A COMPUTER SOFTWARE SYSTEM FOR ADAPTIVE ABILITY MEASUREMENT

Louis J. DeWitt

and

David J. Weiss

Research Report 74-1

Psychometric Methods Program
Department of Psychology
University of Minnesota
Minneapolis, MN 55455

January 1974

Prepared under contract No. N00014-67-A-0113-0029
NR No. 150-343, with the Personnel and
Training Research Programs, Psychological Sciences Division
Office of Naval Research

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
Research Report 74-1

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
A Computer Software System for Adaptive Ability Measurement

5. TYPE OF REPORT & PERIOD COVERED
Technical Report

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Louis J. DeWitt and David J. Weiss

8. CONTRACT OR GRANT NUMBER(s)
N00014-67-0113-0029

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Department of Psychology
University of Minnesota
Minneapolis, MInnesota 55455

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
P.E.:61153N PROJ.: RR042-04
T.A.:RR042-04-01
W.U.:NR150-343

11. CONTROLLING OFFICE NAME AND ADDRESS
Personnel and Training Research Programs
Office of Naval Research
Arlington, VA 22217

12. REPORT DATE
January 1974

13. NUMBER OF PAGES
65

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited. Reproduction in whole or in part is permitted for any purpose of the United States Government.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
testing
ability testing
computerized testing
adaptive testing
sequential testing
branched testing
individualized testing
tailored testing
programmed testing
response-contingent testing
automated testing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A system of computer programs designed to control the administration of adaptive ability tests was developed and used for over 2500 hours of ability measurement. The system is capable of administering any combination of two kinds of tests to a given individual without interruption. Each test can be based on one of six different testing strategies and can administer items selected from up to nine different item

(over)

20 (continued)

pools within each strategy. The system is designed to accept either multiple-choice responses or free-response numeric responses. No feedback, feedback after each item, or item-by-item feedback upon completion of testing can be selected differentially for each test and each testee. The requirements of the research design and constraints of the computer system as well as practical considerations are detailed and their role in the design of the system are discussed. Technical requirements of the software system and problems which might arise in a transfer of the software system to another computer system are considered. Some basic concepts of computer programming are developed as an aid to the reader not technically trained in computer concepts.

Contents

A COMPUTER SOFTWARE SYSTEM FOR ADAPTIVE ABILITY MEASUREMENT

Adaptive, computer-controlled, ability test administration has considerable promise for improving ability measurement. Research to date (Weiss and Betz, 1973) indicates that adaptive tests can yield ability estimates which are more precise than those derivable from conventional tests, in addition to providing scores with higher validity. Adaptive tests can achieve reliabilities and validities at least as high as those of conventional ability tests with considerably fewer test items, thus saving time on the part of the testee. Such time savings can be put to use in the measurement of other abilities to supplement information available on each individual.

Computerized ability measurement also has other potential advantages over paper and pencil testing procedures (Weiss and Betz, 1973, pp. 56-57). Among these are the elimination of time limits, thus reducing artifactual speed components in test scores; improved standardization of test administration conditions, thus reducing administrator effects; the possibility of measuring new content and modes of abilities not now measurable with paper and pencil tests; development of tests measuring problem-solving abilities using "within-item" branching as opposed to the more typical "between-item" branching; computerized control of confidence-weighting schemes of response, including techniques for differential weighting of response distractors; use of item response latency data as additional diagnostic (and perhaps predictive) information derivable from ability testing; the _a priori_ control of the degree of precision attached to any ability estimate derived from ability test administration; the tailoring of test difficulty to the individual's ability level, potentially reducing test-taking anxiety and increasing test-taking motivation; and the use of knowledge of results or feedback after each item is answered (with varying ratios of positive to negative reinforcement) potentially increasing test-taking motivation.

Should computerized ability testing prove capable of providing at least some of these potential improvements in measuring individual differences in abilities, operational computerized ability testing systems will be developed in the next decade. Such systems will find their optimal use in institutions which serve populations of wide-ranging ability, and in applications where it is desired to measure a large number of abilities in a minimum amount of testing time. Thus, military placement and selection activities, industrial personnel departments, vocational-technical schools, vocational guidance departments of high schools and colleges, and public and private vocational counseling agencies are likely users of computerized ability tests. In the majority of cases these agencies will simply purchase or lease time-shared computer terminal devices, such as cathode-ray-terminals (CRTs), or hard-copy devices such as medium-speed teletype terminals, for interfaces between the testee and the computer. Computer programs to administer ability tests will probably be maintained by centralized facilities which are accessed by conventional telephones linked to the central time-shared computer. Just as there are currently different testing agencies providing a variety of conventional paper and pencil tests for these same test users, it is likely that different computer software systems will be available for computerized ability testing. These systems will differ not only in the strategies of testing available but also in the item pools available for ability measurement (both nature and quality) and in other system characteristics which allow for user flexibility and control.

Although adaptive ability testing has generated a small amount of research to date, very little of the research has actually used computer-controlled test administration. Yet computerized ability testing raises a host of new questions waiting to be addressed by the psychometric research community. These problems include (Weiss and Betz, 1973, pp. 49-54) comparisons of the almost endless variety of adaptive testing procedures possible under computer-controlled administration; empirical investigation of the characteristics of the various scoring methods within each of the adaptive strategies; new problems and approaches to item pool development for computerized testing; the effects of chance success on the scores derived from adaptive test administration; the variety of termination rules possible in some adaptive strategies; applications of on-line differential weighting procedures; the actual effects on the testee of adaptive test administration, in terms of reducing anxiety and increasing motivation; and the effects on test scores of immediate knowledge of results.

There is likely, then, to be a demand in the near future for computerized ability testing systems for both research and applied testing purposes. The present authors have developed such a system which is operational and which has logged over 2,500 hours of actual testing. This report describes the design of that system and its rationale, so that it might serve as a starting point for others who in the future might wish to develop computerized ability testing systems for research and/or applied uses. It is hoped that future systems will improve upon the present software system so that computerized ability testing will reach its maximum potential.

Requirements for a Computerized Ability Testing System

A minimal system must 1) select a test item from an item pool following a predetermined set of instructions, 2) present the item on a CRT or other display unit, 3) accept a response, 4) check it for validity and compatibility with the instructional sequence, 5) determine whether the response is correct or incorrect, 6) calculate response latency, 7) save the obtained data, and 8) select the next item to be administered in accordance with a specified testing strategy. In addition, previous experience with computerized ability testing and plans guiding the present research dictated a number of features which would be desirable in any computerized testing system, for either research or applied use.

One primary characteristic is operational simplicity in terms of supervision and testee/system interaction. This dictates several important features of the system:

1. The system must provide for monitoring by proctors with little or no computer experience. Thus, the proctor should know simply how to dial the telephone, call the testing program, assign testing conditions to the testee, and handle common hardware (equipment) or software (program) problems as they arise.

