the need for this type of software. One of the conclusions that can be drawn from this review is that the memory task around which the package is constructed is a valid and "pure" means by which to assess and train fundamental memory skills. Another conclusion is that the computer is a near-perfect vehicle for this package because of its ability for logical analysis and its capacity to deal with large amounts of information in an interesting, effective, and efficient manner. This package accomplishes the marriage of a sophisticated theoretical framework and empirical knowledge base to a practical, educationally-sound assessment and training package. The educational software marketplace currently fails to offer products reflecting large-scale efforts of this nature even though the desirability of such a package has been acknowledged by researchers, service delivery personnel, and software publishers.

The ARC has the responsibility for highlighting the capabilities and positive attributes of the package to potential marketers and eventually to consumers. Detailed description of these features would be lengthy; only the major points will therefore be summarized here as follows. First, the package offers both assessment and remedial components, and remediation is based logically on the assessment results. Second, this assesment permits the remediation components to be individually tailored to each student's needs. Third, the package uses the unique features of the computer to their fullest extent in assisting the student to understand and perform the required tasks. Fourth, the package employs innovations such as digitized speech output and light pen input to enhance its educational validity and appeal. Finally, the child's performance is analyzed and interpreted by the computer, which allows the teacher to obtain practical information for classroom purposes. This analysis and interpretation was designed with assistance from the leading cognitive psychologists in the country. The student's performance data are also permanently recorded for later review by the teacher.

In meeting the fifth responsibility, the ARC will suggest marketing strategies for use by a publisher based upon the factors addressed above. We will assist the publisher in highlighting this program's appeal and value to parents, teachers, and school districts. The ARC's final responsiblity to the distributor will be met through the ARC's ability to disseminate information across a nationwide network. Through our network of 1300 state and local chapters, our core of over 160,000 members, our national publications including our national newspaper which is distributed six times a year to every member, our computerized technology data base, our national electronic mail and bulletin board system, and our Bioengineering Program, we are in unique position to raise the awareness of school personnel on the availability of effective educational software in the marketplace.

Market Demographics

The market for the software is not limited to school systems, but since this is the most likely consumer of this product, it would be pertinent to look at some statistics that will give an indiction of the potential size of the market:

- In 1983, the special education field spent $10 billion on materials used by or on behalf of students.
- Again, as of 1983, 330,000 microcomputers were in this country's schools.
- In the 1984-85 school year, approximately 15 million students and 500,000 teachers used computers in the public schools.
- By the end of 1986, there was an estimated 1,025,000 microcomputers in public schools. Approximately one-quarter of these, or 225,000, were used in special education, benefitting 4.3 million special education students.

- Special education accounts for 11% of the purchases in the educational computing marketplace.

- Market analysts' projections of the amount to be spent on educational software in the 1987-88 school year range from $250 million to $500 million.

These data indicate that there is clearly a significant market for this instructional package both now and in the future. Indeed, since this package appears to have application to both regular and special education programs, the potential market is enormous. The statistics cited above also indicate that substantial dollars are available to purchase equipment and other supplies that will enhance the learning of children and youth in classrooms across the country. Couple these factors with the viability of the cognitive software package and it would appear that the elements are in place to make it a successful product: a legitimate attractive product, an educational void to be filled, recognition of the computer as a viable educational aid, and the availability of monies for purchase.

1. Vest, C.R. (1983). Marketing and procurement of technology assisted learning systems. Micro Market Examiner, 1(3), 1.

2. Blaschke, C.L. (1982). Microcomputers in special education trends and projections. Journal of Special Education Technology.

3. Carol Daniels, LINC Resources (personal communication based on Quality Education Data survey, LINC survey of states, and Johns Hopkins survey).

4. Electronic learning: The guide to the educational marketplace (1984). Scholastic Inc.

Some initial marketing strategies:

1. In practical terms, highlight the effects that deficient, underlying cognitive strategies can have on educational activities, activities of daily living, and vocational activities.

2. Publicize availability of software through trade journals, newsletters, and teacher and education magazines.

3. Promote product at conventions of education, special education, rehabilitation, and computer technology professionals.

4. Organize demonstration seminars for educators.

5. Identify several schools in which to set up package on a trial basis at no charge to the school in an effort to elicit word-of-mouth publicity and testimonials.

Feedback From Commercial Publishers on Software Design

In compliance with SEP's request for feedback on the marketability, useability, and suitability of the product, the ARC identified several commercial software companies having product lines compatible with the software under development in this project. The company presidents and/or product developers were contacted and their participation was requested. Several companies denied our request, citing most frequently the non-remunerative or time-consuming aspects of the task. Non-disclosure agreements were obtained from three marketers who agreed to participate.

These companies were sent an information packet including a statement of the problem addressed in the project, production and marketing plans, and the Program Narrative with accompanying documentation. The company representatives were guided in their review by the survey form created by the ARC project staff. The form was intended to direct the reviewers' comments to the specific aspects of suitability, useability, and marketability.

The evaluations we received were overwhelmingly positive. On a scale from 1 to 6 with 6 being the most favorable score, the software package received an average score of 5.33 in regard to its suitability, an average score of 5.67 in regard to its useability, and a unanimous rating of 6 in regard to its marketability.

Several suggestions were made regarding the eventual marketing of the package. One reviewer commented that while our development work was being done for the Apple and Commodore computers, we might want to eventually consider adapting a version for MS-DOS machines. In the time that has passed between our survey and the present, we received approval to modify our workplan to replace the Commodore version with an MS-DOS version in response to the current trend in the educational marketplace. One reviewer commented that we might suggest to the eventual marketer that a lightpen be included with the instructional package to enhance convenience and discourage piracy. Reviewers also suggested that the market for this software could be increased greatly if field tests demonstrated it to be effective in improving the cognitive skills of the non-handicapped student population. The prevailing opinion appeared to be that the instructional package was a good one, and that effective marketing was the key to its viability as a commerical product.

Locating a Commercial Marketer

Project staff worked in conjunction with the staff at LINC Resources to locate potential marketers for this product. LINC has in place a procedure by which software developers can locate appropriate potential publishers for their products. LINC provided assistance in the identification of appropriate publishers, in the preparation of informational documents to be sent to the publishers, and in the conduct of negotiations between the developer and marketer.

In the last months of the project, approximately 20 software companies were contacted regarding their interest in marketing the software package. Interested parties were asked to demonstrate their willingness to collaborate by submitting to ARC project staff a statement of the company's capabilities in regard to advertising and production as well as its distribution channels. Project staff reviewed responses to this request and selected five companies with which to conduct more detailed discussions. The three companies with the most appropriate capabilities, product lines, and demonstrated interest were invited to a software demonstration and marketing discussion held at the ARC National Headquarters.

Project staff met with representative of Laureate Learning Systems and American Guidance Services. Laureate offers a product line geared toward remediating communicative and learning difficulties. The large AGS market is primarily oriented toward regular education. Representatives of both publishers saw a comprehensive demonstration of the software's capabilities, discussed field test results, and presented strategies for further refinements to the software.

Both publishers agreed that this software package has wide applicability that can be enhanced further by the addition of some features and the definition of specialized markets. For instance, one publisher suggested enlarging the selection of available stimuli to include pictures, which would render the software appropriate for younger children. Manipulation of the minimum and maximum memory requirements would also increase the potential user population. Making all of these options selectable would allow teachers to customize the presentation to meet the needs of individual students to an even greater extent than the software presently allows.

Laureate is primarily interested in the special education market, while AGS would focus on the regular education market. This circumstance permits negotiation with both parties since their markets are for the most part exclusive of one another, and our research indicates the wisdom in defining separate versions of the software for each market. AGS is also interested in developing some workbook-based activities as an adjunct to computer instruction.

Project staff recognize another market for this software development. A package that would permit selectability for features such as list length, viewing time, criterion levels, and stimulus items would be a valuable tool for scientists conducting research on cognitive processing. This type of research is currently ongoing but is hampered by cumbersome presentation techniques and tedious data collection procedures. The software, as it is currently configured, automically presents stimulus trials, records responses, and analyzes data. Only minor modifications would be necessary to make this a very useful research tool. Project staff have identified a publisher targeting this narrow market, and the company is enthusiastic about adding this product to its existing offerings.

The companies, owing to differences in size and policy, are obligated to pursue collaborations with the ARC via different routes. For instance, AGS needs to discuss the opportunity at an annual review meeting. In contrast, Laureate immediately stated an intense interest and after negotiation with the ARC submitted a proposal to the Small Business Innovation Research Grant program to support the necessary refinement work. The ARC has secured permission from the Department of Education to hold the copyright on the software, which helps to insure that the integrity of the software will be maintained on the road toward commercial publication.

The ARC will maintain its commitment to bring this product to market by guiding commercial publishers in their attempts to refine the package for use by their particular market population. Project staff have identified pedagogical and cosmetic changes that would improve the package, and these will be shared with the eventual publishers. It is not unreasonable to believe that this software might eventually be commercially available in three different forms targeted to meet the needs of a variety of populations. The ARC will continue to work with LINC and potential publishers to make this possibility a reality. When this occurs, it will indicate that the marketplace is accepting a revolutionary new type of software and that a strong theoretical base is a viable positon from which to initiate a software development project.

APPENDIX F

Field Test Report

Field Test Report

This report documents the field testing of the software developed by the ARC under Contract No. 300-84-0156. The educational and cognitive research on which this project was based has shown that memory process deficiencies are pervasive in persons with mental retardation and learning disabilities. The ARC translated proven yet cumbersome cognitive assessment and remediation procedures into an instructional package that employs the computer as the tutor, the interactional interface, and the data collection and analysis system. The field testing permitted an examination of the package's validity and instructional potency as well as a comparison of the remediation effects among groups of students with varying degrees of learning handicaps. The report contains a description of the research plan, presentation of the data-based findings, and a discussion of the implications of these findings as well as the anecdotal observations made during the course of the seven-month field test period.

Student Population

Sixty students participated in the complete study. Several other students were dropped from subject rolls because of scheduling and relocation problems. Of the 60 students participating in all aspects of the study, 20 had mental retardation, 20 had learning disabilities, and 20 were non-handicapped. The nonhandicapped students were those functioning adequately in regular education classrooms. Students with learning disabilities and students with mental retardation were identified as such in accordance with school district evaluation and placement procedures. Learning disabled students had average or above average intellectual functioning (as measured by the WISC-R) but demonstrated significant delays in one or more academic areas. Students with mental retardation demonstrated significantly delayed functioning in academic areas commensurate with their overall academic functioning (as measured by the WISC-R) and adaptive behavior. All students were between 12 and 14 years of age, attended schools within the Dallas Independent School District, and furnished written parental permission for participation in the study.

Measurement Instruments

Four measurement instruments were used. The first is a computer-based assessment of memory competencies that was used to generate pre- and post-test data regarding students' ability to remember items presented in sequence. The second is the assessment and remedial instruction incorporated into the software. The software allows for the automatic recording and analysis of data. The third and fourth instruments are structured interviews of student familiarity with computers and of opinions regarding the instructional software package.

General Procedures

The general data collection procedures were as follows:

a) Data on the educational and psychometric measures pertaining to the students' diagnostic classifications were obtained from school records, and informed consent letters were obtained from the parents of all potential subjects.

b) Five complete computer systems with all necessary peripherals were placed in the schools. Fifteen public schools participated, although 2/3 of the subjects attended a single school. Thus, the majority of the computers remained in that particular school and the others travelled with the itinerant research assistant.

c) Students were interviewed in their classrooms regarding their previous experience with computers and their opinions on the most effective and enjoyable uses of computers in the schools.

d) A pre-assessment (pre-test) of memory processes was conducted with all students. This assessment was computer-based, and all participants underwent a brief familiarization with the computer before the assessment began. Each assessment required between 60 and 90 minutes.

e) Participants receiving the training intervention interacted daily with the computer until they had completed the instruction or attained their maximum level of performance. Daily sessions lasted 30-45 minutes unless the student chose to terminate the session early. Students interacted independently with the computer, and a trained research assistant was available to provide assistance and record data for use in reliability checks.

f) Following completion of a post-assessment (post-test), a brief interview was conducted with each student in which they were asked to comment on the software's ability to teach a new skill, hold student attention, and motivate improved performance.

Research Design

Procedure: Pre- and Post-Test. All subjects received training to familiarize them with the computer, after which a computer-based pre-test was administered. The subjects received six trials at each of eight levels (hierarchically arranged in regard to difficulty of circular recall requirement) in which they were shown items in a list and asked to recall the list. The last three of these six trials at each level were used in the data compilation. During the pre-test, no subjects received memory-strategy training. This procedure was the same for the post-test that followed the training sessions.

Intervention. Half of the subjects in each subject classification received the computer-based instruction in the use of effective memory strategies. The other half received no intervention between pre- and post-tests. A total of six recall requirements were included in the training. These six were a subset of the eight levels used in the pre-and post-tests. Students progressed through training until they had completed the entire sequence or had failed to meet the minimum criterion for continuation of training.

The intervention section consisted of assessment/training cycles that involved assessment of recall accuracy at a level and then training on cognitive strategies appropriate to that level if the student did not pass the assessment. If the student passed the assessment without training, or passed it after receiving training, s/he advanced to assessment and possibly training on the next level of difficulty, and so on until s/he failed to pass the assessment and failed to benefit from training.

Experimental Design and Data Analyses

There were several factors, or independent variables, addressed in the experimental design and data analyses: subject Classification (learning disabilities (LD), mental retardation (MR), non-handicapped (NH)), instructional Condition (training, control), Test (pre-, post-), and circular recall requirement (the different difficulty levels). Classification and Condition were between-subjects factors, Test was a within-subjects factor. The dependent variables were a measure of cognitive strategy use (as reflected by the omega-squared statistic), a recall accuracy measure (as reflected by the number of items correctly recalled), and a memory-improvement measure (as reflected by the increase in the number of difficulty levels passed from pre-test to post-test).

Accuracy - 2/2 Level. A 3-way analysis of variance (ANOVA) on Classification x Condition x Test was conducted on the 2/2 (circular recall) level of difficulty, using the median number of items recalled accurately over the last three assessment trials as the dependent variable. The analysis showed that the main effects of Classification ($p<.001$), Condition ($p<.014$), and Test ($p<.001$) were all significant. These main effects were qualified by a 3-way interaction of Classification x Condition x Test ($p<.057$). This analysis reveals that:

- o the NH group recalled more than the LD group who recalled more than the MR group,

- o the subjects who received training recalled more than the subjects who received no training,

- subjects recalled more on the post-test than they did on the pre-test,

- the MR group who received training increased their recall from pre-test to post-test more than MR group who did not receive training and this difference was greater than the corresponding differences in the NH and LD groups.

Omega-squared - 2/2 Level. A 3-way ANOVA of Classification x Condition x Test was conducted on the 2/2 level of difficulty, using the median omega-squared value over the last three assessment trials as the dependent variable. No significant interaction or main effects were obtained.

Accuracy - 3/2 Level. A 3-way ANOVA of Classification x Condition x Test was conducted on the 3/3 level of difficulty on recall accuracy. Significant main effects for Classification ($p<.001$) and Test ($p<.001$) and a significant 2-way interaction of Condition x Test ($p<.001$) were obtained. These effects were qualified by a significant 3-way interaction of Classification x Condition x Test ($p<.004$). This analysis reveals that:

- the LD and MR groups who received training showed a greater increase in recall from pre-test to post-test than the LD and MR groups who received no training. This was not true for the NH groups.

Omega-squared - 3/2 Level. A 3-way ANOVA of Classification x Condition x Test was conducted on the 3/3 level of difficulty on omega-squared. No significant main effects or interaction effects were obtained.

Levels Passed. As a result of the pass/fail criterion operating at each difficulty level, the numbers of students in each group who participated at the greater difficulty levels decreased with each succeeding level. This provides validity to the original hierarchical ordering of the different circular recall requirements. It also precludes conducting ANOVA's at these greater difficulty levels. To analyze the overall change that the instructional package created in student performance a more appropriate analysis is an analysis of variance on the increase or decrease in the number of difficulty levels (circular recall requirements) passed on the post-test relative to the highest level passed on the pre-test as a function of subject classification and instruction condition.

A 2-way ANOVA of Classification x Condition was conducted using the levels increase/decrease as the dependent variable. A significant main effect for Condition ($p<.003$) was obtained. This analysis revealed that:

- the subjects who received training showed significantly greater increases in the number of difficulty levels passed on the post-test relative to the highest level passed on the pre-test than subjects who received no training.