2. The system must be self-instructional. Thus, the program must present a series of instructional screens and enforce performance to a criterion on each screen. Should the testee not master the instructions for use of the equipment after a given number of trials, the proctor should be called to assist the testee.

3. Since testees will occasionally wish to change an answer typed to a test item, because of either typing errors or actual changes in their answer to an item, the system must provide some opportunity to change responses, at least before the next item is presented.

4. In acoustically coupled time-shared systems, data are occasionally lost in transmission between the computer and the testing terminal (e.g., CRT). This results in either the computer being unable to interpret the response a testee has made to a test item, or in a displayed test item which appears as garbled information. In either case, the system must be able to present any test item over again if data should be lost in transmission. Implementation of this feature should not require elaborate instructions.

A computer-controlled ability testing system used for either research or applied testing must be flexible. Thus, the following features are important:

5. When tests are administered for research purposes, the only tangible reward for the testee may be knowledge of "how he did" on the test. Thus, a computerized ability testing system designed for research should give test scores at the end of the test, normative comparisons (if desired by the administrator) and an opportunity for the testee to examine his performance in detail--on an item-by-item basis--under some circumstances. This feature should operate following completion of testing, with no necessity for intervention by the proctor.

6. All computer systems fail occasionally. Failures can result from interruption on the telephone lines, power failures, failure of the computer's central processor, peripheral processor or disks, software failures, or failure of terminal equipment. With the large number of system components subject to failure, any computerized testing system must be capable of recovery with as little loss of data as possible. Thus, when abnormal terminations occur, the proctor should be able to re-start the testee at the item on which termination occurred once the system is again operational. This "recovery" feature would also be important where the testee or proctor interrupts testing for some reason (e.g., testee becomes ill, needs to keep an appointment elsewhere), and it is desired to continue testing at a later time.

7. On occasion it might be desirable to terminate a testee at some point prior to normal termination, not allowing for eventual re-entry at the point of termination. In this case, the proctor would wish to move directly to the post-testing de-briefing routine, including total scores and potential review of performance for the testee.

8. There are many strategies of computerized ability testing (Weiss and Betz, 1973, pp. 15-40; Weiss, 1974). These include two-stage, pyramidal, flexilevel, stradaptive and Bayesian strategies, among others, with new ones yet to be developed. For research purposes, it is desirable to be able to choose a strategy for administration to a given testee at the time he presents

himself for testing. For applied testing purposes, it might be discovered that certain kinds of strategies are appropriate for certain applied measurement situations. Thus, an operational computerized ability testing system must permit the proctor to select, at the time of testing, one of a number of adaptive testing strategies for administration to a given testee. At the same time, different testees being tested simultaneously on different terminals should be able to be administered different strategies.

9. Both applied and research adaptive measurement systems should concern themselves with different ability domains. Thus, the computer system should have several item "pools" consisting of, for example, items measuring verbal comprehension, numerical reasoning, clerical ability, and spatial ability. Any adaptive testing strategy should be able to access any of these pools, with the nature of the pool (i.e., the ability being measured) selected at the point of test administration. Needless to say, the system should also permit the use of different pools with different testing strategies simultaneously.

10. Given an item pool (e.g., vocabulary or number series) and a single testing strategy, different subsets of items should be available for presentation to different testees. For example, it might be appropriate to administer to one testee a 15-stage pyramidal vocabulary test and at the same time administer a 10-stage pyramidal vocabulary test to another testee. Further, the 15-stage test might use a fixed step size while the 10-stage test used a "shrinking" step size. In this case, one item subset would be comprised of the item numbers pre-selected to form the 15-stage fixed step size pyramid, while the other item subset would include a different set of items structured into a 10-stage pyramid. The item subset to be used for a given testee, nested within the total vocabulary item pool and administered under a specified adaptive testing strategy, would be selected at the point of test administration.

11. Some strategies of adaptive ability testing (e.g., stradaptive, Bayesian) require an initial estimate of ability to determine a starting point for the test administration branching procedure. When these strategies are to be administered, the system must request the appropriate information from the testee (e.g., his

cumulative grade point average or his subjective estimate of his standing on the trait in deciles; see Weiss, 1973, pp. 12-15) or obtain it from previously stored data on the testee.

Certain other features are dictated by the requirements of the research design of the present research:

12. Research in adaptive testing should be concerned with studies of the inter-relationships among ability estimates derived from the various testing strategies. Thus, it is desirable to administer two or more tests to a given testee using different testing strategies. At the same time, it might be undesirable for the testee to know that he is completing one testing strategy and beginning another, when the same item pool is being used. Conversely, when two different item pools are being used for the same testee in one test administration (e.g., vocabulary and numerical reasoning) it would be necessary to interpose additional instructions between the two tests. Thus, the system must allow for smooth and uninterrupted flow between two tests within the same item pool, yet permit structured movement between item pools. In addition, it should permit the administration of only one test when multiple tests are not desired.

13. When multiple tests are desired in one test administration period, the system should permit tests to be administered in any order. Such flexibility will permit the analysis of, and control for, order effects in the administration of various testing strategies.

14. Research in adaptive testing should also be concerned with the test-retest stability of the scores obtained from various testing strategies. If the testee is taking a retest, he should be given the option of reviewing the instructions. Further, if an initial ability estimate is needed for the retest, the estimate from the previous administration of that test must be retrieved from a data file. This is necessary to ensure comparability of test and retest results.

Several features of the software system are also important to the programmer:

15. It is obvious that new strategies of computerized ability testing are yet to be developed. In order to facilitate the integration of these new adaptive

strategies into an operational software system, it should be possible to develop and debug individual testing strategy programs as separate units and add them to the system with as little revision of the strategy program and system as possible.

16. Item pools, particularly those used for research, constantly change. New items are added, old ones deleted, and items are constantly re-normed. Thus, it might be appropriate to frequently change item files to be used within a given testing strategy. Likewise, termination rules or criteria within an adaptive strategy, or other internal aspects of a strategy, might require occasional changes. In addition, new item pools, measuring different kinds of abilities, might be added to a testing system. For all these reasons, data which might be frequently changed should be stored separately and be easily accessible so that changes can be easily made.

17. Most time-shared computer systems can have only a limited number of programs in central memory at one time. Programs waiting to be executed are usually entered into central memory in inverse order of the amount of memory they require. Thus, an operational computerized ability testing system must use as little central memory space as possible to speed system response time and eliminate unnecessary delays for the testee.

18. To permit easy analysis of obtained data, the program system must store all obtained data in a form readily amenable to statistical analysis. At the same time, data must be stored as compactly as possible since data storage charges can add an appreciable amount to the cost of computerized testing.

One last feature is especially important to testing systems being used for applied purposes, although it has implications for research applications also:

19. The system should permit file interpretation capabilities. Thus, at any time after the termination of testing, the psychologist should be able to recall the test record of any individual and examine his performance in detail. This would include a reporting of test scores for the strategy or strategies administered to a given testee. Further, the system should permit a graphical display of the test performance on the

adaptive strategies. This display could, for example, chart the testee's progression through a pyramidal test structure, through a flexilevel test, or graph his performance on a Bayesian strategy test. Such displays would also be of great benefit in explaining test performance to testees in a de-briefing session or for didactic purposes in explaining adaptive testing to a variety of audiences.

COMPUTERIZED ABILITY TESTING SYSTEM--FILE AND PROGRAM STRUCTURE

Computers work by manipulating stored data. Data can be stored by a computer in one of two ways: 1) in the computer's central memory, which is a basic storage mode limited in storage capacity but with very rapid accessibility, and 2) on peripheral devices such as magnetic disc or tape which have almost unlimited storage capacity but relatively slower accessibility.

Files for Computerized Ability Testing

A file represents a block of data which is stored on a magnetic disc, much as a file in the usual sense represents a block of information in a filing cabinet.

The testing system works with two types of files: program files and data files. Program files are those which contain the machine language instructions for the different programs and subprograms used. They can be copied from a disc to central memory in a specified order, and then be executed.

Data files contain information used by the testing system at various stages of testing. Some files are updated (i.e., information is added to them) as they are used. The others are information (or reference) files only and as such can be easily modified as testing requirements change from time to time.

Updated Files

1. Testing data. A file is saved for each testee, using his identification number as a key for the file name. The details of this file structure are described in Appendix A. The first 300 characters (i.e., 30 10-character words) on this file contain demographic and summary information. Space is also allowed for some information necessary for the recovery feature. The

remainder of the file contains information on each test item presented. This file is updated after a response is obtained to each item.

2. Identification number file. When a testee enters his identification number it is added to this file. Any other programs which refer to the individual testing data files can use this file to get a list of all subjects who have taken tests.

Information Files

1. CATCODE. Before a subject begins testing, a 3 to 7 character keyword is entered by the proctor. This keyword determines each subject's testing assignments. The testing assignments controlled by the keyword include type of testing strategy, specific item file, and type of feedback. File CATCODE contains a list of the valid keywords at any given time. To simplify the proctor's task, and to reduce errors in entering keywords, each keyword (i.e., combination of testing assignments and conditions) can be coded with a single letter. Thus, by entering a single letter, the proctor can designate the complete course of testing for any given testee. Appendix B explains current keyword characters.

2. ABILS. Two of the testing strategies require an initial ability estimate. For retests it may be desirable to use the initial ability estimate from the first test. ABILS contains initial ability estimates, where taken, for all who took a test the first time and is referred to upon retesting.

3. VOCAB4. A random access file containing all of the vocabulary test items. Random access simply means that any test item can be directly accessed by item number as opposed to reading all items from the first to last sequentially. Random access files thus provide more rapid access to data than do sequential files. This file contains the vocabulary word to be defined, the five response choices and the correct response for each of about 600 vocabulary test items. Each item is identified by a unique item number, the "item reference number."

4. NUM5R. A random access file containing a number series item and correct answers for each of about 200 number series items.

5. MATRIX. A file used only by the Bayesian testing strategy. MATRIX contains item difficulty and discrimination indices on the 328 selected vocabulary items used by the Bayesian strategy program.

6. Item subset files. Each testing strategy program is written to work with sequential or working item numbers 1 through N, where N is the maximum number of items which can be presented by that testing strategy. This simplifies programming by permitting the programmers writing testing strategy programs to avoid dealing with actual item reference numbers. It also permits generality of testing strategy programs so that the same testing strategy can be used simultaneously with different item subsets. Prior to presenting a specific test form, a file is built up containing all of the item reference numbers to be used in that test in the order required by the branching paradigm. The file name consists of a mnemonic for a particular strategy followed by a single digit. This allows up to 9 different item subsets for each strategy. The item subset number is entered with the keyword and can therefore be easily changed from testee to testee. At the time of item presentation, actual item reference numbers are assigned to each item in the testing strategy.

   For example, two-stage testing can administer up to 40 items to any testee. The actual two-stage strategy program works only with working item numbers 1 to 40. A file TWOSTG2 contains 40 item reference numbers ranging from 1 to 600 which are entered into the file in the order required for items 1-40 in the two-stage strategy program. Thus, the program might ask for item 12. Before presenting the item, the reference item number of the 12th item would be gotten from file TWOSTG2, e.g., 283, and item number 283 would be obtained from file VOCAB4 and presented to the testee.

Programs, Subprograms, and Overlays

A computer program is a complete set of instructions for a specified task. A program has a number of functions. It must 1) initialize any values used, or get them from a disc, tape, or other input device such as a CRT, 2) perform any calculations, and 3) cause any calculations yielding data that should be saved for later reference to be output in some retrievable form. As a computer performs the instructions specified in a program it is said to be running or executing that program.

In many instances, subsets of calculations or instructions must be executed repeatedly, using different sets of data. This repeated set of instructions can be written as a subprogram (commonly called a subroutine). A subprogram consists of a set of instructions which is complete for some small task. The subroutine generally receives data from the main program and/or returns data to it. Subroutines can also input and output data. Thus, for example, instead of writing instructions to calculate a correlation coefficient at a number of different points in a program, the programmer would write one subprogram to perform those calculations. He would then refer to (or call) that subprogram whenever computation of a correlation coefficient was needed.

A program and its associated subprograms represent a unit which must be stored in the computer's central memory during execution. Since central memory storage is at a premium, problems arise if a large program (i.e., one with a large number of instructions or stored data) is to be executed. To minimize this problem, overlays are used. An overlay is a program (with associated subprograms) which is stored on a peripheral storage device (e.g., magnetic disc) but, at the direction of another program, can be moved into memory and executed. When it is moved into memory, it replaces part of the program that was previously in memory. Thus a large program can be broken into units (overlays) which are moved into memory and executed at different times. Regardless of the size of the total program, the maximum central memory requirement is determined by the size of the largest combination of overlays which will be used at one time. (The reader unfamiliar with the concept of overlays is referred to Appendix C for a more basic description of the overlay concept.)

Overlay Structure for Computerized Ability Testing

The computerized testing system is currently composed of a main overlay, 5 primary overlays, and 11 secondary overlays. A diagram of the overlay structure is shown in Figure 1.

Main Overlay

The main overlay is in memory at all times. Its primary function is to call the necessary primary overlays in the required order, more than once if necessary. The main overlay also contains a large number of general purpose subprograms used by other overlays. These are mostly machine language subprograms obtained from the system program library which handle special arithmetic processes, and data input and output.