The dependent measures for each subject in this analysis were derived using the original criterion for scoring a level as passed, that is, perfect performance on the last three assessment trials on tha level. A unanimous conclusion among project staff during the field testing was that this criterion was too rigid for all subject groups and suppressed the true effects that were taking place. We observed that occasionally a student made a simple mistake on one of the last three trials, e.g., momentarily losing his/her train of thought on the sixth trial after concentrating intensely on the first five trials, inadvertently touching the lightpen to the wrong place on the screen thereby selecting an incorrect letter, or turning away from the display screen momentarily because of a classroom distraction. This last example represented a large number of these types of "oops" errors, as each of the 60 subjects was tested in the natural classroom environment that, in most cases, was replete with a wide variety of "distractions". The "perfect-on-the-last-three" criterion forced each one of these instances to be scored as a failure. We believe a more reasonable criterion for passing a level that better reflects the actual gains and losses made is correct performance on four of the six assessment trials. We believe this is not an easier criterion but is a more fair criterion that gives a truer picture of the actual effects.

A 2-way ANOVA of Classification x Condition was conducted using the levels increase/decrease as the dependent variable, scored with the "four-out-of-six" passing criterion. Significant main effects for Classification ($p<.001$) and Condition ($p<.001$) were obtained. This analysis revealed that:

- the subjects who received training showed significantly greater increases in the number of difficulty levels passed on the post-test relative to the highest level passed on the pre-test than the subjects who received no training.

- the NH group showed greater increases than the LD group who showed greater increases than the MR group.

<u>Conclusions</u>. In any task allowing measurement of cognitive activity that will be used to compare cognitively-impaired and non-impaired individuals, it is important to employ difficulty levels that permit unrestricted assessment of the performance of both groups of subjects, i.e., levels that are not too difficult for the cognitively-impaired subjects, nor too easy for the non-impaired subjects. While typical clinical use of this software

in the classroom would most likely focus individually on the assessment and instruction of each student, the software incorporates sufficient flexibility in selecting difficulting levels that, when comparisons across student groups are desirable, difficulty levels that avoid floor and ceiling effects in the analyses should be easily identified.

In the research in this project, the initial level of difficulty, 2/2 circular recall, was too easy for many of the NH subjects. The task at this level did not tax them and they did not need to employ any particular cognitive strategies to perform successfully. As a result, the most appropriate analyses in this research are the ANOVA's on the 3/2 level of difficulty and on the increase/decrease in levels passed from pre- to post-test. From these analyses the following conclusions can be drawn:

- the data derived from this computer-based assessment-and-instruction package are orderly and the general results are consistent with the types of results that have been previously obtained in laboratory research; therefore, the software represents a valid transfer of a sophisticated cognitive tool from laboratory equipment to a standard microcomputer,

- the software provides effective instruction within a level of difficulty for students who are learning disabled and students who are mentally retarded; it also improves the performance of nonhandicapped students,

- after receiving instruction through the software, students who are non-handicapped, learning disabled, and mentally retarded can perform successfully on memory tasks that are more difficult than the tasks which they completed successfully prior to instruction,

- the relationship between the amount of instruction provided and the criterion used to evaluate whether a student benefitted from that instruction needs to be researched for this application, as it appears that some beneficial effects may be obscured by the choice of inappropriate levels.

Student Interviews

Interviews on the following two topic areas were conducted with students who participated in the research: (1) their experience with and beliefs about computers and (2) their views about the instructional package that they helped to evaluate.

The first interview was conducted in the orientation stage of the research. The following questions were asked:

1. Do you have a computer at home?
   If so, which brand?

2. Have you used any computers that your school owns?

3. If so, on how many days have you used them (approximately) this school year?

4. Would you like to use the computers more often at school?

5. In what ways, if any, can a computer help you learn?

6. In what ways, if any, can a teacher help you learn something better than a computer can?

7. On what subjects, if any, would you like to receive computer instruction in addition to teacher instruction?

8. Do you think you remember information better when a teacher instructs you or when a computer instructs you?

9. Do you prefer to learn by yourself or in a group?

10. What features make up a good video arcade game?

11. Name your top 5 video arcade games.

The results provide a view of how the participants view computer usage as it pertains to the learning process. This was studied in questions 5-9, while experiences with computers are probed in questions 1-4. Questions 10 and 11 provide general information concerning the participants' interests in video games on the market, and this information further reveals features that appeal to and motivate the interests of those studied.

Thirty percent of the participants answered that they had computers at home, while the Non-Handicapped Training group made up 36% of those with home computers. The MR groups reported no home computers. However, most students had used computers at school (77%); among the MR groups, 70% had used school computers before. Among those students who had access to school computers, they most frequently spent a total of 1 semester in computer-related learning and activity. When asked if they would like to use computers more often at school, 96% answered Yes. Among the ways in which students thought a computer can help in learning, math was the most popular answer, followed by general studies and learning skills.

The benefits of a teacher over a computer were largely factors of communication: most participants in the LD group answered that teachers could explain things more clearly than a computer could. The NH groups felt that teachers could communicate more coherently in topics that ranged from science laboratory applications to mastering the computer itself. While the MR groups found difficulty in citing ways a teacher could help them learn better than a computer, they did, however, give suggestions in favor of the computer. Math, reading, and language arts were subjects in which the MR groups felt computer use would be helpful. Likewise, both the LD and the NH groups cited general studies as subjects in which computer instruction could supplement teacher instruction. Forty-seven percent of the participants said they could remember information better when a teacher instructs them, 38% favored computer instruction, and 10% favored both. Forty-seven per cent preferred to learn by themselves, while 49% preferred group settings.

The preferences in videogame features and attractions were relatively consistent within groups, however they varied among groups. For instance, the NH groups repeatedly cited color as an important videogame feature, while action and challenge were favorites of the LD group. Shooting was the major feature that attracted the MR group.

Of the video arcade games that most subjects liked the best, the top five were those with animated characters and character personalities. The NH group cited the most high-level, skill-related games, but still maintained agreement with the other groups who favored the "animated character" type games. The overall favorites were: 1. Pac-Man 2. Ms. Pac-Man 3. Donkey Kong 4. Centipede 5. Mario Brothers

The second interview was conducted at the completion of the student's participation. The results reveal the views of the participants on the software developed in this project. The following questions were asked:

1. What do you think this software was trying to teach you?

2. Do you think you gained new skills?

3. If so, what were they?

4. What things did you like best about it?

5. What things did you like least?

6. How would you change the software?

In response to the first question, almost all participants responded with an answer pertaining to memory, or remembering, indicating that most of the participants had a clear picture of what skills were being tested. Over 90% of the participants felt that they had gained new skills as a result of the project and memory skills were cited frequently.

The video game was unquestionably the favorite aspect of the software. The feature mentioned second most frequently was the memory task, confirming the subjective impressions of our research assistants that many subjects truly enjoyed the type of memory tasks that the software presented. The students also cited the challenge presented in both the competition with their peers on the video game and on the memory tasks.

Subjects were more heterogeneous in response to the question on the features they liked the least. Some cited the difficulty of the higher levels of testing. Possibly reflecting greater anxiety, subjects in the LD and MR groups often mentioned that they liked least "making mistakes" or other related answers. The NH group that received training frequently cited "training", confirming our subjective impression that after the initial training they did not require the same frequency and degree of training as the other groups.

When asked how they would change the software, over 53% said they would make no changes at all. Among those who felt a change was needed, those from the NH groups suggested making the memory task easier and reducing the training. Subjects in the LD groups mentioned reducing the levels of difficulty, giving more examples, and giving more playing time on the videogame. No members of the MR group suggested any changes.

Subjective Observations

Observations made over the course of field testing provided valuable information about the software package as well as about the characteristics and needs of each of the subject groups. They are discussed in this section because they have a direct bearing on the outcomes reflected in the data.

As a unit, the subject groups brought different skills and experiences to the task presented via the software. The nonhandicapped group had more experience with computers outside of the school setting and were less fearful of the situation. Of all three groups, the students with mental retardation appeared particularly tentative in their physical interaction with the computer system. Use of a lightpen interface was novel for all students, and the majority mastered its use with ease.

Considering the age of the subjects, the experimenters expected to encounter behavior patterns typically attributed to middle school students. We expected them to be somewhat disdainful of tasks requiring a concerted effort and a serious attitude, and we were concerned that the presence of their peers would exacerabate these reactions. We also were concerned that subjects would be unable to maintain concentration of sufficient quality and duration for successful task completion. On these counts, our concerns were unfounded.

Although students initially approached the testing situation with some trepidation, and while many attempted to mask this with an air of nonchalance, once subjects were familiarized with the system they worked earnestly to complete the tasks successfully. The videogame reward was one that the students worked toward and enjoyed playing. They also took pride in their accomplishments both on the task and on the videogame.

One very concrete indication of the intensity with which they approached the work was their concentration. It was common for students to maintain attention to the display screen for 30 to 45 minutes without distraction. This was an especially surprising observation in regard to the students with learning disabilities and mental retardation. This may be attributable to the students' perception that they needed the information that was being imparted and trained in order to be successful on the assessments, and also to the quickly-paced interactional format in which instruction and practice were couched.

These observations should not be interpreted as suggesting that the instruction had the same attention-maintaining effect across students. In fact, the non-handicapped students who received training became increasingly distractible as they completed more cycles through training. Since this was seen to a lesser degree with the learning disabled students, and was not observed at all with students with retardation, these circumstances lead to the conclusion that detailed instruction during each cycle and at every level was not necessary for the nonhandicapped students. Rather, it may have been sufficient to provide this group with some initial instruction followed by repeated opportunity for practice with different recall requirements. As the instructional program is currently designed, it provides intensive instruction at each and every difficulty level. In our field testing, this circumstance appeared to engender some frustration in those students who understood the task and simply needed practice at generalizing these basic skills to new recall requirements.

Further support for this notion came from the attending behaviors of the students with mental retardation. It was within the group of subjects who received training that attention was most intense. Subjects with mental retardation were in greatest need of the instruction presented, and their extremely high attention levels would indicate that they were aware of their lack of skills necessary for correct performance of the memory task and were actively seeking to improve their performance.

These findings indicate that an appropriate modification to the existing software package might be an option permitting selection of various degrees of instruction. In this way, students requiring all that the original package offered in terms of task breakdown, hierarchical presentation, and fading of cues could benefit from all of these features, while students requiring a less detailed training package would be relieved of excessive detail that might impede motivation or learning.

A related observation is that nonhandicapped and learning disabled students rarely required additional explanation beyond that offered as a part of the software. On the other hand, students with mental retardation frequently required the research assistant to provide additional information as an adjunct to the software's instruction. This suggests that an even more detailed version of the instruction than the version currently offered might be helpful to a subset of students with significant learning problems. The need for additional instruction could be a factor determined by the software on-line as it analyzes accuracy and pause time data and detects consistent error patterns in the data. This additional feature would greatly enhance the power of this package.

APPENDIX G

Preliminary Marketing Plan

U.S. DEPARTMENT OF EDUCATION

OFFICE OF SPECIAL EDUCATION PROGRAMS

CONTRACT NO. 300-84-0156

TECHNOLOGY TO ENHANCE SPECIAL EDUCATION: REMEDIATION OF PROBLEMS IN LOGICAL THINKING AND MEMORY

PRELIMINARY MARKETING PLAN

PROJECT STAFF:

AL CAVALIER, PH.D.

BETH MINEO, PH.D.

SUE EADES

THE BIOENGINEERING PROGRAM

DEPARTMENT OF RESEARCH AND PROGRAM SERVICES

ASSOCIATION FOR RETARDED CITIZENS OF THE UNITED STATES

TABLE OF CONTENTS

OVERVIEW

ARC of the United States Bioengineering scientists have developed a computer-based instructional package that can be used to assess and assist in remediating problems in memory and metacognition in children and youth with mental retardation and other developmental disabilities. This innovative package, using a memory task used previously only in laboratory applications, focuses on basic cognitive skills which are crucial to learning and performance.

THE TARGET POPULATION

This package will be field tested on children and youth who are non-handicapped, those with learning disabilities and those with mental retardation because of its applicability to the entire school-age population. However, the population to benefit the most from this aid is that group with identifiable memory difficulties who are able to interact with the computer system. Generally, the audience that falls into this category is composed of school-aged children and youth with mild to moderate mental retardation and those with learning disabilities.

APPLICATIONS

Although the user population for the cognitive software is extensive, primary usage is anticipated in school systems, specifically special education classrooms. The program will enable a teacher in a special education classroom to assess whether a student has significant memory process deficiencies, identify the nature of the deficiencies and provide the student with individualized instruction that will help him/her improve memory skills.

Ultimately, the system will provide teachers and, possibly, parents, a means by which they can begin to remediate serious and pervasive learning problems encountered so often in children and youth with mental retardation and learning disabilities.

Finally, the system is designed for use on either the Apple II series or the IBM compatible, MS-DOS computers. Since these are the most widely used types of computers in public school systems, this can be considered an additional benefit and selling point.

CHANNELS OF DISTRIBUTION

Once the development, field testing and refinement stages have been completed, the instructional package of software and documentation will be ready to fulfill its primary purpose, which is assisting teachers in the assessment and remediation of memory deficiencies. Vital to the attainment of this end are effective channels for distribution of the product.

The most preferred distributor would be a software publisher with an established reputation and a wide distribution network in the education market. Since there are literally hundreds of software companies, the pool of relevant potential marketers would be comprised of those offering a product line consistent with the offering of the ARC project. Three types of product lines potentially offer this compatibility: those with regular educational software, those with software designed for special needs populations, and those with software designed specifically around cognitive tasks. These three product lines are not mutually exclusive; in fact, from our compilation of information on software publishers, there are a few companies promoting software appropriate to all three categories.

The responsibility of the ARC in the distribution process is six-fold. The first responsibility is the technical specification of the capabilities of the instructional system. The second is the identification of the target population. The third is to establish and document the need for software of this type among the target population. The fourth responsibility is to highlight the unique features of the system. The fifth is to identify, based on the target population and needs assessments, marketing strategies that a software publisher could employ to increase networks of information dissemination. The final responsibility entails dissemination of information regarding the package's availability from the eventual marketer.

Through the field-testing process, the ARC will obtain the information necessary to meet the first of these outlined responsibilities. We have devised a detailed plan for evaluation in which participants will be comprised of persons with mental retardation, those with learning disabilities, and nonhandicapped persons. The instructional package will be evaluated in terms of its validity (that is, its ability to differentiate among ability groups) and its instructional value (that is, its ability to assist in the remediation process). This research will allow us to draw conclusions regarding the capabilities of the software for assessment and instructional purposes.

This research will also allow us to determine the breadth of effective application of the package across the populations in need. Although the appropriateness of this software for the population of nonhandicapped students is not the primary focus of the development project, a determination of such appropriateness is incorporated in the research design to provide a base of comparison for the handicapped users. As a result, there is a distinct possibility that the results will show that the larger market of non-handicapped students could derive enhancements in memory functioning through use of this software. Project staff conducted an extensive review of the cognitive psychology and special education literatures to identify the characteristics and nature of the populations to benefit from the aid. We have determined that approximately 55% of the students enrolled in special education classrooms in this country could benefit from this instructional package. This figure represents a market of close to two and one-half million students. More detailed information will be provided to the eventual marketer.

The ARC's extensive literature review also assisted in confirming the need for this type of software. One of the conclusions that can be drawn from this review is that the memory task around which the package is constructed is a valid and "pure" means by which to assess and train memory skills, and also that the computer is a near-perfect vehicle for this package because of its ability for logical analysis and its capacity to deal with large amounts of information in an interesting, effective, and efficient manner. This package accomplishes the marriage of a theoretical knowledge base to a practical, educationally-sound assessment and training package. The educational software marketplace currently fails to offer products reflecting large-scale efforts of this nature even though the desirability of such packages has been acknowledged by researchers, service delivery personnel and software publishers.

The ARC has the responsibility for highlighting the capabilities and positive attributes of the system to potential marketers and eventually to consumers. Detailed description of these features would be lengthy; only the major points will therefore be summarized as follows. First, the package offers both assessment and remedial components, and remediation is based logically on the assessment results. Second, this assessment permits the remediation components to be individually tailored to each student's needs. Third, the package uses the unique features of the computer to their fullest extent in assisting the child to understand and perform the required tasks; and fourth, the package employs innovations such as digitized speech output and light pen input to enhance its educational validity and appeal. Finally, the child's performance is analyzed and interpreted by the computer, which allows the teacher to obtain practical information for classroom purposes. This analysis and interpretation was designed with assistance from the leading cognitive psychologists in the country. The student's performance data is also permanently recorded for later review by the teacher.