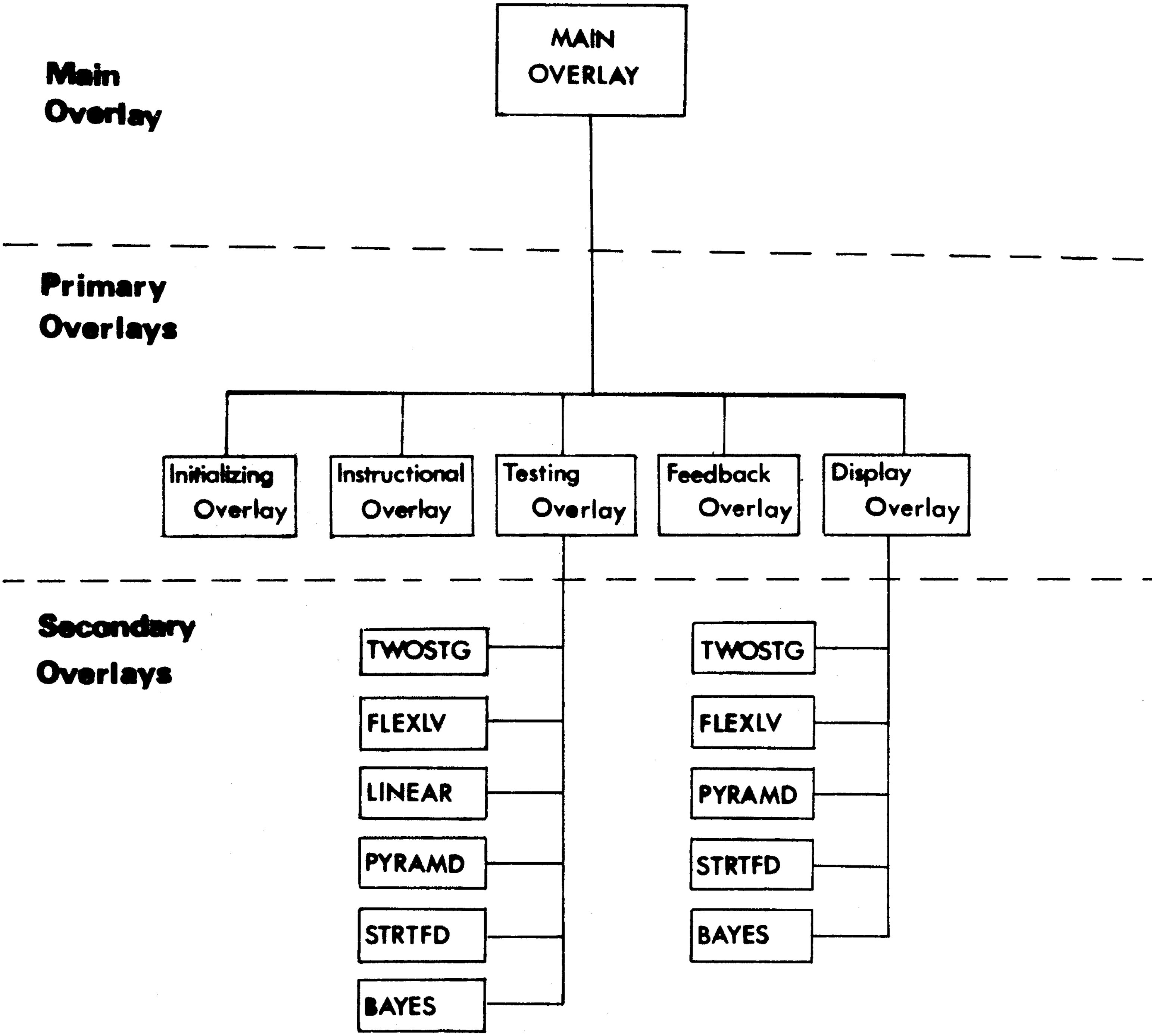


FIGURE 1. OVERLAY STRUCTURE FOR THE SOFTWARE SYSTEM.

Primary and Secondary Overlays

Initializing Overlay. This overlay reads in a keyword or a single-character keyword code which is verified by the CATCODE file, and breaks it into its components. These are checked for validity and coded in a convenient way for access by other parts of the program system (see Appendix B). This overlay locates and then attaches the appropriate item number files. The maximum number of different items in the item subset, maximum number of items to be given (test length) and item numbers to be used are read by this overlay from the item number file for each test to be given to a specific testee.

Instructional Overlay. This overlay presents instructions and collects some demographic data. Date and time of beginning of testing are saved. A series of instructional screens on CRT usage are presented covering such topics as entering information, correcting errors, clearing transmission line errors and entering a "?". These screens, and a flow chart describing their structure, are shown in Appendix D. Most of these screens terminate with a question covering the material presented. The answer is checked, and if it is incorrect the screen is presented up to two additional times. If the testee does not answer correctly on the third attempt a bell is sounded to summon the proctor who can give the subject individualized assistance.

Next, name and identification number are requested. If no identification number (either University or special number given by the proctor) is obtained, a social security number is requested. If a social security number cannot be obtained a bell is sounded and the subject is instructed to get a number from the proctor. If an initial ability estimate is needed, it is requested. If the test is a retest, the estimate will be read from the data file ABILS. After this, several screens containing instructions for either number series or vocabulary items are presented.

When the instructional screens are completed, a data file containing all information obtained to that point is created and saved. The IDFILE containing testee identification numbers is updated and control returns to the main overlay.

A code to skip instructions (S) can be entered with the keyword. Only the identification number is then obtained. Likewise, the keyword can also contain a RETEST code (R). In this case, the subject is given an opportunity to skip the instructions if he feels sufficiently comfortable with equipment usage.

The original instructions were developed for testing University students and were slightly modified for high-school students. The form of instructions to be used is determined by a program switch which is set by the particular procedure file used to initiate the testing program (See Appendix E for an explanation of procedure files).

Testing Overlay. This overlay consists of a program PREITEM and a subprogram ITEM.

PREITEM serves essentially as a routing program, as its main function is to call the appropriate testing strategy overlay. Before it does this, however, it executes a call to subroutine ITEM which instructs that subroutine to perform certain initializing tasks in preparation for test item presentation. After this is accomplished, the appropriate testing strategy overlay is called by program ITEM. The operation of the primary testing overlay is shown diagrammatically in Figure 2.

As Figure 2 shows, program PREITEM first executes an initializing call to subroutine ITEM (A). Once the initialization operations have been completed, control returns to PREITEM (B) which then calls the appropriate testing strategy overlay (C). The testing strategy program then selects a working item number and calls the item presentation section of subroutine ITEM (D). After the item has been presented and a response received, control returns to the testing strategy program for selection of the next item (E). The D and E cycle is repeated until the strategy program specifies termination of testing at which time control returns to PREITEM (F) and then the main overlay.

ITEM is the core of the test administration system. It is responsible for presenting items, reading responses, and saving data. There are essentially two parts: initializing and item presentation. The initialization section is called from PREITEM. Data such as number of questions attempted and number correct are zeroed out. The time when actual testing starts is recorded. The appropriate item pool file (e.g., vocabulary or number series) is attached and prepared for random access read operations.

The item presentation section is called from the testing strategy overlay and is further divided into two sections. One section reads a vocabulary item from the disc, presents it, reads a response and checks it for validity and correctness. The second section performs the same operations but uses number series items. Since the two sections are functionally

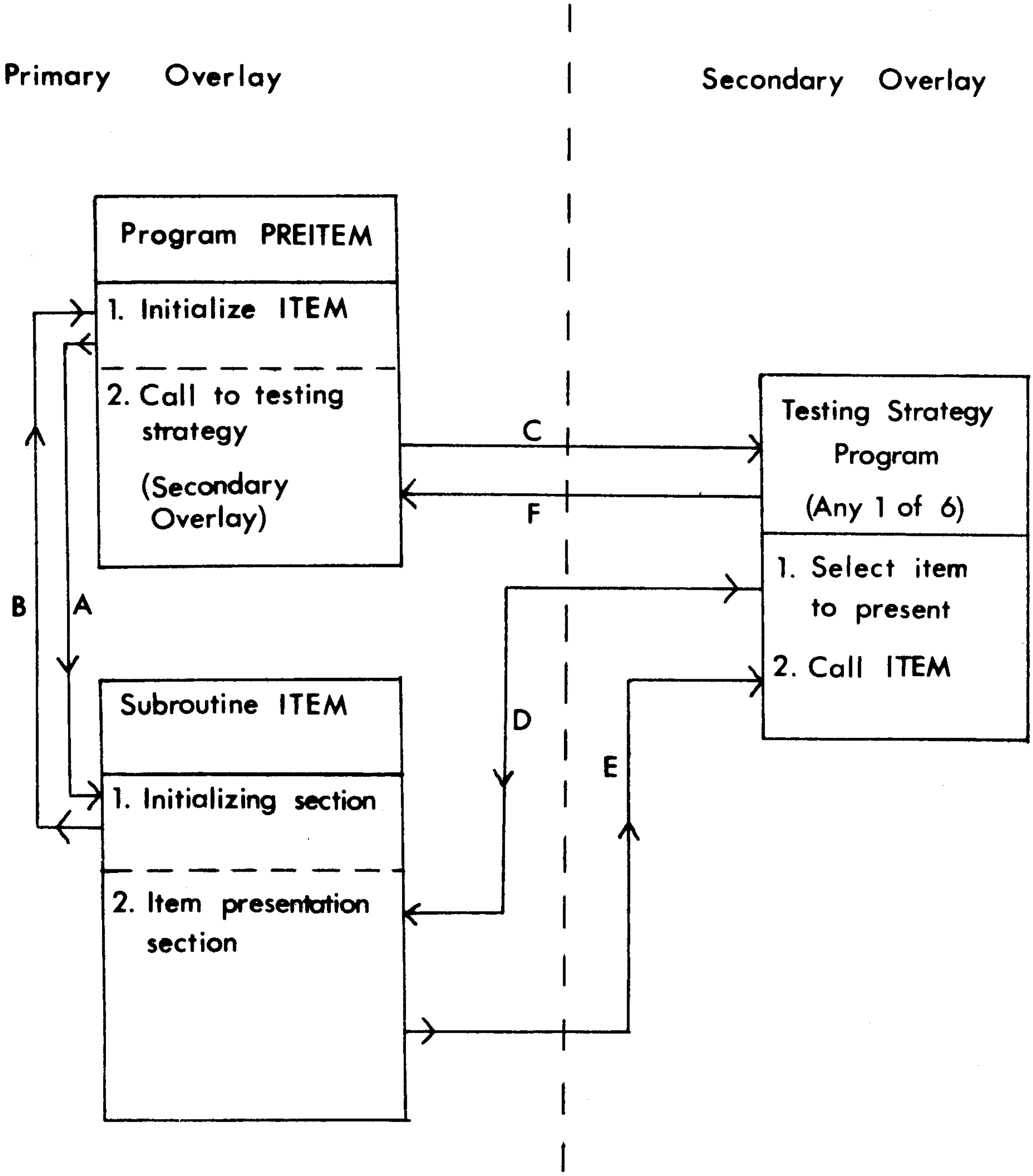


FIGURE 2. STRUCTURE OF THE PRIMARY TESTING OVERLAY AND ITS INTERACTION WITH THE SECONDARY STRATEGY OVERLAYS.

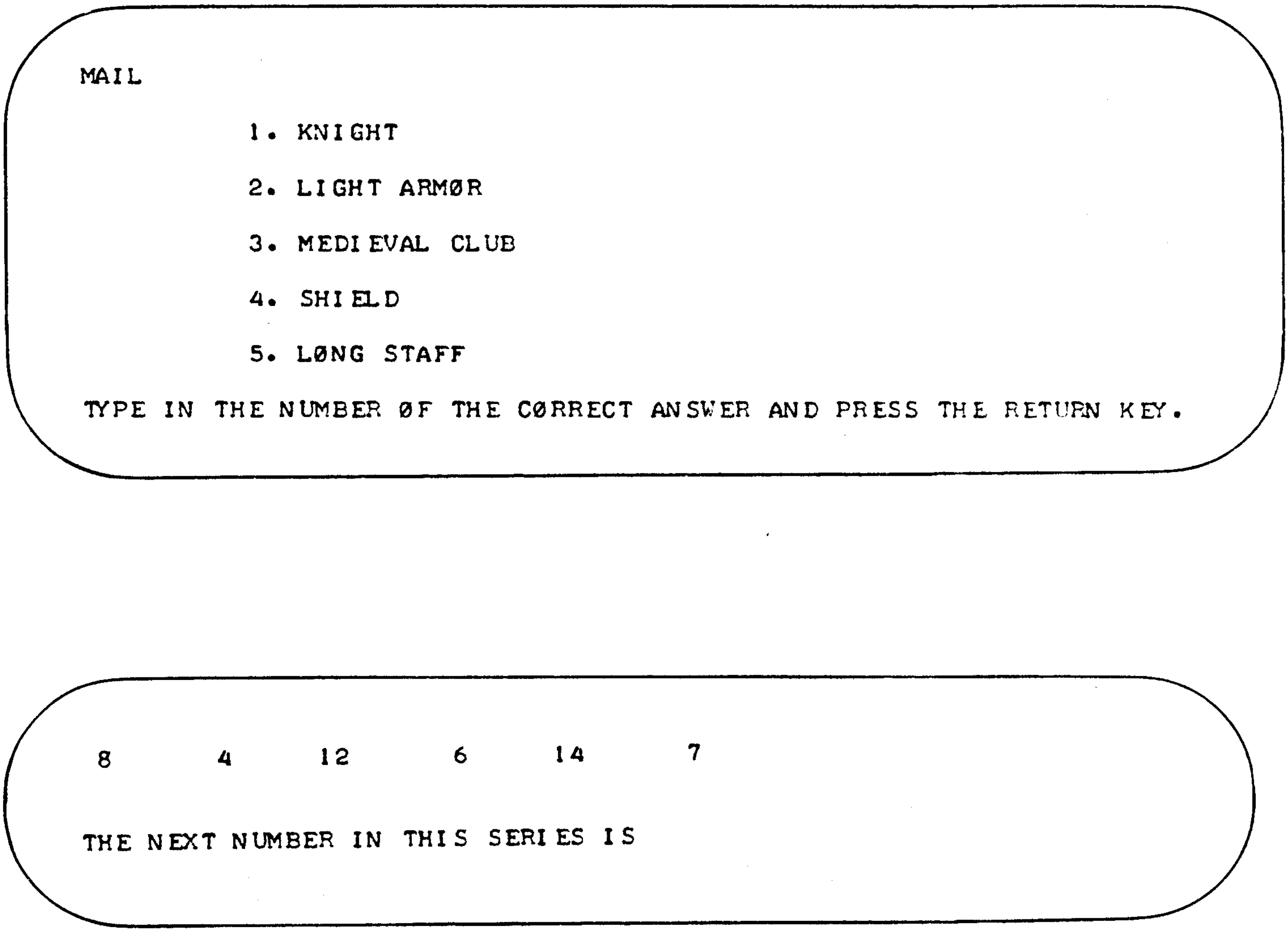
similar no further distinction will be made between them. Figure 3 illustrates the formats used by ITEM in presenting vocabulary and number series items on the CRT.