In meeting the fifth responsibility, the ARC will suggest marketing strategies for use by a potential publisher based upon the factors addressed above. We will assist the publisher in highlighting this program's appeal and value to parents, teachers, and school districts. The ARC's final responsibility to the distributor will be met through the ARC's ability to disseminate information across a nationwide network. Through our network of 1300 state and local affiliates, our core of 200,000 members, our national publications including our national newspaper which is distributed six times a year to each member, our computerized technology data base, our national electronic mail and bulletin board system, and our Bioengineering Program, we are in a unique position to raise the awareness of school personnel on the availability of effective educational software in the marketplace.

MARKET DEMOGRAPHICS

The market for the software is not limited to school systems, but since this is the most likely "buyer" of this product, it would be pertinent to look at some statistics that will give an indication of the potential size of the market:

- In 1983, the special education field spent $10 billion on materials used by or on behalf of students.[1]
- Again, as of 1983, 330,000 microcomputers were in this country's schools.[1]
- By the end of 1986, there will be an estimated 1,025,000 microcomputers in public schools.[2] Approximately one-quarter of these, or 225,000, will be used in special education, benefitting 4.3 million special education students.[3]
- Special education accounts for 11% of the purchases in the educational computing marketplace.[3]
- In the 1984-85 school year, approximately 15 million students and 500,000 teachers used computers in the public schools.[3]
- Market analyst's projections of the amount to be spent on educational software in the 1987-88 school year range from $250 million to $500 million.[4]

Vest, C.R. (1983). Marketing and procurement of technology assisted learning systems. Micro Market Examiner, 1(3), 1.

Blaschke, C.L. (1982). Microcomputers in special education trends and projections. Journal of Special Education Technology.

Carol Daniels, LINC Resources (personal communication based on Quality Education Data survey, LINC survey of states, and Johns Hopkins survey).

Electronic Learning: The Guide to the Educational Marketplace (1984). Scholastic Inc.

These data indicate that there is clearly a significant market for this instructional system both now and in the future. Indeed, since this system appears to have application to both regular and special education programs, the potential market is enormous. The statistics cited above also indicate that substantial dollars are available to purchase equipment and other supplies that will enhance the learning of children and youth in classrooms across the country. Couple these factors with the viability of the cognitive software system and it would appear that the elements are in place to make it a successful product: a legitimate, attractive product, an educational void to be filled, recognition of the computer as a viable educational aid, and the availability of monies for purchase.

SUGGESTED MARKETING STRATEGIES:

1. Publicize availability of software through trade journals, newsletters, and teacher and education magazines.

2. Promote product at conventions of education, special education, rehabilitation, and computer technology professionals.

3. Organize demonstration seminars for educators.

4. Identify several schools in which to set up system on a trial basis at no charge to the school in an effort to elicit word of mouth publicity and testimonials.

FEEDBACK FROM COMMERCIAL PUBLISHERS

In compliance with SEP's request for feedback on the marketability, useability, and suitability of the product, the ARC identified several commercial software companies having product lines compatible with the software under development in this project. The company presidents and/or product developers were contacted and their participation was requested. Several companies denied our reqtest, citing most frequently the non-remunerative or time-consuming aspects of the task. Non-disclosure agreements were obtained from three marketers who agreed to participate.

These companies were sent an information packet including a statement of the problem addressed in the project, production and marketing plans, and the program narrative with accompanying documentation. The company representatives were guided in their review by the survey form created by the ARC project staff. The form was intended to direct the reviewers' comments to the specific aspects of suitability, useability, and marketability.

The evaluations we received were overwhelmingly positive. On a scale from 1 to 6, with 6 being the most favorable score, the software package received an average score of 5.33 in regard to its suitability, an average score of 5.67 in regard to its useability, and a unanimous rating of 6 in regard to its marketability.

Several suggestions were made regarding the eventual marketing of the package. One reviewer commented that while our development work was being done for the Apple and Commodore computers, we might want to eventually consider adapting a version for MS-DOS machines. In the time that has passed between our survey and the present, we received approval to modify our workplan to replace the Commodore version with an MS-DOS version in response to the current trend in the educational marketplace. One reviewer commented that we might suggest to the eventual marketer that a lightpen be included in the instructional package to enhance convenience and discourage piracy. Reviewers also suggested that the market for this software could be increased greatly if field tests demonstrated it to be effective in improving the cognitive skills of the non-handicapped student population. The prevailing opinion appeared to be that the instructional package was a good one, and that effective marketing was key to its viability as a commercial product.

LOCATING COMMERCIAL MARKETER

We will be working closely with the staff at LINC Resources to locate potential marketers for this product. LINC has in place a procedure by which software developers can locate appropriate potential publishers for their products. LINC provides assistance in the identification of appropriate publishers, in the preparation of informational documents to be sent to the publishers, and in the conduct of negotiations between the developer and marketer. LINC will advertise the availability of the software package to potential marketers through the LINC Notes newsletter. The availability of a resource such as LINC is invaluable to this project, and ARC staff will take full advantage of the services that LINC offers.

During the period in which beta tests are being completed, approximately 20 software companies will be contacted regarding their interest in marketing this product. Interested parties will be asked to demonstrate their willingness to collaborate by submitting to ARC project staff a statement of their company's capabilities in regard to advertising and production as well as its distribution channels. Project staff will review the responses to this request, and will select at least five companies with which to conduct more detailed discussions. The two companies with the most appropriate capabilities, product line, and demonstrated interest will travel to the ARC for a demonstration of the software and to review field test data and marketing ideas. During these visits the potential marketers will be asked to project an appropriate unit-price for the instructional system as well as an approximate expectation as to annual sales volume. The results of this process will be reported to the project officer in the final marketing report.

APPENDIX H

Software Coding Documentation

FUNCTION DOCUMENTATION

FOR

COGNITIVE TRAINING TOOL

TABLE OF CONTENTS

FUNCTION DESCRIPTION

Name: OVCTT.C

Synopsis: main()

Description: overlayed Cognitive Training Tool

VARIABLES:

Global:

char	bufdate[7]	/* buffer to get date */
int	clvl	/* current level */
op.dsk		
int	dfd	/* data file descriptor */
char	indev	/* lightpen or joystick */
int	meet_accuracy	
int	prep	
int	set	/* training set asmt,ff,cr,id,ch */
int	train	
int	xscale	
int	yscale	

Local:

c	recieve menu selection
d	randum number
struct grphcss grphcsl	graphics module parameters
level	current student level +1
oncemore	receives response from keeping to terminate assessment loop
struct sounds soundl	sound module parameters

Returns:

Functions called: asmt, chlp1, chlp2, chlp3, chlp4, close, crlp1, crlp2, crlp3, crlp4, cursdsp, delete, fflp1, fflp2, fflp3, fflp4, game, gameintr, idlp1, idlp2, idlp3, idlp4, keepgoin, l2j, l2l, l3a, l3b, l3c, l3d, l3e, lvl1, menu, opscrn13, preat, prepgm, printf, putchar, ran, tdate, trans, viewdata

FUNCTION DESCRIPTION

Name: ASMT.C

Synopsis:

```
asmt(l)
l       /* level of assessment training */
```

Description: do all assessment for required level

VARIABLES:

See Next Page

Returns:

Functions called: act_box, act_rltr, chkaccu, cursoff, decide_patn, dissolvebox, freewrd, getch, init, ioctlsa, loadwrd, loboxes, lseek, omeg, outlinebox, prin, printf, question, ready, rstrtwin, rule, set_blink, setmem, show_ltr, show_row, strncpy, wait, wipe, write

FUNCTION DESCRIPTION

Name: **ASMT.C**

Description:

VARIABLES:

```
     Global:
          struct          assessment_r ca,
          struct          log          clog
          int             clvl         /* current level */
          int             crlen        /* length of # of elements in
          struct          student      cs
          dfdslog
          int             listlen      /* total # of elements in an
                          assessment
          int             logfd        /* log  file descriptor */
          int             loglen       /* length of log structure */
          log
          int             loop         /* loop number 1-4 */
          char            lrbl[10]     /* letters held by lower row boxes */
          char            ltr[10]      /* letters to appear below lower row
          int             meet_accuracy
          int             prep
          int             set          /* training set asmt,ff,cr,id,ch */
          int             setctr       /* count of failures for set */
          struct          log          slog
     Local:
          c               recieve menu selection
          f               index
          i               index
          j               index
          logscore        accuracy score
          ok              flag indicating upper & lower box match
          score           accuracy score
          time            pause time to select box or letter
          which           letter selected
```

FUNCTION DESCRIPTION

Name: CHLP1.C

Synopsis:

```
chlp1(l)
l   /* level of assessment training */
```

Description: chaining loop #1 DEMO

VARIABLES:

Global:

int	crlen	/* length of # of elements in
int	listlen	/* total # of elements in an assessment
int	loop	/* loop number 1-4 */
char	lrbl[]	/* letters held by lower row boxes */
char	ltr[]	/* letters to appear below lower row
int	set	/* training set asmt,ff,cr,id,ch */

Local:

i	index for element within group
index	index for rehearsal line
j	index for number of rehearsals
k	index for letter within a group
struct ansistr ans	parameters for screen manipulation
temp[]	buffer for rehearsal line

Returns:

Functions called: ansiscr, decide_patn, freewrd, init, ioctlsa, loadwrd, loboxes, prin, putchar, question, ready, rstrtwin, rule, set_blink, set_box_clr, show_row, simpler, stop_blink, wait, wipe

FUNCTION DESCRIPTION

Name: CHLP2.C

Synopsis: chlp2(l)
l /* level of assessment training */

Description: chaining loop #2

VARIABLES:

See Next Page

Returns:

Functions called: act_box, act_rbox, act_rltr, chkaccu, cursoff, decide_patn, dissolvebox, init, ioctlsa, loadwrd, loboxes, lseek, mus, outlinebox, prin, question, ready, rstrtwin, rule, set_blink, setmem, show_ltr, show_row, stop_blink, wait, wipe, write

FUNCTION DESCRIPTION

Name: CHLP2.C

Description:

VARIABLES:

```
     Global:
          struct            log        clog
          int               crlen      /* length of # of elements in
          cs
          int               dfd        /* data file descriptor */
          int               itr2ctr    /* iteration number for loop 2 */
          int               listlen    /* total # of elements in an
                            assessment
          int               logfd      /* log  file descriptor */
          int               loglen     /* length of log structure */
          int               loop       /* loop number 1-4 */
          char              lrbl[]     /* letters held by lower row boxes */
          int               meet_accuracy
          int               set        /* training set asmt,ff,cr,id,ch */
          struct            log        slog
          char              urbl[]     /* letters held by upper row boxes */
     Local:
          c                 recieve menu selection
          first             flag parameter for box selection
          i                 index for element within group
          incorrect         counter for number of misses
          index             index for rehearsal line
          j                 index for number of rehearsals
          k                 index for letter within group
          match             flag indicating match
          score             accuracy score
          temp[]            buffer for rehearsal line
          which             letter selected
          time              pause time to select a box or letter
          wrongctr          iteration counter for training loop
```

FUNCTION DESCRIPTION

Name: CHLP3.C

Synopsis:
```
chlp3(l)
l  /* level of assessment training */
```

Description: chaining loop #3

VARIABLES:

Global:

```
struct        log            clog
int           crlen          /* length of # of elements in
int           listlen        /* total # of elements in an
              assessment
int           loglen         /* length of log structure */
char          lrbl[10]       /* letters held by lower row boxes */
int           meet_accuracystruct log slog
char          urbl[10]       /* letters held by upper row boxes */
```

Local:

```
c             receive menu selection
first         flag parameter for box selection
i             index for element within group
incorrect     counter for number of misses
index         index for rehearsal line
j             index for number of rehearsals
match         flag indicating match
score         accuracy score
time          pause time to select a box or letter
which         letter selected
wrongctr      iteration counter for training loop
```

Returns:

Functions called:

act_box, act_rbox, act_rltr, chkaccu, cursoff, decide_patn, dissolvebox, freewrd, init, ioctlsa, loadwrd, loboxes, mus, outlinebox, prin, question, ready, rstrtwin, rule, set_blink, setmem, show_ltr, show_row, stop_blink, wait, wipe, write

FUNCTION DESCRIPTION

Name: CHLP4.C

Synopsis: chlp4(l)
l /* level of assessment training */

Description: chaining loop #4

VARIABLES:

See Next Page

Returns:

Functions called: act_box, act_rlt, chkaccu, cursoff, decide_patn, dissolvebox, freewrd, getch, init, ioctlsa, loadwrd, loboxes, lseek, mus, outlinebox, prin, printf, question, ready, rstrtwin, setmem, show_ltr, show_row, wait, wipe, write

FUNCTION DESCRIPTION

Name: <u>CHLP4.C</u>

Description:

VARIABLES:

Global:

```
struct      log        clog
int         crlen      /* length of # of elements in
struct      student    cs
int         dfd        /* data file descriptor */
int         listlen    /* total # of elements in an
            assessment
int         loop       /* loop number 1-4 */
char        lrbl[]     /* letters held by lower row boxes */
char        ltr[]      /* letters to appear below lower row
int         set        /* training set asmt,ff,cr,id,ch */
struct      log        slog
```

Local:

```
c           receive menu selection
first       flag parameter for box selection
i           index for element within a group
incorrect   counter for number of misses
index       index for rehearsal line
j           index for number of rehearsals
k           index for letter within group
match       flag indicating match
score       accuracy score
time        pause time to delect a box or letter
which       letter selected
wrongctr    iteration counter for training loop
```

FUNCTION DESCRIPTION

Name: CRLP1.C

Synopsis:
```
crlp1(l)
l   /* level of assessment training*/
```

Description: cumulative rehearsal loop #1

VARIABLES:

Global:

```
struct      assessment_r ca,
int         crlen      /* length of # of elements in
struct      student    cs
int         dfd        /* data file descriptor */
int         listlen    /* total # of elements in an
            assessment
int         loglen     /* length of log structure */
int         loop       /* loop number 1-4 */
int         set        /* training set asmt,ff,cr,id,ch */
struct      log        slog
```

Local:

```
i           index for element within group
i2          index for recall list
index       index for rehearsal line
j           index for number of rehearsals
k           index for letter within group
temp[]      buffer for rehearsal line
```

Returns:

Functions called: decide_patn, freewrd, getch, init, ioctlsa, loadwrd, loboxes, lseek, outlinebox, prin, printf, question, ready, rstrtwin, set_blink, set_box_clr, setmem, show_ltr, show_row, stop_blink, wait, wipe, write

FUNCTION DESCRIPTION

Name: CRLP3.C

Synopsis:

```
crlp3(l)
l   /* level of assessment training */
```

Description: cumulative rehearsal loop #3

VARIABLES:

```
     Global:
          struct          assessment_r ca,
          struct          log        clog
          int             crlen      /* length of # of elements in
          int             dfd        /* data file descriptor */
          int             itr        /* iteration number */
          int             loglen     /* length of log structure */
          int             loop       /* loop number 1-4 */
          char            lrbl[]     /* letters held by lower row boxes */
          int             set        /* training set asmt,ff,cr,id,ch */
          struct          log        slog
          char            urbl[]     /* letters held by upper row boxes */
     Local:
          first           flag parameter for box selection
          i               index for element within group
          incorrect       counter for number of misses
          index           index for rehearsal line
          j               index for number of rehearsal
          match           flag indicating match
          score           accuracy score
          time            pause time to select a letter or box
          which           letter selected
          wrongctr        iteration counter for training loop
```

Returns:

Functions called:

act_box, chkaccu, cursoff, decide_patn, dissolvebox, freewrd, getch, ioctlsa, loadwrd, loboxes, lseek, mus, outlinebox, prin, printf, question, ready, rstrtwin, set_blink, set_box_clr, setmem, show_ltr, show_row, stop_blink, wait, wipe, write

FUNCTION DESCRIPTION

Name: CRLP2.C

Synopsis:
```
crlp2(l)
l   /* level of assessment training */
```

Description: cumulative rehearsal loop #2

VARIABLES:

Global:

struct	log	clog	
struct	student	cs	
int	dfd	/* data file descriptor */	
int	listlen	/* total # of elements in an assessment	
int	loglen	/* length of log structure */	
char	lrbl[]	/* letters held by lower row boxes */	
int	set	/* training set asmt,ff,cr,id,ch */	
struct	log	slog	
char	urbl[]	/* letters held by upper row boxes */	