As Figure 2 shows, the item presentation section of ITEM is called from the individual testing strategy programs in the secondary overlay. That section of ITEM is given (as data) the working number of the item to be presented. On receipt of this information, ITEM performs several operations. First, the testee's data file is updated to include the last item presented as well as any test scores which might have changed. The working item number is then converted to an actual item reference number and that item is read from a disc file. If, for any reason, that item is not available, an error return is given to the testing strategy program which can take appropriate steps. Otherwise, the item is presented and a response requested. The response is checked for validity. If it is invalid, the testee is informed of that fact and the item is presented again. After the third invalid response to any item, a bell is sounded to summon the proctor.

If a valid response is obtained from the testee it is checked for correctness (up to three alternative correct responses are checked for the free-answer number series items). If intermediate feedback (information on correctness/incorrectness of each item) is to be given, one of seven different appropriate feedback messages is presented in a pseudo-random order. Latency of the subject's response is calculated. The response code (right-wrong-?), actual response, item reference number, number of times the item was presented, and the response latency are all added to the subject's data file before the next question is presented. The response code is returned to the testing strategy program. (See Appendix F for an explanation of response codes.)

Secondary Overlays within the Primary Testing Overlay

Testing strategies. There are six secondary overlays associated with the primary testing overlay--one for each testing strategy. These programs are all written with general parameters to utilize any specified item subset file. The programs select a working item number to be presented, and pass that number to subroutine ITEM. Subroutine ITEM locates and presents the appropriate item, processes the testee's response and returns a response code to the strategy program. The strategy program then chooses, in conjunction with the item file it is using, the next item to be presented. A termination criterion is also checked to determine whether or not more items should be presented.



```
MAIL

            1. KNIGHT

            2. LIGHT ARMØR

            3. MEDIEVAL CLUB

            4. SHIELD

            5. LØNG STAFF

TYPE IN THE NUMBER ØF THE CØRRECT ANSWER AND PRESS THE RETURN KEY.
```

```
8      4      12      6      14      7

THE NEXT NUMBER IN THIS SERIES IS
```

FIGURE 3. PRESENTATION SCREENS FOR VOCABULARY AND NUMBER SERIES ITEMS.

Currently, any two testing strategies combined are limited to 500 unique items for potential selection and presentation. Thus, unless one of the testing strategies is the Bayesian strategy, no more than 500 items can be presented in a single testing situation, although the actual number presented to any testee is usually considerably less. The Bayesian testing strategy program draws from its own item pool of 328 items and does not enter into the above combination. Thus, if the Bayesian strategy program is one of the two tests presented to a given testee a maximum of 828 items could be presented in one testing situation.

Six testing strategy programs are currently operational on the secondary testing overlay (see Weiss, 1974 for more complete descriptions of these testing strategies):

TWOSTG. The two-stage testing strategy presents an N_1-item routing test followed by one of 3 or 4 N_2-item measurement tests. The criteria for branching as well as the number of items on the routing (N_1) and measurement (N_2) tests are variable. Currently a 10-item routing test and a 30-item measurement test are being used (see Betz and Weiss, 1973 for empirical results derived from two-stage testing).

FLEXLV. The flexilevel testing strategy uses an item pool with items arranged in order of difficulty. A testee starts with the middle item, an item of average difficulty. A correct response leads to the next more difficult item; an incorrect response leads to the next easier item not previously presented. Currently a 79-item flexilevel structure, in which every testee is presented with 40 items, is being used. The program is written to easily handle flexilevel designs of different numbers of items.

LINEAR. Linear testing presents a fixed number of items in a fixed order (i.e., no branching). It is used for norming and for simulated paper and pencil testing. Currently 40-item linear tests are being used as the "conventional" tests, with which the various adaptive strategies can be compared (e.g., Betz and Weiss, 1973). The linear tests used for item norming consist of varying numbers of items.

PYRAMD. The pyramidal testing strategy is based on a tree structure. All subjects begin with an item of average difficulty. The program selects a more difficult item upon receiving a correct response and a less difficult item following an incorrect response. The program is capable of administering a 24-stage pyramid (i.e., 24 items are administered to each testee). In addition, six different stepping patterns are availble. With this feature, it is possible to skip the next more difficult or easier item and select an item more difficult or easier than that one. This enables coarser adjustments or branching earlier in the test and finer branching later after the testee's approximate ability level has been reached. Additional variations in step size for pyramidal tests can be made by structuring specific pyramidal item subset files.

STRTFD. The stradaptive testing strategy (Weiss, 1973) uses an item pool with items divided into levels or strata based on their difficulty. Testees start at a level determined by a prior ability estimate, which may be requested from the testee. After each response, the stratum from which a testee's items are chosen is examined to determine if he should be administered items from an easier or more difficult stratum. This decision is currently made only on the basis of whether or not the last response was correct, but a number of different criteria (latency, average latency at a level, number correct at a given level, etc.) can easily be included. In addition, differential step sizes are possible which branch the testee up or down by differing numbers of strata. This step size can be unique for each item. This variable step size enables larger branches to be made at the beginning of a test and smaller branches later in the test when an approximate ability level has been established. In its current form, the program can theoretically present 50 items at each of 25 strata. In its current applications the program can present a maximum of 36 items at each of 9 strata. Because of item pool limitations not all strata have 36 items and a total of 179 items can actually be presented.