Local:

first	flag parameter for box selection
incorrect	counter for number of misses
index	index for rehearsal line
j	index for number of rehearsals
k	index for letter within group
match	flag indicating match
score	accuracy score
temp[]	buffer for rehearsal line
time	pause time to select box or letter
which	letter selected
wrongctr	iteration counter for training loop

Returns:

Functions called: act_box, act_rltr, chkaccu, decide_patn, freewrd, getch, init, ioctlsa, loadwrd, loboxes, lseek, mus, outlinebox, prin, printf, question, ready, rstrtwin, set_blink, setmem, show_ltr, show_row, stop_blink, wait, wipe, write

FUNCTION DESCRIPTION

Name: CRLP4.C

Synopsis:

```
crlp4(l)
l   /* level of assessment training */
```

Description: cumulative rehearsal loop #4

VARIABLES:

See Next Page

Returns:

Functions called: ct_box, act_rltr, chkaccu, cursoff, decide_pat, dissolvebox, freewrd, getch, init, loadwrd, loboxes, lseek, mus, outlinebox, prin, printf, question, ready, rstrtwin, setmem, show_ltr, show_row, wait, wipe, write

FUNCTION DESCRIPTION

Name: CRLP4.C

Description:

VARIABLES:

Global:

int	crlen	/* length of # of elements in
int	listlen	/* total # of elements in an assessment
int	logfd	/* log file descriptor */
int	loglen	/* length of log structure */
int	loop	/* loop number 1-4 */
char	lrbl[]	/* letters held by lower row boxes */
int	set	/* training set asmt,ff,cr,id,ch */
int	setctr	/* count of failures for set */
struct	log	slog
char	urbl[]	/* letters held by upper row boxes */

Local:

c	receive menu selection
first	flag parameter for box selection
i	indx for element within group
incorrect	counter for number of misses
j	index for number of rehearsals
k	index for letter within group
match	flag indicating match
missing	flag indicating correct disk is missing
score	accuracy score
time	pause time to select a box or letter
which	letter selected
wrongctr	iteration counter for training loop

FUNCTION DESCRIPTION

Name: **DELETE.C**

Synopsis: delete()

Description: delete a student's performance record

VARIABLES:

Global:

struct	student	cs

Local:

c	receive menu selection

Returns:

Functions called: del_std, getcomm, ioctlsa, printf

FUNCTION DESCRIPTION

Name: FFLP1.C

Synopsis: fflp1(l)

```
l    /* level of assessment training */
```

Description: fast finish loop #1

VARIABLES:

```
Global:
     int              crlen      /* length of # of elements in
     int              listlen    /* total # of elements in an
                      assessment
     int              loop       /* loop number 1-4 */
     char             ltr[]      /* letters to appear below lower row
     int              set        /* training set asmt,ff,cr,id,ch */
     int              train
Local:
     buf[]            buffer for prompt line
     i                index for occurance within group
     i2               index for recall list
```

Returns:

Functions called: decide_patn, freewrd, init, ioctlsa, loadwrd, loboxes, outlinebox, prin, question, ready, rstrtwin, set_blink set_box_clr, show_ltr, show_row, sprintf, wait, wipe

FUNCTION DESCRIPTION

Name: FFLP2.C

Synopsis: fflp2(1)
1 /* level of assessment training */

Description: fast finish loop #2

VARIABLES:

See Next Page

Returns:

Functions called: act_box, chkaccu, cursoff, decide_patn, dissolvebox, freewrd, getch, init, ioctlsa, loadwrd, loboxes, lseek mus, outlinebox, prin, printf, question, rstrtwin, set_blink, set_box_clr, setmem, show_ltr, show_row, sprintf, stop_blink, wait, wipe, write,

FUNCTION DESCRIPTION

Name: FFLP2.C

Description:

VARIABLES:

Global:

```
struct      log         clog
int         crlen       /* length of # of elements in
int         fflen       /* length of list that holds len# of
int         listlen     /* total # of elements in an
            assessment
int         logfd       /* log  file descriptor */
int         loglen      /* length of log structure */
int         loop        /* loop number 1-4 */
char        lrbl[]      /* letters held by lower row boxes */
int         meet_accuracy
int         set         /* training set asmt,ff,cr,id,ch */
struct      log         slog
char        urbl[]      /* letters held by upper row boxes */
```

Local:

```
buf[]       buffer for prompt line
first       flag parameter for box selection
i           index for element within group
incorrect   counter for number of misses
j           index for number of rehearsals
match       flag indicating match
score       accuracy score
time        pause time to select a box or letter
which       letter selected
wrongctr    iteration counter for training loop
```

FUNCTION DESCRIPTION

Name: FFLP3.C

Synopsis:

```
fflp3(1
1   /* level of assessment training */
```

Description: fast finish loop #3

VARIABLES:

See Next Page

Returns:

Functions called: _exit, act_box, act_rltr, chkaccu, cursoff, decide_patn, dissolvebox, freewrd, getch, init, ioctlsa, loadwrd, loboxes, lseek, mus, outlinebox, prin, printf, questio, ready, rstrtwin, set_blink, setmem, show_ltr, show_row, sprintf, stop_blink, wait, wipe, write

FUNCTION DESCRIPTION

Name: FFLP3.C

Description:

VARIABLES:

```
Global:
    struct      assessment_r ca,
    struct      log         clog
    int         crlen       /* length of # of elements in
    struct      student     cs
    int         listlen     /* total # of elements in an
                assessment
    int         loglen      /* length of log structure */
    int         loop        /* loop number 1-4 */
    char        lrbl[]      /* letters held by lower row boxes */
    char        ltr[]       /* letters to appear below lower row
    int         meet_accuracy
     int        set         /* training set asmt,ff,cr,id,ch */
    struct      log         slog
    char        urbl[]      /* letters held by upper row boxes */
Local:
    buf[]       buffer for prompt line
    first       flag parameter for box selection
    i           index for element within group
    incorrect   counter for number of misses
    j           index for number of rehearsals
    match       flag indicating match
    score       accuracy score
    time        pause time to select a box or letter
    which       letter selected
    wrongctr    iteration counter for training loop
```

FUNCTION DESCRIPTION

Name: FFLP4.C

Synopsis:

```
fflp4(l
l   /* level of assessment training */
```

Description: fast finish loop #4

VARIABLES:

See Next Page

Returns:

Functions called: act_box, act_rltr, chkaccu, cursoff, decide_patn, dissolvebox, freewrd, getch, init, ioctlsa, loadwrd, loboxes, lseek, mus, outlinebox, prin, printf, question, ready, rstrtwin, setmem, show_ltr, show_row, sprintf, wipe, write

FUNCTION DESCRIPTION

Name: **FFLP4.C**

Description:

VARIABLES:

Global:

```
struct      assessment_r ca,
struct      log         clog
int         crlen       /* length of # of elements in
struct      student     cs
int         dfd         /* data file descriptor */
int         listlen     /* total # of elements in an
            assessment
int         logfd       /* log file descriptor */
int         loglen      /* length of log structure */
int         loop        /* loop number 1-4 */
char        lrbl[]      /* letters held by lower row boxes */
char        ltr[]       /* letters to appear below lower row
int         meet_accuracy
int         set         /* training set asmt,ff,cr,id,ch */
int         setctr      /* count of failures for set */
struct      log         slog
char        urbl[]      /* letters held by upper row boxes */
```

Local:

```
buf[]       buffer for prompt line
c           receive menu selection
first       flag paramete for box selection
i           index for occurance within group
i2          index for recall list
incorrect   counter for number of misses
j           index for number of rehearsals
k           index for letter within group
match       flag indicating match
score       accuracy score
time        pause time to select a box or letter
which       letter selected
```

GAME.C

```
game()
indev     input device
level     game difficulty level
time      time limit
xscale    horizontal scaling factor
yscale    vertical scaling factor
```

VARIABLES:

```
    Static:
        struct startype stars[]
    Local:
        ANSI                ans
        fd
        hitbuf[]
        i
        num
        nums[][]
        rocbuf[]
        shipbuf[]
        shipcnt
        shipspd
        x
        y
```

game2, gameinit, wait

FUNCTION DESCRIPTION

Name: GAMEINTR.C

Synopsis: gameintr()

Description: Introduction and primer for game

VARIABLES:

Static:

bx[]	pl[]
by[]	vx[]
gx[]	vy[]
gy[]]	working[]
nums[]	
ox[]	
oy[]	

Local:

ANSI	ans
base	pointer to rocket base shape
cnt	random number for ship movement
curs	pointer to cursor shape
i	general purpose index
num	pointer to array of ASCII digits
objlng	object length box vocabulary
rbase	rocket base shape
rcurs	cursor shape
x	horizontal character position
y	vertical line position

Returns:

Functions called: ansiscr, drawnum, freewrd, plt, pltchr, shipmov, speak, wait, whistle

FUNCTION DESCRIPTION

Name: IDLP1.C

Synopsis:
```
idlp1(1)
1   /* level of assessment training */
```

Description: interpolated loop #1

VARIABLES:

See Next Page

Returns:

Functions called: act_box, act_rltr, chkaccu, cursof, decide_patn, dissolvebox, freewrd, getch, init, listlen, loadwrd, loboxes, lseek, mus, outlinebox, prin, printf, qrestion, ready, rstrtwin, set_box_clr, setmem, show_ltr, show_row, wait, wipe, write

FUNCTION DESCRIPTION

Name: IDLP1.C

Description:

VARIABLES:

```
     Global:
          struct         assessment_r ca,
          struct         log        clog
          int            crlen      /* length of # of elements in
          struct         student    cs
          int            dfd        /* data file descriptor */
          int            logfd      /* log  file descriptor */
          int            loglen     /* length of log structure */
          int            loop       /* loop number 1-4 */
          char           lrbl[]     /* letters held by lower row boxes */
          char           ltr[]      /* letters to appear below lower row
          int            meet_accuracy
          int            set        /* training set asmt,ff,cr,id,ch */
          struct         log        slog
          char           urbl[]     /* letters held by upper row boxes */
     Local:
          first          flag parameter for box selection
          i              general purpose index
          incorrect      counter for number of misses
          match          flag indicating match
          score          score accuracy
          temp[]         buffer for rehearsal line
          time           pause time to select box or letter
          which          letter selected
          wrongctr       iteration counter for training loop
```

FUNCTION DESCRIPTION

Name:	IDLP2.C
Synopsis:	idlp2(l) l /* level of assessment training */
Description:	interpolated loop #2

VARIABLES:

See Next Page

Returns:

Functions called:

act_box, act_rltr, chkaccu, cursoff, decide_patn, dissolvebox, freewrd, getch, init, ioctlsa, loadwrd, loboxes, lseek, mus, outlinebox, prin, printf, question, ready, rstrtwin, setmem, show_ltr, show_ltr. show_row, wait, wipe, write

FUNCTION DESCRIPTION

Name: IDLP2.C

Description:

VARIABLES:

```
Global:
        struct          assessment_r ca,
        struct          log          clog
        int             crlen        /* length of # of elements in
        struct          student      cs
        int             dfd          /* data file descriptor */
        int             listlen      /* total # of elements in an
                        assessment
        int             logfd        /* log  file descriptor */
        int             loglen       /* length of log structure */
        int             loop         /* loop number 1-4 */
        char            lrbl[]       /* letters held by lower row boxes */
        char            ltr[]        /* letters to appear below lower row
        int             meet_accuracy
        int             set          /* training set asmt,ff,cr,id,ch */
        struct          log          slog
        char            urbl[]       /* letters held by upper row boxes */
Local:
        first           flag parameter for box selection
        i               general purpose index
        incorrect       counter for number of misses
        j               index for number of rehearsals
        k               index for letter within group
        match           flag indicating match
        scc             score accuracy
        temp[]          buffer for rehearsal line
        time            pause time to select box or letter
        which           letter selected
        wrongctr        iteration counter for training loop
```

FUNCTION DESCRIPTION

Name: IDLP3.C

Synopsis:

```
idlp3(1)
1   /* level of assessment training */
```

Description: interpoplated loop #3

VARIABLES:

See Next Page

Returns:

Functions called: act_box, act_rltr, chkaccu, cursoff, decide_patn, dissolvebox, freewrd, getch, init, ioctlsa, loadwrd, loboxes, lseek, mus, outlinebox, prin, printf, question, ready, rstrtwin, setmem, show_ltr, show_row, wait, wipe, write

FUNCTION DESCRIPTION

Name: IDLP3.C

Description:

VARIABLES:

```
Global:
     struct          assessment_r ca,
     struct          log          clog
     int             crlen        /* length of # of elements in
     struct          student      cs
     int             dfd          /* data file descriptor */
     int             listlen      /* total # of elements in an
                     assessment
     int             logfd        /* log  file descriptor */
     int             loglen       /* length of log structure */
     int             loop         /* loop number 1-4 */
     char            lrbl[]       /* letters held by lower row boxes */
     char            ltr[]        /* letters to appear below lower row
     int             meet_accuracy
     int             set          /* training set asmt,ff,cr,id,ch */
     struct          log          slog
     char            urbl[]       /* letters held by upper row boxes */
Local:
     delaysec        seconds to delay error recall
     first           flag parameter for box selection
     i               general purpose index
     incorrect       counter for number of misses
     j               index for number of rehearsals
     k               index for letter within group
     match           flag indicating match
     score           score accuracy
     temp[]          buffer for rehearsal line
     time            pause time to select box or letter
     which           letter selected
     wrongctr        iteration counter for training loop
```

FUNCTION DESCRIPTION

Name: IDLP4.C

Synopsis:

```
idlp4(l)
l    /* level of assessment training */
```

Description: interpolated loop #4

VARIABLES:

See next Page

Returns:

Functions called: act_box, act_rltr, chkaccu, cursoff, decide_patn, dissolvebox, freewrd, getch, init, ioctlsa, loadwrd, loboxes, lseek, mus, outlinebox, prin, printf, question, ready, rstrtwin, setmem, show_ltr, show_row, wait, wipe, write

FUNCTION DESCRIPTION

Name:

IDLP4.C

Description:

VARIABLES:

```
     Global:
          struct          assessment_r ca,
          int             crlen      /* length of # of elements in
          struct          student    cs
          int             itrctr     /* count of correct iterations for
                          set
          int             listlen    /* total # of elements in an
                          assessment
          int             logfd      /* log  file descriptor */
          int             loglen     /* length of log structure */
          int             loop       /* loop number 1-4 */
          char            lrbl[]     /* letters held by lower row boxes */
          int             meet_accuracy
          int             set        /* training set asmt,ff,cr,id,ch */
          struct          log        slog
          char            urbl[]     /* letters held by upper row boxes */
     Local:
          c               receive menu selection
          first           flag parameter for box selection
          i               general purpose index
          incorrect       counter for number of misses
          j               index for number of rehearsals
          k               index for letter within group
          match           flag indicating match
          missing         flag to indicate disk is missing
          score           score accuracy
          time            pause time to select box or letter
          which           letter selected
          wrongctr        iteration counter for training loop
```

FUNCTION DESCRIPTION

Name: L2J.C

Synopsis:
```
L2J(1)
1   /* level of assessment training */
```

Description: familiarize the user with Joy Stick

VARIABLES:

Global:

char	ltr[]	/* letters to appear below lower row

Local:

time	pause time to select a box or letter
which	letter selected

Returns:

Functions called: act_rltr, cursoff, jystk, loboxes, prin, rstrtwin, set_blink, show_ltr, show_row, stop_blink, wait, wipe

FUNCTION DESCRIPTION

Name: L2L.C

Synopsis:

```
L2L(l)
l   /* level of assessment training */
```

Description: familiarize user with lightpen

VARIABLES:

Global:

char	ltr[]	/* letters to appear below lower row

Local:

time	pause time to select a box or letter
which	letter selected

Returns:

Functions called: act_rltr, loboxes, ltpen, prin, rstrtwin, set_blink, show_ltr, show_row, stop_blink, wait, wipe

FUNCTION DESCRIPTION

Name: L3A.C

Synopsis: L3A()

Description: letter display primer

VARIABLES:

Global:

```
int        crlen      /* length of # of elements in
int        fflen      /* length of list that holds len# of
int        listlen    /* total # of elements in an
                      assessment
```

Local:

```
ANSI       ans
c          escape character
```

Returns:

Functions called: actandshow, getchar, initnum, ioctlsa, loadwrd, loboxes, prin, printf, putchar, ready, rstrtwin, set_blink, set_box_clr, show_ltr, stop_blink, wait, wipe

FUNCTION DESCRIPTION

Name: L3B.C

Synopsis: L3B()

Description: recall primer demonstration

VARIABLES:

Global:

```
int       crlen     /* length of # of elements in
int       fflen     /* length of list that holds len# of
```

Local:

```
struct    ansistr   ans
c         escape character
```

Returns:

Functions called:

ansiscr, clr_ltr, initnum, ioctlsa, ioboxes, movandshow, outlinandshow, prin, question, rstrtwin, rule, set_blink, show_ltr, stop_blink, truck, wait, wipe,

FUNCTION DESCRIPTION

Name: L3C.C

Synopsis: L3C()

Description: recall primer and exercise

VARIABLES:

Local:

time	pause time to select a box or letter
which	letter selected

Returns:

Functions called: act_rbox, cursoff, loboxes, prin, rstrtwin, rule, set_blink, stop_blink, wait, wipe

FUNCTION DESCRIPTION

Name: L3D.C

Synopsis: l3d()

Description: letter display and recall exercise

VARIABLES:

```
     Global:
          int            crlen      /* length of # of elements in
          int            fflen      /* length of list that holds len# of
          int            listlen    /* total # of elements in an
                         assessment
          int            loop       /* loop number 1-4 */
          char           lrbl[]     /* letters held by lower row boxes */
          int            set        /* training set asmt,ff,cr,id,ch */
          char           urbl[0]    /* letters held by upper row boxes */
     Local:
          i              index for box position
          match          flag indicating match
          n
          struct         ansistr   ans
          time           pause time to select a box or letter
          which          letter selected
          whichone       index for box tor recall
```

Returns:

Functions called: act_box, act_rltr, ansiscr, clr_ltr, dissolvebox, initnum, loboxes, outlinebox, prin, putchar, question, ready, rstrtwin, set_blink, set_box_clr, show_ltr, show_row, stop_blink, wait, wipe

FUNCTION DESCRIPTION

Name: L3E.C

Synopsis: L3e()

Description: assessment exercise

VARIABLES:

Global:

```
int     clvl       /* current level */
int     crlen      /* length of # of elements in
int     dfd        /* data file descriptor */
int     fflen      /* length of list that holds len# of
int     listlen    /* total # of elements in an
                   assessment
int     loop       /* loop number 1-4 */
char    lrbl[]     /* letters held by lower row boxes */
int     set        /* training set asmt, ff, cr, id, ch
char    urbl[]     /* letters held by upper row boxes */
```

Local:

```
c          receive menu selection
f          index
i          index for element within group
j          index for number of rehearsals
match      flag indicating match
missing    flag to indicate disk is missing
time       pause time
which      letter selected
```

Returns:

Functions called:

act_box, act_rltr, dissolvebox, getch, initnum, loboxes, lseek, outlinebox, prin, printf, question, ready, rstrtwin, rule, set_blink, show_ltr, show_row, stop_blink, wait, wipe, write

FUNCTION DESCRIPTION

Name: LVL1.C

Synopsis: lvl1()

Description: brief introduction

Returns:

Functions called:

freewrd, hello, loadwrd, prin, rstrtwin, wait , wipe

FUNCTION DESCRIPTION

Name: **MENU.C**

Synopsis: menu()

Description: Main menu for CTT program

VARIABLES:

Global:

struct	student cs

Local:

c	menu selection

Returns:

Functions called:

cursdsp, getch, ioctlsa, printf, putchar, warning, window

FUNCTION DESCRIPTION

Name: OPSCRN13.C

Synopsis: opscrn13()

Description: manipulate screen13

VARIABLES:

Local:

c	menu selection
flag	flag indicating a successful student record creation

Returns:

Functions called:

get_data, getch, ioctlsa, opscrn30, printf, putchar, review, revlist

467

FUNCTION DESCRIPTION

Name: PREAT.C

Synopsis: preat()

Description: prepare for assessment and training

VARIABLES:

See Next Page

Returns:

Functions called:

access, cursdsp, found, getch, init_stik, initpen, ioctlsa, lseek, printf, read, rstrtwin, selstrt, selstrtp, window

FUNCTION DESCRIPTION

Name: PREAT.C

Description:

VARIABLES:

Global:

```
int        SCREEN
int        VOICE
char       bufdate[7] /* buffer to get date */
int        clvl       /* current level */
struct     personal_r cp
struct     personal_r *cpr
struct     student    *csp
int        currs      /* current student record # */
int        dfd        /* data file descriptor */
char       indev      /* lightpen or joystick */
int        itrctr     /* count of correct iterations for
           set
int        loglen     /* length of log structure */
int        loop       /* loop number 1-4 */
int        maxtime    /* voice or text output */
int        prep       flag to indicate pre-post assessment
int        set        /* training set asmt,ff,cr,id,ch */
int        setctr     /* count of failures for set */
struct     log        slog
```

Local:

```
c          response to prompts
missing    flag to indicate correct disk is missing
```

FUNCTION DESCRIPTION

Name: <u>**PREPGM.C**</u>

Synopsis:

```
prepgm()
nstud       number of students
sfd         student file data pointer
```

Description:

Say hello and open files before program

VARIABLES:

```
Global:
     int            dfd        /* data file descriptor */
     int            sfd        /* student file descriptor */
     int            logfd      /* log  file descriptor */
Local:
     ans
     buf            command line buffer
     cmnd           command line buffer
     i              index for box position
     sufix          sufix to log name
```

Returns:

Functions called:

access, ansiscr, close, creat, cursdsp, getch, ioctlsa, open, printf, read, sprintf, system

FUNCTION DESCRIPTION

Name: T_DATE()

Synopsis: t_date()

Description: To read today's date from keyboard

VARIABLES:

Global:
c
i
ok
Static:
msg

Returns:

Functions called: cursdsp, getch, ioctlsa, printf, putchar

471

FUNCTION DESCRIPTION

Name: TRANS.C

Synopsis:

```
trans(l)
l   /* level of assessment training */
```

Description: this procedure decides if the user can go to next level

VARIABLES:

Global:

int	crlen	/* length of # of elements in
int	listlen	/* total # of elements in an assessment

Local:

i	general purpose index

Returns:

Functions called: decide_patn, freewrd, ioctlsa, loadwrd, loboxes, prin, set_box_clr, wait, wipe

FUNCTION DESCRIPTION

Name: VIEWDATA.C

Synopsis: viewdata()

Description: This routine views a student's assessment performance analysis

VARIABLES:

Global:

struct	student	cs	

Local:

c	menu response		
struct databuf	vd	parameter for viewdata	

Returns:

Functions called:

getch, graphs, printf, shrink, table, vmenu, wipe

FUNCTION DESCRIPTION

Name: <u>L3A.C</u>

Synopsis: actandshow(n,which)

```
int n          number of boxes
int which      which box
```

Description: Accept a box selection and display a letter

VARIABLES:

Local:

int time

Returns:

Functions called: act_box, show_ltr, stop_blink

474

FUNCTION DESCRIPTION

Name: ACTLTR.C

Synopsis:

```
act_rbox(n,c,k,f)
char c /* indicates row 'h' or 'l'*/
int  f     flag indicating first time
int *k /* number of selected box*/
int n  /* number of boxes */
```

Description: This function selects one of 'n' boxes, setting k to the number of the selected box, and returning the time in increments of 50 msec if it is less than 30 sec else it returns -1.*/

VARIABLES:

Global:
stikss

Local:

int c	pause time
int x	horizontal character position
int y	vertical line position

Returns: return(t)

Functions called: ansiscr, lightpen, stik

475

FUNCTION DESCRIPTION

Name: ACTLTR.C

Synopsis:

```
act_rltr(n,k,f)
int *k      pointer to number of selected box
int f       flag indicating first time
int n       number of boxes
```

Description:

This function selects a character from 'n' characters, sets k=postion of selected letter and returns time in increments of 50 msec if less than 30 sec else returns -1./

VARIABLES:

Global:

```
stikss
```

Local:

```
int c              pause time
int x              horizontal character position
int y              vertical line position
```

Returns: return(c)

Functions called: ansiscr, lightpen, stik

476

FUNCTION DESCRIPTION

Name: CHKACCU.C

Synopsis: chkaccu(logp)

See Next Page

Description: according to current level decides corresponing circular recall accuracy and omega squared requirements

VARIABLES:

Local:

int pass flag indicating a passing score

Returns:

return(pass)

Functions called:

477

FUNCTION DESCRIPTION

Name:

CHKACCU.C

Synopsis:

```
chkaccu(logp)
struct log *logp    pointer to log data
typedef struct log {
struct student std  /* student identification data
int date  /* assessment date; mmddy
int set   /* training set asmt,ff,cr,id,ch
int loop  /* loop number 1-4
int itr   /* iteration number
struct trial   trl  /* data for recall trial
float     omega2cr  /* omega squared for cumulative rehersal
float     omega2f   /* omega squared represented as a float
int  crscore    /* # right for cumulative rehersal
int  ffscore    /* # right for fast finish
```

Returns:

Functions called:

478

FUNCTION DESCRIPTION

Name: **CLRLTR.C**

Synopsis:

```
clr_ltr(pos,num,which,chr,forgrnd,backgrnd)
char chr          letter to display
char pos          high or low row
int backgrnd      background color
int forgrnd       foreground color
int num           number of letters
int which         box in which letter is to be displayed
```

Description:

color letter paints a char any color and returns to blk/wht

VARIABLES:

Local:

```
ANSI ans          parameters for screen display
int x             horizontal character position
int y             vertical line position
```

Returns:

Functions called:

ansiscr, putchar

479

FUNCTION DESCRIPTION

Name: STIK.C

Synopsis: cursoff()

Description:

VARIABLES:

LOCAL:

ANSI	ans	parameters for screen display

Returns:

Functions called:

ansiscr, putchar

480

FUNCTION DESCRIPTION

Name:

DELSTD.C

Synopsis:

```
del_std(name)
char *name     pointer for name of student to delete
```

Description:

Deletes all identifying and assessment information for a selected student

VARIABLES:

Global:

struct student	bufcsp	pointer to current student record
int currd	current data record #	
int currs	current student record #	
int nstud	# of students	
int sfd	student file descriptor	

Local:

int n	general purpose index

Returns:

return(-1) not found
return(currs) current student number

Functions called:

found, lseek, read, write

FUNCTION DESCRIPTION

Name:
LOADWRD.C

Synopsis:
freewrd()

Description:

Free up memory used by vocabulary

Returns:

Functions called: free, getch, printf

482

FUNCTION DESCRIPTION

Name:

GAME2.C

Syropsis:

```
ovmain(dummy,level,shipbuf,hitbuf,rocbuf,nms,time,star,xscale,ysc
ale,indev,shipspd,shipcnt)
struct startype {
        int color;
        int x,y;
        int cx,cy,dx,dy,mx,my;
        int on;
        int cnt,timeon,timeoff;
}
char hitbuf[]
char indev
char rocbuf[]
char shipbuf[]
int level
int nms
int shipcnt
int shipspd
int time
int xscale
int yscale
struct startype star[]
```

Returns:

Functions called:

ansiscr, blnkstar, drawnum, drawstr, erasenum, intvls, loadrocket, movrocket, pltchr, renew, shipmov, starhit, stick, update

483

FUNCTION DESCRIPTION

Name: GAME2.C

Description:

Do you want to play a game?

VARIABLES:

Static:

char	*curspt	
char	CURS	
char	nums[]	
char	scores[]	
char	tempchar	
char	times[]	
int	jx	
int	jy	
int	maxtime	
int	pba	

Local:

ans;	ansiscr parameters	for screen display
char	*base	pointer to base shape
char	*strp	pointer to star shape
char	rbase	base shape
char	strl	start shape
int	curtime	remaining time
int	oldscore	
int	score	
int	shiphit	flag indicating that a ship has been hit
int	alloff	flag indicating all stars are off
int	first	first time flag for joystick
int	i	general purpose index
int	j	general purpose index
int	renewed	flag indicating stars have been renewed
int	rx1	rocket left most character position
int	rx2	rocket right most character position
int	ry1	rocket top most line position
int	ry2	rocket bottom most line position
int	t1	receives intrvls return to compute time
int	t2	receives intrvls return to compute time
int	tim	general purpose iteration counter
int	x	horizontal character position
int	y	vertical line position

484

FUNCTION DESCRIPTION

Name: **GAMEINIT.C**

Synopsis:

```
ovmain(dummy,level,star,shipspd,shipcnt)
struct startype
        int color;
        int x,y;
        int cx,cy,dx,dy,mx,my;
        int on;
        int cnt,timeon,timeoff;
int *shipspd
int level
struct startype *star
unsigned *shipcnt
```

Description:

Initialize game parameters

VARIABLES:

Static:

```
int mult
int table[]
int tablen
static char mcolor[]
static int mx[]
static int my[]
```

Local:

double	d	random number
int	dx[]	horizontal distance moved
int	dy[]	vertical distance moved
int	i	general purpose index
int	j	general purpose index
int	stopm	
int	stops	
int	timeoff	time period star is off
int	timeon	time period star is on
struct	startype *starp	

Returns:

Functions called:

485

FUNCTION DESCRIPTION

Name: **GRAPHS.C**

Synopsis: **ovmain(graphs)**

Description: This routine graphs pause times

VARIABLES:

See Next Page

Returns:

Functions called: acgraph, ioctlsa, keyin, printf, ptgraph, window

FUNCTION DESCRIPTION

Name: <u>GRAPHS.C</u>

Description:

VARIABLES:

```
Static:
     struct trlvals {
     struct              trial     *trlp
     int                 *valsp
     }
     struct mapdats {
     int                 value
     int                 *valsp
     char                mrk
     char                *linep
     }
     typedef struct lvldscs {
     char                *title
     char                levelc
     }
     char                asmtdsc[]
     struct              lvldscs   dsclvla[]
     char                baseln[]
     char                *head1
     char                *head2
     char                *head3
     char                *heada
     char                *headb
     struct              assessment_r *bufcar
     struct              assessment_r bufca
Local:
     char c              prompt response
```

487

FUNCTION DESCRIPTION

Name:

INITIAL.C

Synopsis:

```
init(lvl)
int lvl    level 1..6
```

Description:

initializes list of letters

VARIABLES:

Global:

char	lrbl[]	letters held by lower row boxes
char	ltr[]	letters to appear below lower row
char	urbl[]	letters held by upper row boxes
int	crlen	length of # of elements in
int	fflen	length of list that holds len# of
int	listlen	total # of elements in an assessment

Local:

struct rsel{	
char	flag;<-- flag='1' for selection-'2'for shuffle
char	letter; <-- letter= selected char.
char	*xclset; excluded set of characters
}	
char c	receives next letter
int j	general purpose integer
int k	general purpose index
struct rsel pass	parameters for randsel

Returns:

Functions called:

ran, randsel

488

FUNCTION DESCRIPTION

Name: INITSTK.C

Synopsis: init_stik()

Description: Initializes joystick

VARIABLES:

Global:

int	pba
int	xscale
int	yscale

Local:

char c	receive character
double d	random number
int x	horizontal character position
int y	vertical line position

Returns:

Functions called: ioctlsa, joypos, printf, ran

459

FUNCTION DESCRIPTION

Name:
INITPEN.C

Synopsis:
initpen()

Description:
Initializes pseudo random number table based on current clock reading

VARIABLES:

Global:

int	pba
int	xscale
int	yscale

Local:

char c	block character
double d	random number
int i	receives intvls for calculating elapsed time

Returns:

Functions called:
intvls, ioctlsa, ran

FUNCTION DESCRIPTION

Name:
JYSTK.C

Synopsis:
jystk()

Description:
Displays a picture of a joystick

VARIABLES:

Static:

char	*icon[]
char	clmrw[][]

Local:

ANSI ans	parameters for screen display
double rand	random number
int clm	current column
int clmlmt	number of columns
int i	receives intvls for calculating elapsed time
int row	current row
int rowlmt	number of rows
int startx	start column
int x1	
int xadj	column width
int y1	
struct	grphcss grphcsl

Returns:

Functions called:
ansiscr, cursdsp, plticon, wait

491

FUNCTION DESCRIPTION

Name:
KEEPGOING.C

Synopsis:
keepgoing(t)
t flag indicates if it is after training sec

Description:
function that asks if user wants to continue

VARIABLES:

Global:

int	loop	loop number 1-4
int	meet_accuracy	
int	set	training set asmt,ff,cr,id,ch
int	train	

Local:

c	prompt response
c1	prompt response
missing	flag indicating correct disk is missing

Returns:

Functions called:
access, getch, ioctlsa, printf, rstrtwin

492

FUNCTION DESCRIPTION

Name:

LOADWRD.C

Synopsis:

```
loadwrd(stringp)
char *stringp  pointer to voacbulary
```

Description:

Loads wocabulary of words

VARIABLES:

Static:

int	lodadr
int	objadr

Local:

char ldinfo[]	header information
int fd	file description number
int n	bytes read
unsigned objlng	length of object for vocabulary

Returns:

Functions called:

close, getch, malloc, open, printf, read

4?3

FUNCTION DESCRIPTION

Name: LOBOXES.C

Synopsis:

```
loboxes(p,n,first,last)
char p;    row position (high or low)
int first  first box to draw
int last   last box to draw
int n      number of boxes
```

Description: Draws display and selection boxes

VARIABLES:

Local:

```
int i        index for box position
int x        horizontal box position
int y        vertical line position
```

Returns:

Functions called: lobox

494

FUNCTION DESCRIPTION

Name: L3B.C

Synopsis:

```
movandshow(n,which,chr)
char *chr
int n      number of boxes
int which box number to color
```

Description:

Color box selected, show a letter and return box to previous color

Returns:

Functions called: set_box_clr, show_ltr, wait

495

FUNCTION DESCRIPTION

Name: <u>**OPSCRN30.C**</u>

Synopsis: opscrn30()

manipulate screen 30 to locate a student

Description:

VARIABLES:

```
     Global:
          struct              student   cs
          struct              student   *csp
          int                 currs      current student record #
          int                 sfd        student file descriptor
     Static:
          static char *blank
     Local:
          char c              prompt response
          int flag            indicate student selected
```

Returns:

Functions called: cutsp, find_name, get_data, ioctlsa, keyin, lseek, modify, printf, read, review, showname

4:6

FUNCTION DESCRIPTION

Name: MUS.C

Synopsis:

Description: Plays song and displays scale as a reward

VARIABLES:

Static:

char	*songp[] pointers to filename
char	notemsg[] note shpae
char	readonly[] read only attribute
int char	songfil[][] set of file names
int	count number of bytes in song file
int	loaded flag indicating scale has bee drawn
int	x horizontal box position
struct	display dsply[] displayed notes
struct	filenote *fnote pointer to note description

Returns:

Functions called: cursdsp, mode, palette, sheet, song, wait

FUNCTION DESCRIPTION

Name: L3B.C

Synopsis:

```
outlinandshow(n,which)
int n       number of boxes
int which box number to outline
```

Description: Display a letter in high and low row and draw connection between boxes

Returns:

Functions called: outlinebox, show_ltr

FUNCTION DESCRIPTION

Name: PLT.C

Synopsis:

```
plt(color,x,y)
int color color code for star
int x     horizontal character position
int y     vertical line position
```

Description: Plot a star at the specified location

VARIABLES:

Local:

struct grphcss grphcsl   parameters for graphics module

Returns:

Functions called: grphcs

FUNCTION DESCRIPTION

Name: QUESTION.C

Synopsis: question()

Description: sound three bells

Returns:

Functions called: ioctlsa, printf

FUNCTION DESCRIPTION

Name: **READY.C**

Synopsis: **ready()**

Description: display the ready image and moving arrow

VARIABLES:

Static:

char arrowhd	arrow head shape
char arwshft	arrow start shape
char readymsg[]	word "ready"

Local:

ANSI	ans parameters for screen display
char *arrowhdp	pointer to arrowhead shape
char *arwshftp	pointer to arrowhead shapeint
int i	idle counter
int x1	horizontal character position
int y1	vertical line position

Returns:

Functions called: ansiscr, cursdsp, pltchr , pltmsg, wait

501

FUNCTION DESCRIPTION

Name:

REVLIST.C

Synopsis:

revlist()

Description:

review student file, can get a hard copy of student list

VARIABLES:

```
Global:
     struct              student    cs
     int                 nstud      # of students
     int                 sfd        student file descriptor
Static:
     int more than26 assume less than 26 student in file
Local:
     char c              prompt response
```

Returns:

Functions called:

getch, ioctlsa, list, lseek, ops13a2, printf, putchar, read

502

FUNCTION DESCRIPTION

Name:
RULE.C

Synopsis:
rule()

Description: display the rule image and blinking stars

VARIABLES:

Static:

```
struct startype{
int            color;
int             x,y;
int            on;
int            cnt,timeon,timeoff;
char           *objp;
}
char           rulemsg[]            word rule
char           staro                star shape
struct         startype  stars[]    set of star locations
```

Local:

```
ANSI           ans parameters for screen display
char           *starp pointer to star shape
int            i odle counter
int            j star index
int            x1 horizontal character position
int            y1 vertical line position
```

Returns:

Functions called: ansiscr, blnkstars, cursdsp, pltmsg, wait

503

FUNCTION DESCRIPTION

Name: PREAT.C

Synopsis: selstrt()

Description: select a starting point for training

VARIABLES:

Global:

int	clvl	current level
struct	personal_r cp	
int	loop	loop number 1-4
int	set	training set asmt,ff,cr,id,ch

Static:

strtlv[0]

Local:

char	c prompt response
int	curs cursor index
int	nsel maximum number of selections

Returns:

Functions called: cursdsp, ioctlsa, keyin, printf

FUNCTION DESCRIPTION

Name: PREAT.C

Synopsis: selstrtp()

Description: Selelct a prepost starting level

VARIABLES:

```
Global:
     int                 clvl      current level
     struct              personal_r op
Static:
     strtlv[]
Local:
     char c              prompt response
     int curs            cursor index
     int nsel            maximum number of selections
```

Returns:

Functions called: cursdsp, ioctlsa, keyin, printf

505

FUNCTION DESCRIPTION

Name:

LOBOXES.C

Synopsis:

```
set_blink(position,num,which)
char position  row position (high or low)
int num        number of boxes
int which      box to set
```

Description:

Set blink attritubute for box

VARIABLES:

Local:

int	x horizontal character position
int	y vertical line position

Returns:

Functions called:

lobox

FUNCTION DESCRIPTION

Name: <u>SHOWROW.C</u>

Synopsis:

```
show_row(n,rol)
char *rol      pointer to indicate string of characters
int n number of characters
```

Description: Displays the row of characters at the bottom of the screen

VARIABLES:

Local:

ANSI	ans
char str[]	row of characters
int first	first character position for row
int x	horizontal character position
int y	vertical line position

Returns:

Functions called: anisor, putchar

507

FUNCTION DESCRIPTION

Name:

SHRINK.C

Synopsis:

```
ovmain(sh, rink,name,n,nff,accum,accterm,ptcum)
char *name
int      accterm      percent accuracy of ff
int      accum        percent accuracy of cum
int      n            total number of items
int      nff          number of fast finish items
int      ptcum        pause time cummultive
```

Description:

provide interpretive remarks

VARIABLES:

Local:

```
char  ans
int first          first fast finish location
int hiaccterm      flag indicating passing fras finish
int hiaccum        flag indicating passing cumulative rehearsal
int hiptcum        flag indicating passing pause time
int i              general purpose index
int len            string index
int pascum         passing cumulative rehearsal score
int paspt          passing pause time score
int pasterm        passing fast finish score
int print          flag indicating hard copy
int x              general purpose index
```

Returns:

Functions called:

getch, ioctlsa, puts, s000, s001, s010, s011, s100, s101, s110, s111, sprintf, strcpy, strlen, window

508

FUNCTION DESCRIPTION

Name: CHLP1.C

Synopsis:
simpler(start,stop,n)

int n	number of boxes
int start	starting box number
int stop	ending box number

Description:
Color conection between high and low rows for a series of boxes

VARIABLES:

Local:

int i	general purpose index

Returns:

Functions called: outlinebox, prin, rstrtwin, wait

509

FUNCTION DESCRIPTION

Name:

GAMEINTR.C

Synopsis:

```
speak(string)
char *string          pointer for string to be spoken
```

Description:

VARIABLES:

```
    Global:
        curpos
        int                 VOICE
    Local:
        ANSI                ans parameters for screen display
        int i               general purpose index
        struct              grphcss grphcsl parameters for graphics module
```

Returns:

Functions called:

ansiscr, prin

FUNCTION DESCRIPTION

Name: TABLE.C

Synopsis:

```
ovmain(table,len,acc,pau)
int *acc        pointer to accuracy scores
int *pau        pointer to pause times
int len         number of entries
```

Description: This routine prints table formatted assessment record

VARIABLES:

See Next Page

Returns:

Functions called: decide_patn, fclose, fopen, fputs, getch, getlevel, ioctlsa, keyin, lseek, omeg, printf, read, sprintf, window

511

FUNCTION DESCRIPTION

Name: TABLE.C

Description:

VARIABLES:

```
Global:
     bufacc[]
     bufcar
     struct            student    cs
     int               dfd        data file descriptor
Static:
     static char *head1
     static char *head2
     static char *head3
     static char *heada
     static char *headb
     struct *bufcar
     struct assessment_r bufca
Local:
     FILE *pf          pointer to printer file
     char c            prompt response
     char c1           prompt response
     char c2           prompt response
     char level        student level
     char str[]        print line
     float omega2      omega squared rating
     int pfi
     int bufacc[]      assessment data
     int hard          hard copy flag
     int i             general purpose index
     int j             general purpose index
     int k             general purpose index
     int l             general purpose index
     init length       elements per trial
     int bufpau[]      pause time data
```

512

FUNCTION DESCRIPTION

Name: VMENU.C

Synopsis:

```
ovmain(vmenu,dd)
struct databuf *dd  pointer to accuracy table
struct databuf{
int len         entry length
int ffnum       fast finish length
int ffacc       fast finish accuracy score
int  cracc      cumulative rehearsal accuracy score
int  pau[]      pause times
int  acc[]      accuracy scores
}
```

Description: This routine is a menu for viewdata routine

VARIABLES:

Global:

```
struct            assessment_r *car
struct            student   cs
int               dfd       data file descriptor
int               fflen     length of list that holds len# of
int               listlen   total # of elements in an
                            assessment
```

Local:

```
char c            prompt response
char c1           prompt response
int craccbuf[]    cululative rehearsal accuracy data
int ffaccbuf[]    fast finish accuracy data
int i             general purpose index
int j             general purpose index
int seeklvl       level of data to search for
```

Returns: return (c)   menu selection

Functions called: decide_patn, getch, ioctlsa, lseek, ,printf, putchar, read, window

513

FUNCTION DESCRIPTION

Name: **MENU.C**

Synopsis: **warning()**

Description: Supplies warning messages, collects overiding responses and deletes the student information

VARIABLES:

Global:

struct	student	cs
int	maxstud	max students in file
int	nstud	# of students
int	sfd	student file descriptor

Local:

char c	prompt response	

Returns:

Functions called: dbcreat, getch, ioctls, lseek, printf, putchar, read

514

FUNCTION DESCRIPTION

Name:	WINDOW.C
Synopsis:	wipe()
Description:	Clears a display window for subsequent text
Returns:	
Functions called:	iootlsa, window

515

FUNCTION DESCRIPTION

Name:

GRAPHS.C

Synopsis:

acgraph()

Description:

VARIABLES:

See Next Page

Returns:

Functions called: decide_patn, fclose, fopen, fputs, getch, ioctlsa, keyin, lseek, mapdatf, printf, read, sellvl, setmem, sprintf

516

FUNCTION DESCRIPTION

Name:

GRAPHS.C

Description:

VARIABLES:

Global:

bufca		
char	bufdate[7]	buffer to get date
buftrl		
int	dfd	data file descriptor
dsclvla[]		
line[]		
mapdat		
seeklvl		
setscp		
trval		

Local:

FILE	*pf	pointer to printer file
char	*setdscp	pointer to set descriptions
char	buf[]	edited date
char	hlin	horizontal line charcater
char	n[]	
char	vlin	vertical line character
char	c	prompt response
char	c1	prompt response
char	c2	prompt response
char	level	student level
char	line[]	print line
char	str[]	print line
int	pfi	printer file return code
int	vals[]	
int	bufacc[]	accuracy scores
int	bufpau[]	pause times
int	l	
int	length	
int	seeklvl	level to seek
int	t	trial number
int	data	
int	hard	flag indicating hard copy
int	i	general purpose index
int	j	general purpose index
int	k	general purpose index
int	lines	line count
struct mapdats	mapdat	data mapped to vertical scale
struct trial	buftrl	trials
struct trlvals	trlval	trial values

FUNCTION DESCRIPTION

Name:

ACTLTR.C

Synopsis:

```
act_box(n,i,c,f)
char    c    which row 'h' or 'l'
int     f    first time flag
int     i    box to activate
int     n    number of boxes
```

Description:

This function activates the i-th box of c row and returns the time inincrements of 50 msec if the time is less than 30 sec else returns -1.

VARIABLES:

Global:

```
stikss.xmax = n;
```

Local:

```
int     t     elasped time
int     x     horizontal character position
int     y     vertical line position
```

Returns:

return(-1)	over 30 seconds elasped time
return(t)	elasped time

Functions called:

ansiscr, lightpen, stik

518

FUNCTION DESCRIPTION

Name:

GAME2.C

Synopsis:

```
blnkstar(starp)
struct    startype *starp     pointer to set of star desoriptions
```

Description:

Review status of stars erasing those that are due to go off and replotting those that are due to go on

VARIABLES:

Global:

```
char        *strp     pointer to start description
char        strl      star shape
```

Returns:

Functions called: pltchr

FUNCTION DESCRIPTION

Name: CURSDSP.C

Synopsis:

```
cursdsp(x)
int  x     type of cursor to display
```

Description: Display a cursor

VARIABLES:

Local

```
struct            pcdosbs {
int               func;
int               ax;
int               bx;
int               cx;
int               dx;
};
struct pcdosbs    pcdosbl   parameters for PC-DOS interface
```

Returns:

Functions called: pcdosb

FUNCTION DESCRIPTION

Name:

DBCREAT.C

Synopsis:

```
dbcreat(ns)
int  ns        number of students
```

Description:

Create a new student data base

VARIABLES:

```
Global:
    struct          assessment_r *car
    struct          student   *csp
    int             dfd       data file descriptor
    int             sfd       student file descriptor
Local:
    int             i         index for student index file
    int             j         index for student assessment data
```

Returns:

Functions called:

close, creat, write

FUNCTION DESCRIPTION

Name:

GAME2.C

Synopsis:

```
drawstr(strg,x,y)
char *strg      pointer to string to be drawn
int x           horizontal line position
int y           vertical character position
```

Description:

Plot a star

VARIABLES:

Local:

struct grphcss grphcsl

Returns:

Functions called: grphcs

FUNCTION DESCRIPTION

Name:

FINDNAME.C

Synopsis:

find_name(n)
int n student number

Description:

to retrieve currs rec, prev rec, and next rec

VARIABLES:

Global:

struct	student	bufcs	buffer cs
struct	student	bufcs1	

Returns:

Functions called:

getns, strncpy

523

FUNCTION DESCRIPTION

Name:

GETDATA.C

Synopsis:

get_data()

Description:

To get necessary information of a new student

VARIABLES:

Global:

struct	personal_r	cp
struct	student	cs
int	dfd	data file descriptor
int	loglen	length of log structure
struct	log	slog

Static:

static	char	*blank

Local:

char	b_date[]	date buffer
char	c	prompt response
int	flag	student inserted ok
int	i	general purpose index
int	l	level number
int	ok	date valid flag

Returns:

flag

Functions called:

cutsp, ins_std, ioctlsa, keyin, lseek, printf, putchar, setmem, write

FUNCTION DESCRIPTION

Name:

GETLEVEL.C

Synopsis:

```
char getlevel(1)
int 1          horizontal character position
```

Description:

VARIABLES:

Local:

```
char            level
```

Returns:

Functions called: level

525

FUNCTION DESCRIPTION

Name:

REVLIST.C

Synopsis:

```
list(i)
int i       horizontal character position
```

Description:

List the remainder of the student file

VARIABLES:

Global:

struct	student	*osp
int	nstud	# of students
int	sfd	student file descriptor

Local:

int	x	horizontal character position
int	y	vertical line position

Returns:

Functions called:

ioctlsa, lseek, printf, putchar, read

526

FUNCTION DESCRIPTION

Name:

GAME2.C

Synopsis:

```
loadrocket(x1,roobuf)
char      roobuf[]
int       x1    horizontal character position
```

Description:

Plot a rocket at its base location

VARIABLES:

Local:

int	i	general purpose index
int	x	horizontal character position
int	x2	
int	y	vertical line position

Returns:

Functions called: pltrkt

527

FUNCTION DESCRIPTION

Name:

OUTLINEB.C

Synopsis:

```
outlinebox(n,k)
int       k     box to outline
int       n     number of boxes
```

Description:

This routine connects the k-th box of the upper row with the k-th box of the lower row of n boxes.

VARIABLES:

Local:

ANSI	scrndat	parameters for screen manulipation
char	block	block shape
int	x	horizontal character position
int	y	vertical line position

Returns:

Functions called: ansiscr, putchar, dissolvebox

528

FUNCTION DESCRIPTION

Name:

OMEG.C

Synopsis:

```
omeg(n,nff,r)
int        n     list length
int        nff   fast finish length
int        r[]   pause times
```

Description:

Compute the omega square rating

VARIABLES:

```
Static:
     int                idl2[]
     int                idlr[]
Local:
     float              idl[]     ideal values
     float              m         omega squared rating
     float              mserr
     float              mspr
     float              ssspr
     float              sstr
     float              z[]       z - scores
     int                cum       cumulative rehearsal length
     int                i         general purpose index
```

Returns:

m

Functions called:

msp, sssp, sst, zoo

529

FUNCTION DESCRIPTION

Name:

PLTICON.C

Synopsis:

```
pltioon(color,x,y,icon,xmag,ymag)
char *     icon       pointer of shape to be plotted
int        color      color of plotted object
int        x          horizontal character position
int        xmag       magnetude in horizontal direction
int         y         vertical line position
int        ymag       magnetude in vertical direction
```

Description:

Plot an icon

VARIABLES:

Local:

```
struct          grphcss   grphcsl   parameters for graphics
                                    routine
```

Returns:

Functions called: grphcs

FUNCTION DESCRIPTION

Name: PLTMSG.C

Synopsis:

```
pltmsg(color,x,y,chr,xmag,ymag)
char    *chr      message
int     color     message color
int     x         x start position
int     xmag      x character multiple
int     y         y start position
int     ymag      y line multiple
```

Description: Plot a message on the graphics screen

VARIABLES:

Local:

struct	grphcss	grphcsl	parameters for graphics routine

Returns:

Functions called: grphcs

531

FUNCTION DESCRIPTION

Name: PRIN.C

Synopsis:

```
prin(string)
char        *string    pointer for string to speak
```

Description: Print text and send text to speech routine

VARIABLES:

```
    Global:
         int int          SCREEN
         int int          VOICE
    Local:
         char             string1[] print line
         char             c         general purpose character
         int              i         source string index
         int              j         destination string index
```

Returns:

Functions called: printe, printf, toupper

532

FUNCTION DESCRIPTION

Name:

GRAPHS.C

Synopsis:

ptgraph()

Description:

Report a pause time graph

VARIABLES:

See Next Page

Returns:

Functions called:

fopen, getch, ioctlsa, printf

533

FUNCTION DESCRIPTION

Name:

GRAPHS.C

Description:

VARIABLES:

Global:

dsclvla[]

Local:

FILE *pf		
char	*setdscp	pointer to set descriptions
char	buf[	edited date
char	hlin	horizontal line character
char	n[]	
char	vlin	vertical line character
char	c	prompt response
char	c1l	prompt response
char	c2	prompt response
char	level	student level
char	line[]	print line
char	strl[]	print line
int	pfi	printer file return code
int	vals[]	
int	t	trial number
int	data	
int	hard	flag indicating hard copy
int	i	general purpose index
int	j	general purpose index
int	k	general purpose index
int	lines	line count
struct mapdats	mapdat	data mapped to vertical scale
struct trial	values	trlvals trlval

534

FUNCTION DESCRIPTION

Name:

GAME2.C

Synopsis:

See Next page

Description:

Establish a new set of stars

VARIABLES:

Local:

char	*strp	pointer to star shpae
char	strl	star shape
int	i	general purpose index
int	timeon	time star is on
int	times	

Returns:

Functions called: pltchr, shipmov, update

535

FUNCTION DESCRIPTION

Name:

GAME2.C

Synopsis:

```
renew(star,level,shipbuf,hitbuf,rocbuf,shipspd,indev,xscale,yscal
e,time,tim,shiphit,curtime,oldscore,score,nums)
char *nums
char hitbuf[]
char indev
char rocbuf[]
char shipbuf[]
int *curtime
int *oldscore
int *score
int *time
int level
int shiphit
int shipspd
int tim
int xscale
int yscale
struct startype star[]
```

Returns:

Functions called:

536

FUNCTION DESCRIPTION

Name: <u>**REVIEW.C**</u>

Synopsis:

```
review()
char        *out pointer to output type ttle
char        *spd pointer to speed title
char        c    keyboard input
```

Description: to review information of a student, either new or old student

VARIABLES:

Global:

```
struct          personal_r cp
struct          student    cs
```

Returns:

Functions called: ~~getcomm, ioctlsa, modinfo, printf~~

537

FUNCTION DESCRIPTION

Name:	WINDOW.C
Synopsis:	rstrtwin(y) int y horizontal line position
Description:	clear the window defined as line 20 through 24 and set the cursor at yhe selected line y.

VARIABLES:

Local:

ANSI	ans	parameters for screen display

Returns:

Functions called: ansiscr, printf

538

FUNCTION DESCRIPTION

Name: **S000.C**

Synopsis:

```
ovmain(s000,name,first,nff,accum,accterm,print)
char    *name      pointer to student name
int     accterm    fast finish accuracy score
int     accum      cumulative rehearsal score
int     first      list length
int     nff        number of fast finish elements
int     print      flag indicating hard copy
```

Description: provide interpritive remarks

VARIABLES:

Local:

```
int        a        cumulative rehearsal accuracy score
int        b        fast finish accuracy score
int        len      length of string
int        n        fast finish length
int        x        list length
```

Returns:

Functions called: sprintf, strlen

529

FUNCTION DESCRIPTION

Name: S001.C

Synopsis:

```
ovmain(s001,name,first,nff,accum,accterm,print)
char    *name      pointer to student name
int     accterm    fast finish accuracy score
int     accum      cumulative rehearsal score
int     first      list length
int     nff        number of fast finish elements
int     print      flag indicating hard copy
```

Description: provide interpritive remarks

VARIABLES:

Local:

```
int        a        cumulative rehearsal accuracy score
int        b        fast finish accuracy score
int        len      length of string
int        n        fast finish length
int        x        list length
```

Returns:

Functions called: sprintf, strlen

510

FUNCTION DESCRIPTION

Name: S010.C

Synopsis: ovmain(s010,name,first,nff,accum,accterm,print)

char	*name	pointer to student name
int	accterm	fast finish accuracy score
int	accum	cumulative rehearsal score
int	first	list length
int	nff	number of fast finish elements
int	print	flag indicating hard copy

Description: provide interpritive remarks

VARIABLES:

Local:

int	a	cumulative rehearsal accuracy score
int	b	fast finish accuracy score
int	len	length of string
int	n	fast finish length
int	x	list length

Returns:

sprintf, strlen

Functions called:

511

FUNCTION DESCRIPTION

Name:

S011.C

Synopsis:

```
ovmain(s011,name,first,nff,accum,accterm,print)
char    *name      pointer to student name
int     accterm    fast finish accuracy score
int     accum      cumulative rehearsal score
int     first      list length
int     nff        number of fast finish elements
int     print      flag indicating hard copy
```

Description:

provide interpritive remarks

VARIABLES:

Local:

```
int          a          cumulative rehearsal accuracy score
int          b          fast finish accuracy score
int          len        length of string
int          n          fast finish length
int          x          list length
```

Returns:

Functions called: sprintf, strlen

FUNCTION DESCRIPTION

Name: S100.C

Synopsis:

```
ovmain(s100,name,first,nff,accum,accterm,print)
char    *name      pointer to student name
int     accterm    fast finish accuracy score
int     accum      cumulative rehearsal score
int     first      list length
int     nff        number of fast finish elements
int     print      flag indicating hard copy
```

Description: provide interpritive remarks

VARIABLES:

Local:

int	a	cumulative rehearsal accuracy score
int	b	fast finish accuracy score
int	len	length of string
int	n	fast finish length
int	x	list length

Returns:

Functions called: sprintf, strlen

FUNCTION DESCRIPTION

Name: S101.C

Synopsis:

```
ovmain(s101,name,first,nff,accum,accterm,print)
char     *name       pointer to student name
int      accterm     fast finish accuracy score
int      accum       cumulative rehearsal score
int      first       list length
int      nff         number of fast finish elements
int      print       flag indicating hard copy
```

Description: provide interpritive remarks

VARIABLES:

Local:

```
int      a      cumulative rehearsal accuracy score
int      b      fast finish accuracy score
int      len    length of string
int      n      fast finish length
int      x      list length
```

Returns:

Functions called: sprintf, strlen

511

FUNCTION DESCRIPTION

Name:

S110.C

Synopsis:

```
ovmain(s110,name,first,nff,accum,accterm,print)
char    *name      pointer to student name
int     accterm    fast finish accuracy score
int     accum      cumulative rehearsal score
int     first      list length
int     nff        number of fast finish elements
int     print      flag indicating hard copy
```

Description:

provide interpritive remarks

VARIABLES:

Local:

```
int     a      cumulative rehearsal accuracy score
int     b      fast finish accuracy score
int     len    length of string
int     n      fast finish length
int     x      list length
```

Returns:

Functions called:

sprintf, strlen

545

FUNCTION DESCRIPTION

Name: S111.C

Synopsis:

```
ovmain(s111,name,first,nff,accum,accterm,print)
char    *name      pointer to student name
int     accterm    fast finish accuracy score
int     accum      cumulative rehearsal score
int     first      list length
int     nff        number of fast finish elements
int     print      flag indicating hard copy
```

Description:

provide interpritive remarks

VARIABLES:

Local:

```
int     a      cumulative rehearsal accuracy score
int     b      fast finish accuracy score
int     len    length of string
int     n      fast finish length
int     x      list length
```

Returns:

Functions called: sprintf, strlen

546

FUNCTION DESCRIPTION

Name:

LOBOXES.C

Synopsis:

```
set_box_clr(pos,num,which,forgrnd)
char        pos         row position (high or low)
int         forgrnd     foreground color
int         num         number of boxes
int         which       which box to color
```

Description:

VARIABLES:

Local:

ANSI	ans	parameters for screen display
int	UNDSCORE	character code
int	VERLINE	charactercode
int	i	general purpose index
int	x	horizontal character position
int	y	vertical line position
int	ly	y + 2

Returns:

Functions called:

ansiscr, putchar

517

FUNCTION DESCRIPTION

Name:
MUS.C

Synopsis:
sheet()

Description:
Display a music scale

VARIABLES:

Local:
struct drw lines lines for scale

Returns:

Functions called:
drw

518

FUNCTION DESCRIPTION

Name:

SHOWLTR.C

Synopsis:

show_ltr(num,chr,which,ypos,time)		
char	chr	character to display
char	ypos	row position (high or low)
int	num	number of elements
int	time	time to display in thenth of a second
int	which	which box to display

Description:

This function shows a letter in k-th box (k<=n) for an amount of time specified by time for argument 'time', 0 denotes no time specified by 'time' limit unless screen is wiped out

1 denotes 0.1 sec
5 denotes 0.5 sec
10 denotes 1.0 sec
50 denote 5.0 sec
etc.

VARIABLES:

Statics:

int	t1	
int	t2	
static	char blank	

Local:

ANSI	ans	parameters for screen display
int	intvl	elasped time
int	x	horizontal character position
int	y	vertical line position

Returns:

return(intvl)

Functions called:

ansiscr, intvls, putchar

519

FUNCTION DESCRIPTION

Name:

FINDNAME.C

Synopsis:

showname()

Description:

Display a students name

VARIABLES:

Global:

struct	student	bufcs	buffer cs
struct	student	bufcs1	
struct	student	bufcs2	

Returns:

Functions called: ioctlsa, printf

550

FUNCTION DESCRIPTION

Name:

MUS.C

Synopsis:

```
song(sp)
int  sp
```

Description:

Play a song

VARIABLES:

Local:

int	z	note index
struct	display	disply
unsigned	s	song selection

Returns:

Functions called: free, getsong, loadscr, ran savscr, tone

551

FUNCTION DESCRIPTION

Name:

GAME2.C

Synopsis:

```
starhit(i,starp)
int i
struct startype *starp
```

Description:

VARIABLES:

Local:

char	*strp	pointer to star shape
char	strl	star shape

Returns:

Functions called: pltchr, whistle

552

FUNCTION DESCRIPTION

Name:

LOBOXES.C

Synopsis:

```
stop_blink(pos,num,which)
char      pos       row position (high or low)
int       num       number of boxes
int       which     box number to reset blink attribute
```

Description:

Turn off blink attribute for box

VARIABLES:

Local:

int	y	horizontal character position
int	x	vertical line position

Returns:

Functions called:

lobox

553

FUNCTION DESCRIPTION

Name: **WAIT.C**

Synopsis:

```
wait(time)
int         time        time to wait in tenths of a second
```

Description: Wait a requested interval of time

VARIABLES:

Local:

unsigned	t1	intervals from which to calculate time
unsigned	t2	intervals from which to calculate time
unsigned	t3	intervals from which to calculate time

Returns:

Functions called: intvls

554

FUNCTION DESCRIPTION

Name:

WINDOW.C

Synopsis:

```
window(top,bottom,left,width)
int        bottom     bottom line number
int        left left most character position
int        top   top line number
int        width      width in characters
```

Description:

Sets the text window.

VARIABLES:

Global:

wndbtm
wndlft
wndtop
wndwdth

Returns:

Functions called:

555

FUNCTION DESCRIPTION

Name:

OUTLINER.C

Synopsis:

```
dissolvebox(n,k)
int        k    box to connect
int        n    number of boxes
```

Description:

This routine disconnects the k-th box of the upper row from the k-th box of the lower row of n boxes.