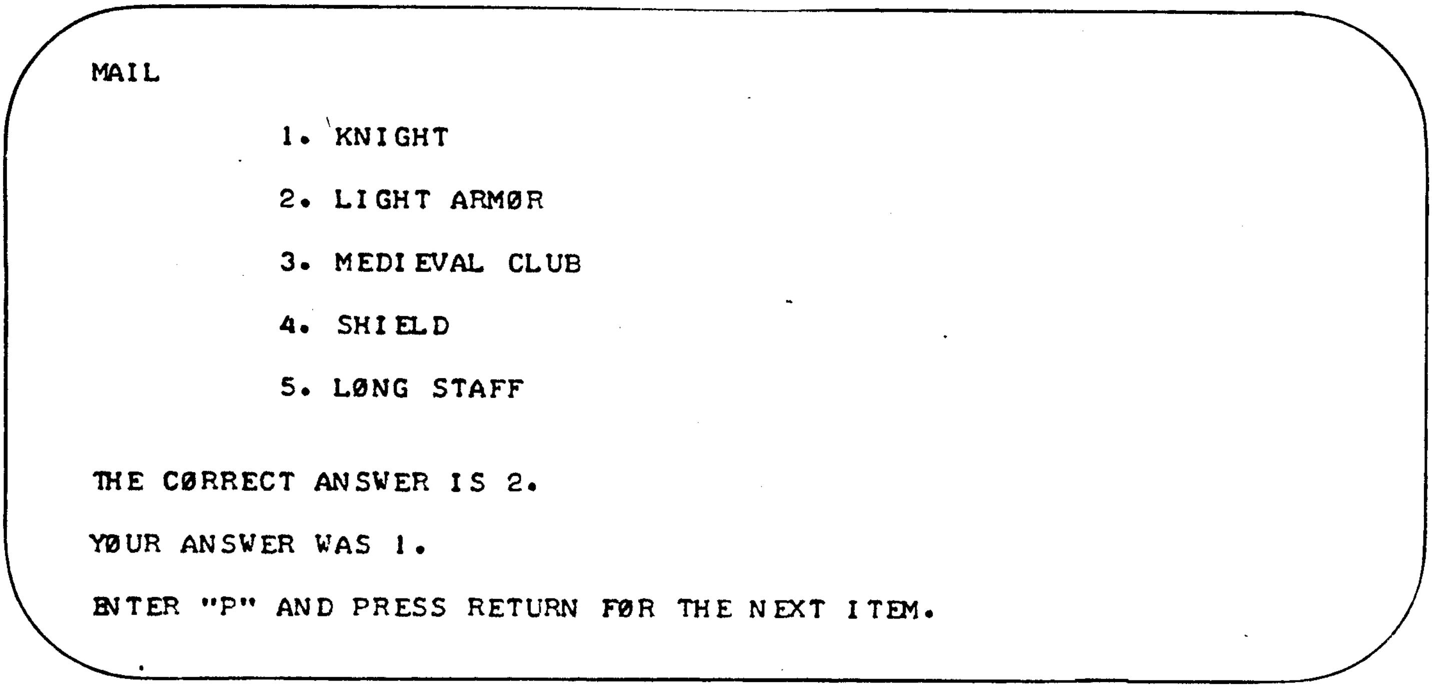
BAYES. The Bayesian testing strategy has no prestructured item pool. Rather, the Bayesian strategy uses a special data file (MATRIX) which presently includes information on the difficulty and discrimination of 328 test items selected from the larger vocabulary item pool. Items are chosen from this special pool using item difficulty and discrimination data in conjunction with estimates of the subject's prior ability based on items previously administered. An initial ability estimate is required and is used as a prior ability estimate to select the first item administered. Following administration of each item, a posterior ability estimate and its variance are computed. Testing terminates after either a specified number of items have been given or a specified posterior variance has been reached.

Feedback Overlay

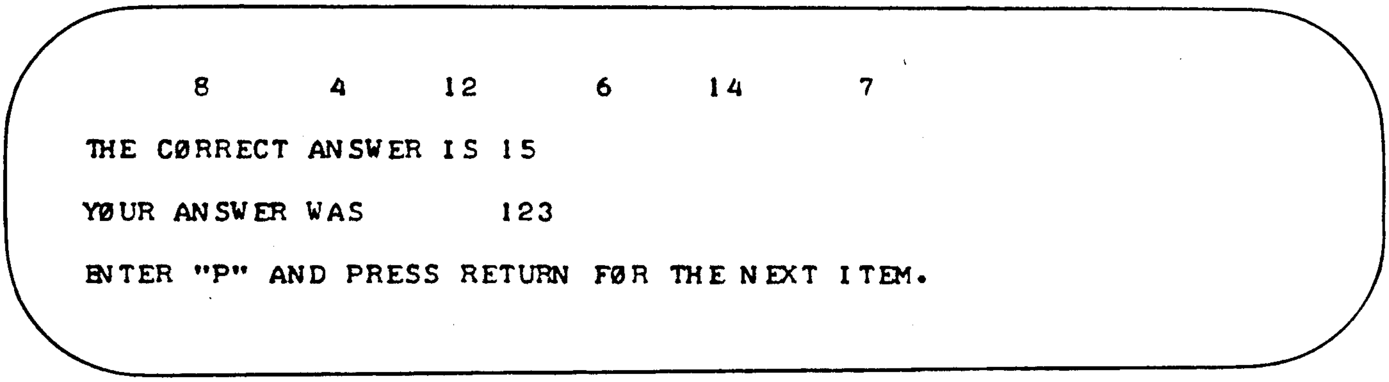
The feedback overlay is a primary overlay which informs the subject of his score on the test or tests administered. If two tests are given and they use different types of items (e.g., verbal and numerical items), a separate score is given for each test. Unless post-feedback (P) for either test was specified by the proctor at the beginning of testing, program operation terminates at this point. If post-feedback was chosen for either test, the subject is given the opportunity to review those test items which he answered incorrectly. This review includes presentation of each incorrect item, the testee's chosen response, and information on the correct response alternative (see Figure 4). Upon completion of the review, or if the testee elects not to review his performance in detail, control returns to the main overlay and the testee is informed that he is free to leave.