VARIABLES:

Local:

ANSI	crndat	parameters for screen display
int	x	horizontal column position
int	y	vertical line position

Returns:

Functions called:

ansiscr, loboxes, putchar

FUNCTION DESCRIPTION

Name:

MUS.C

Synopsis:

```
drw(linep)
struct    drw *linep    pointer to parameters for line drawing
                        position
```

Description:

Draw a line for a musical staff

VARIABLES:

Local:

int	a	
int	color_cd	line color
int	hline_ln	horizontal line shape
int	hor_x	horizontal line starting character position
int	hor_y	horizontal line vertical line position
int	ver_x	vertical line character position
int	ver_y	vertical line starting line position
int	vline_ln	vertical line shape

Returns:

Functions called:

color, line

557

FUNCTION DESCRIPTION

Name:
GETCOMM.C

Synopsis:

```
getcomm(x,y)
int         x    horizontal character position
int         y    vertical line position
```

Description:
get a command character together with a RTN, return commamd char

VARIABLES:

Local:

char	c[]	receives a string of characters from keyboard input

Returns:
return(c[0])   return first keystroke

Functions called:
ioctlsa, keyin

558

FUNCTION DESCRIPTION

Name:

FINDNAME.C

Synopsis:

```
getns(n,bufcsxp)  get n-th stud rec
int       n    student number
struct    student *bufcsxp    pointer to student record
```

Description:

Locate a students name in the student index

VARIABLES:

Global:

```
bufcsxp
int              sfd        / student file descriptor
```

Returns:

```
return(-1); no stud rec#
return(read(sfd,(char *)bufcsxp,26));
```

Functions called:

lseek, read

FUNCTION DESCRIPTION

Name:

MUS.C

Synopsis:

```
getsong( songfilep )
char *songfilep
```

Description:

VARIABLES:

Local:

```
int                     retrn
```

Returns:

Functions called:

malloc, notes, szntfil

FUNCTION DESCRIPTION

Name:

INSSTD.C

Synopsis:

```
ins_std(name,data)
char          *name         pointer to student name
struct personal_r *data     pointer to student personal data
```

Description:

Insert a student in the student data base

VARIABLES:

Global:

struct	student	*bufcsp
struct	student	cs
int	currd	current data record #
int	currs	current student record #
int	maxstud	max students in file
int	nstud	# of students
int	sfd	student file descriptor

Local:

char	c	general usage character
int	n	

Returns: return(-1) /* duplicate keys */ return(-2) /* no space left in sdb */ return(currs) pointer to current student

Functions called:

found, lseek, read, write

561

FUNCTION DESCRIPTION

Name:

LIGHTPEN.C

Synopsis:

lightpen(stiksp)
struct stiks *stiksp parameters for lightpen routine

Description:

requests a lightpen reading

VARIABLES:

Global:

int	int	pba
int	maxtime	voice or text output
stiksp		

Local:

int	i	elasped time
int	t	
int	t1	time of entry
int	t2	time of activation
int	x	horizontal character position
int	y	vertical line position

Returns: return(-1)
return(t2 - t1)

Functions called:

chirp, intvls, penpos, alarm

562

FUNCTION DESCRIPTION

Name: LOBOXES.C

Synopsis:

lobox(x,y,blink)		
char	blink	flag indicating blink attribute
int	x	horizontal character position
int	y	vertical line position

Description: Draw a box for character display

VARIABLES:

Local:

ANSI	ans	parameters for screen display
int	UNDSCORE	horizontal line shape
int	VERLINE	vertical line shape
int	i	horizontal index
int	y1	vertical index

Returns:

Functions called: ansiscr, putchar

FUNCTION DESCRIPTION

Name:
GRAPHS.C

Synopsis:
```
mapdatf(mapdatp)
struct    mapdats   *mapdatp  parameters for mapdat routine
```

Description:
Map the data to vertically scaled positions

VARIABLES:

Local:

int	i	horizontal index
int	scale	scale to appl to y value

Returns:

Functions called:

FUNCTION DESCRIPTION

Name: REVIEW.C

Synopsis: modinfo()

Description:
Modify student personal information

VARIABLES:

Global:

struct	personal_r cp		
struct	student	cs	
int	dfd	data file descriptor	
int	sfd	student file descriptor	
sptr			

Local:

char	*out	pointer to output typetitle	
char	*spd	pointer to speed title	
char	c	input character	
char	input[]	input device name	
char	output[]	output type name	
char	sex[]	sex title	
char	speed[]	speed title	
int	i	elasped time	
struct	personal_r *pp		pointer to personal data
struct	personal_r p		personal data
struct	student *sptr		pointer to student data
struct	student s		student data

Returns:

Functions called: cutsp, found, ioctlsa, lseek, modify, printf, strncpy, write

FUNCTION DESCRIPTION

Name:

OMEG.C

Synopsis:

```
float   msp(n,z,ideal)
float   ideal[]   ideal pause times
float   z[]       z scores
int     n         length of string
```

Description:

Compute mean square

VARIABLES:

Local:

```
float       a
float       b
float       c
float       d
float       e
float       m
int         i
```

Returns:

Functions called:

FUNCTION DESCRIPTION

Name:

MUS.C

Synopsis:

```
savscr(y)
int        y    vertical line position
```

Description:

VARIABLES:

Global:

```
dsply[]
```

Local:

```
int          a
int          al
int          b
int          i
```

Returns:

Functions called:

```
pltchr
```

FUNCTION DESCRIPTION

Name:

SELLVL.C

Synopsis:

```
sellvl(levelp)
int         *levelp   pointer to level number
```

Description:

Get student assessment data for a requested level

VARIABLES:

Local:

Char   c

Returns:

Functions called: decide_patn, fclose, fputs, ioctlsa, keyin, lseek, mapcapf, printf, read, setmem, sprintf

FUNCTION DESCRIPTION

Name:

GAME.C

Synopsis:

```
shipmov(bx,by,shipbuf,hitbuf,nums)
char hitbuf[]
char shipbuf[]
int bx[]
int by[]
int nums[10]   string of digits
```

Description:

Display a ship moving across the screen

VARIABLES:

Local:

See Next Page

Returns:

Functions called: hit, pltrkt, pltshp

FUNCTION DESCRIPTION

Name:

GAME.C

Description:

VARIABLES:

Local:

```
char *ship
char            lship
int             length
int             nt
int             oship
int
int             r
int             umcaptives
int             x
int             y
int             ocket
int             x
int             x1
int             x2
int
int             1
int             x2
int             y
int             y1
int             y2
int
int             1
int             2
```

570

FUNCTION DESCRIPTION

Name:

OMEG.C

Synopsis:

```
float     sssp(n,z,ideal)
float     z[]         z scores
float     ideal[]     ideal pause times
int       n           length of string
```

Description:

VARIABLES:

Local:

```
float     a
float     b
float     c
float     d
float     e
float     m
int       i
```

Returns:

m

Functions called:

571

FUNCTION DESCRIPTION

Name:

OMEG.C

Synopsis:

```
float    sst(n,z,ideal)
float    ideal[]   ideal pause times
float    z[]       z scores
int      n    length of string
```

Description:

VARIABLES:

Local:

```
float    a
float    b
float    c
float    d
float    e
float    m
int      i
```

Returns: m

Functions called:

572

FUNCTION DESCRIPTION

Name:

STIK.C

Synopsis:

```
stik(stiksp)
struct    stiks     *stiksp   parameters for stik routine
```

Description:

requests a joystick reading and replaces current block and writes new cursor

VARIABLES:

```
Global:
    int          pba
    int          xscale
Static:
    char         lastx
    struct       curpos     curposs
Local:
    ANSI         ans        parameters for screen display
    char         currentx   current horizontal box number
    int          t1         start time
    int          t2         stop time
    int          x          horizontal character position
    int          y          vertical line position
```

Returns:

```
return(-1)
return(t2 - t1)
```

Functions called:

ansiscr, intvls, joypos, putchar

FUNCTION DESCRIPTION

Name:

TONE.C

Synopsis:

```
tone(freq,time)
int        freq frequency of tone
int        time time in tenths of a second
```

Description:

Sound a tone for a cpecified period of time

VARIABLES:

Local:

```
int             hibyt
int             lobyt
int             port        port number
long            count
long            divisor
long            i
```

Returns:

Functions called:

inportb, outportb, printf

FUNCTION DESCRIPTION

Name:

GAME2.C

Synopsis:

```
update(oldscore,score,nums,x)
char      *nums          string of digits
int       *oldscore      pointer to previous score
int       score          current score
int       x
```

Description:

Erase old scores and draw updated scores

VARIABLES:

Local:

```
int            i        index into string
int            n[]      array of digits from score
int            x1       horizontal character position
int            y        vertical line position
```

Returns:

Functions called:

drawnum, erasenum

575

FUNCTION DESCRIPTION

Name:

WHISTLE.C

Synopsis:

whistle()

Description:

VARIABLES:

Local:

struct	sounds	sound1	parameters for sound routine

Returns:

Functions called:

sound

FUNCTION DESCRIPTION

Name:

ZOO.C

Synopsis:

```
zoo(n,raw,z)
float     z[]       z scores
int       n         number of data points
int       raw[]     raw pause times
```

Description:

compute z scores

VARIABLES:

Local:

```
float          a
float          b
float          c
float          d
int            i
```

Returns:

Functions called:

sqrt

577

FUNCTION DESCRIPTION

Name:

~~ALARM.C~~

Synopsis:

```
alarm(parm1,parm2,parm3)
int  parm1
int  parm2
int  parm3
```

Description:

Sound alanr for time out

VARIABLES:

Local:

struct	sounds	sound1	paramaters for the sound routine

Returns:

Functions called:

sound

FUNCTION DESCRIPTION

Name:

ANSISCR.C

Synopsis:

```
ansiscr(ansip)
ANSI      *ansip    pointer to parameters for screen display
```

Description:

To provide "ioctl" functions using an ansi terminal interface.

VARIABLES:

Local:

FILE	*fp	pointer to file defination
char	c	general usage character
char	clmn	column number
char	line	line number
int	send	

Returns:

Functions called: putc

579

FUNCTION DESCRIPTION

Name:

CUTSP.C

Synopsis:

```
cutsp(strg,n)
char      strg pointer to string
int            length of string
```

Description:

cut unnecessary space(s) or null terminator(s) in a string

VARIABLES:

Local:

```
int            i         index into string
```

Returns:

Functions called:

580

FUNCTION DESCRIPTION

Name:

DECIDE.C

Synopsis:

```
decide_patn(l)
int l;   current level
```

Description:

according to current level decides corresponing circular recall pattern and assigns appropriate values to fflen and crlen

VARIABLES:

Global:

int	crlen	length of # of elements in
int	fflen	length of list that holds len# of
int	listlen	total # of elements in an assessment
int	prep	

Returns:

Functions called:

FUNCTION DESCRIPTION

Name:

GAME.C

Synopsis:

```
drawnum(nums,x,y)
char      *nums     string of digits
int       x         horizontal character position
int       y         vertical line position
```

Description:

Draw numbers on graphics using graphics characters

VARIABLES:

Local:

struct grphcss grphcsl

Returns:

Functions called:

grphcs

552

FUNCTION DESCRIPTION

Name:

GAME2.C

Synopsis:

```
erasenum(nums,x,y)
char    *nums    string of digits
int     x        horizontal character position
int     y        vertical line position
```

Description:

Rewrite graphics numbers in background color

VARIABLES:

Local:

```
struct          grphcss   grphcsl
```

Returns:

Functions called:

grphcs

FUNCTION DESCRIPTION

Name:
FOUND.C

Synopsis:
found(name)
char *name pointer to student name

Description:

VARIABLES:

Global:
int nstud # of students

Returns:
return(-1)

Functions called:
b_search, lseek, read

584

FUNCTION DESCRIPTION

Name:

~~GAME.C~~

Synopsis:

```
hit(x1,y1,hitbuf)
char hitbuf[]
int x1
int y1
```

Description:

Make a sound and display an icon to signify a shop being hit by a rocket

VARIABLES:

Local:

int	x2	adjusted horizontal character position
int	y2	adjusted vertical line position y2

Returns:

Functions called: chirp, plthit

FUNCTION DESCRIPTION

Name:

INTVLS.C

Synopsis:

```
intvls()
```

Description:

Get clock tick reading to compute interval of time in ticks

VARIABLES:

Local:

```
struct          pcdosbs {
int             func;
int             ax;
int             bx;
int             cx;
int             dx;
}
int             i
struct          pcdosbs   pcdosbl
```

Returns:

```
return(i)
```

Functions called:

pcdosb

586

FUNCTION DESCRIPTION

Name:

JOYPOS.C

Synopsis:

```
joypos(x,y)
int     *x
int     *y
```

Description:

Get a joystick reading

VARIABLES:

Local:

int	i	counter of idle loop
int	js	x,y position of joystick

Returns:

Functions called:

joyposr

587

FUNCTION DESCRIPTION

Name:

KEYIN.C

Synopsis:

keyin()

Description:

to echo off inappropriate char and output a beep

VARIABLES:

Local:

char	c	receive keystroke

Returns:

return(-1)
return(c)

Functions called:

getch, putchar

FUNCTION DESCRIPTION

Name:

GRAPHS.C

Synopsis:

```
mapcapf(mapdatp)
struct    mapdats   *mapdatp   pointer to parameters for mapcapf
                               routine
```

Description:

VARIABLES:

Local:

```
int          i
int          scale
```

Returns:

Functions called:

589

FUNCTION DESCRIPTION

Name:

MODIFY.C

Synopsis:

```
modify(n,str,x1,y,x2)
char      *str pointer to string to modify
int       n    length of string
int       x1   horizontal character position
int       x2   horizontal end character position
int       y    vertical line position
```

Description:

to modify a string on a certain line w/i range x1,x2

VARIABLES:

Local:

```
char c
int i
```

Returns:

Functions called:

ioctlsa, keyin, printf, putchar

FUNCTION DESCRIPTION

Name:

PENPOS.C

Synopsis:

```
penpos(x,y)
int  *x
int  *y
```

Description:

VARIABLES:

```
Global:
    int          pba
Local:
    struct       pcdosbs {
    int          func;
    int          ax;
    int          bx;
    int          cx;
    int          dx;
    }
    int          i
    int          t1
    int          t2
    int          xl1
    int          xl2
    int          yl1
    int          yl2
    struct       pcdosbs   pcdosbl
```

Returns:

return(t2 - t1)
return(t2 - t1)

Functions called:

alarm, intvls, pcdosb

FUNCTION DESCRIPTION

Name:

PLTCHR.C

Synopsis:

```
pltchr(color,x,y,chr,xmag,ymag)
char        *chr        character shape to plot
int         color       color of shape
int         x           horizontal character position
int         xmag        x scaling factor
int         y           vertical line position
int         ymag        y scaling factor
```

Description:

Plot a graphics character

VARIABLES:

Local:

struct	grphcss	grphcs	parameters for graphics routine

Returns:

Functions called:

grphcs

FUNCTION DESCRIPTION

Name:

PLTRKT.C

Synopsis:

```
pltrkt(color,x,y)
int        color     color of rocket
int        x    horizontal character position
int        y    vertical line position
```

Description:

Plot a rocket on a graphics screen

VARIABLES:

Static:

char	rkt		

Local:

struct	grphcss	grphcsl	parameters for graphics routine

Returns:

Functions called:

grphcs

FUNCTION DESCRIPTION

Name:

PLTSHP.C

Synopsis:

```
pltshp(color,x,y,shp)
int        color      color of rocket
int        x          horizontal character position
int        y          vertical line position
```

Description:

Plot a shop on the graphics screen

VARIABLES:

Static:

```
char                ship
```

Local:

```
struct              grphcss   grphcsl   parameters for graphics
                                        routine
```

Returns:

Functions called:

grphcs

534

FUNCTION DESCRIPTION

Name:

FOUND.C

Synopsis:

```
b_search(nm,lb,ub)   binary search
char       *nm   pointer to student name
int        lb    left branch
int        ub    right branch
```

Description:

VARIABLES:

```
Global:
    struct          student    *bufcsp
    struct          student    cs
    int             sfd        student file descriptor
Local:
    char            f1[]
    char            f2[]
    char            l1[]
    char            l2[]
    int             i
    int             k
    int             k1
    int             len
    int             mid
```

Returns:

```
return(b_search(nm,mid+1,ub)) return(lb*(-1))
return(mid) /* both full names match */
```

Functions called:

lseek, read, strlen, strncmp, tolower

FUNCTION DESCRIPTION

Name:

CHIRP.C

Synopsis:

```
chirp(parm1,parm2,parm3)
int        parm1
int        parm2
int        parm3
```

Description:

Make a chirping sound

VARIABLES:

Local:

struct	sounds	sound1	parameters for sound routine

Returns:

Functions called:

sound

596

FUNCTION DESCRIPTION

Name:

GETCH.C

Synopsis:

getch()

Description:

replacement for getchar to provide raw input

Returns:

Functions called:

bdos(0x08);

FUNCTION DESCRIPTION

Name:

GRPHCS.ASM

Synopsis:

```
grphcs_
char c    function code
int  x    x-coordinate
int  y    y-coordinate
int  color
int  xlr  x-coordinate lower right
int  ylr  y-coordinate lower right
```

Description:

graphics screen interface

VARIABLES:

Local:

x0	dw	0
y0	dw	0
x1	dw	0
y1	dw	0
x2	dw	0
y2	dw	0
color	dw	0
xmagn	dw	0
ymagn	dw	0

Returns:

Functions called:

598

FUNCTION DESCRIPTION

Name:

JOYPOS.ASM

Synopsis:

joyposr_

Description:

reads JOYSTICK and returns x and y on; the Aztec stack.

VARIABLES:

Global:

int	pba_

Local:

int	yx
int	cnt

Returns:

Functions called:

559

FUNCTION DESCRIPTION

Name:

PCDOSB.ASM

Synopsis:

```
pcdosb_
struct
        int   ax    ax register
        int   bx    bx register
        int   cx    cx register
        int   dx    dx register
```

Description:

"C" to pcdos bios 10 interface

Returns:

Functions called:

600

FUNCTION DESCRIPTION

Name:

SOUND.ASM

Synopsis:

```
sounda_
char funct      function code
```

Description:

square wave sound primitives

VARIABLES:

Local:

```
int        scale1
int        scale2
int        seed
```

Returns:

Functions called:

6'1

FUNCTION DESCRIPTION

Name:

VCEINTF.ASM

Synopsis:

```
vceintf_
struct
     int  funct      function code
     char *string    words to speak
```

Description:

* to assembler function interface using a BASIC function call interface

VARIABLES:

Local:

```
int        scale1
int        scale2
int        seed
```

Returns:

Functions called:

6 2