Display Overlay

The display overlay, a primary overlay, and its five nested secondary overlays, interpret a subject's data file and provide a graphic portrayal of his movement through the items comprising a test. These overlays can function in two nearly separate modes: 1) As an integral part of the testing system, in which case it is called automatically when the testee has completed testing (both tests if two are given) or, 2) as a separately called overlay.



```
MAIL

          1. KNIGHT

          2. LIGHT ARMØR

          3. MEDIEVAL CLUB

          4. SHIELD

          5. LØNG STAFF

THE CØRRECT ANSWER IS 2.

YØUR ANSWER WAS 1.

ENTER "P" AND PRESS RETURN FØR THE NEXT ITEM.
```



```
      8       4      12       6      14       7

THE CØRRECT ANSWER IS 15

YØUR ANSWER WAS        123

ENTER "P" AND PRESS RETURN FØR THE NEXT ITEM.
```

FIGURE 4. FEEDBACK SCREENS FOR VOCABULARY AND NUMBER SERIES ITEMS.

In the first mode (integral), the data file of the testee who has just completed testing is interpreted and the testee (or the proctor) is given the option of destroying or saving the data file. The second mode (independent) is called by entering "PLOT" when a keyword is first asked for. The system then asks for the identification number or file name of a subject who completed testing at some time in the past. The testee's test results can be plotted and the user is given the option of reviewing other subjects' data files. In either mode, the test or tests the testee took are listed and the user is given the option of displaying either, both, or neither.

Each testing strategy requires a different interpretation and plot so there are currently five seconday overlays, one for each strategy except linear testing.

Plotting programs. These programs provide graphic portrayals of a testee's movement through the item pool, separately for each testing strategy. They also provide scores for each strategy. Sample plots produced by each of these programs are illustrated in Appendix G.

TWOSTG--plots testee performance on the two-stage adaptive test.

FLEXLV--plots testee performance on the flexilevel adaptive test.

PYRAMD--plots testee performance on the pyramidal adaptive test.

STRTFD--plots testee performance on the stradaptive test.

BAYES--plots testee performance on the Bayesian adaptive test.

Additional System Features

Recovery. The testing program is capable of recovering from interruptions caused at any point during testing either by the proctor or computer system failure of any kind. This is accomplished by updating each testee's permanent data file after each test item is presented. This update reflects information on the last item presented as well as information on up to 24 key values which enable the exact item on which interruption occurred to be found. These key values vary between testing strategies, but in general include information such as the number of questions taken in different parts of the test. The recovery procedure identifies the last item presented, i.e., that on which termination occurred, and that item is presented again but is recorded as having been presented more than once. To initiate recovery, the proctor enters RECOVER when the system requests a keyword. The testee's identification number is then requested. The subject's data file is then located, and searched to find the last item completed. The item on which interruption occurred is then presented again and testing continues from that point. Recovery is possible immediately after abnormal termination occurs, or can be initiated at any future time, as long as the testee's data file is available on the disc.

Program expansion. The individual testing strategies can be written and debugged as simple programs. The only interaction they have with other overlay sections is in accepting a prior ability estimate if needed, requesting an item and accepting the response code for the answer. Once the strategy program is working properly, the addition of a few statements to it and to the main overlay permit it to be added to the overlay system.

Data changes. Item subset files and keywords are the most frequently changed information. This information is stored on separate files which can be easily changed without requiring regeneration of the entire overlay system.

Proctoring. Proctors need have no knowledge of computer programming. Once they have logged onto the computer (i.e., dialed the telephone and logged in by account number and password) two 2-word commands are necessary to load the program (see Appendix E). The program then asks for a keyword. A keyword of up to seven characters or a single code letter can be entered. The test is completely self-administering after that point. The proctors must be able to aid subjects who have difficulty with the instructions or with entering their names or identification numbers. This aid requires only knowledge of the terminal hardware. Beyond that, the proctor must be able to recognize a variety of system problems that might occur (e.g., telephone line problems, computer instability, terminal failure) and initiate recovery procedures when necessary.

Computer hardware and software requirements.

Technical requirements (e.g., computer size and characteristics, transferability) of the software system are detailed in Appendix H.

References

Betz, N. E. & Weiss, D. J. An empirical study of computer-administered two-stage ability testing. Research Report 73-4, Psychometric Methods Program, Department of Psychology, University of Minnesota, 1973.

Weiss, D. J. The stratified adaptive computerized ability test. Research Report 73-3, Psychometric Methods Program, Department of Psychology, University of Minnesota, 1973

Weiss, D. J. Strategies of computerized ability testing. Research Report 74-x, Psychometric Methods Program, Department of Psychology, University of Minnesota, 1974 (in preparation).

Weiss, D. J. & Betz, N. E. Ability measurement: conventional or adaptive? Research Report 73-1, Psychometric Methods Program, Department of Psychology, University of Minnesota, 1973.

APPENDIX A

Description of Data Stored on each Testee's Data File

Data is stored in 10-character words

Word No.	Data
1	Identification number or Social Security number.
2	Keyword as entered by proctor.
3	Current location in program: 0-1000 instructions 1001-2000 Test 1 2001-3000 Test 2, if given 3001-4000 Post-Feedback
4	Elapsed time in seconds from time subject began instructions until testing was completed.
5	Total time in seconds spent on instructions.
6	Total time in seconds spent on test 1.
7	Number of errors on instructional screens 1-10, 1 character per screen.
8	Number of errors on instructional screens 11-20 1 character per screen.
9	Number of items correct on test 1.
10-12	Testee's name, 30 characters.
13	Characters 1-2: subject's estimated ability, if taken. Characters 3-8: blank Characters 9-10: college code (01-27)
14	Social security number, if available.
15	Date of testing.
16	Seconds since midnight when testing began.
17	Elapsed time in seconds spent on test 2.
18	Maximum number of questions which could be given on test 1.
19	Maximum number of questions which could be given on test 2.
20	Number of items attempted on test 1.
21	Number of items attempted on test 2.
22	First score on test 1.
23	First score on test 2.
24	(reserved for program for recovery information)
25	Number of items correct on test 2.
26	Second score on test 1.
27	Second score on test 2.
28-30	(reserved for program for recovery information)

Structure of the testing data file (cont'd)

Data on each vocabulary item is packed into one word as follows:

character 1: response code

code	meaning
0	Item answered incorrectly
1	Item answered correctly
2	Item answered with a ?

2: actual response (1-5, ?=6)
3-6: reference number of item presented
7: number of presentations of screen
8-10: response latency in seconds

Data on each number series item is packed into 2 words as follows:

Word 1: Character 1: response code (same as vocabulary items)
2-3: blank
4-10: actual response

Word 2: Character 1-4: item reference number
5-6: blank
7: number of presentations of screen
8-10: latency in seconds

In addition, for the Bayesian testing strategy, a prior ability estimate and prior variance estimate are calculated for each item and saved packed into one word as follows:

characters 1-5 - prior ability estimate
6-10 - prior variance estimate

The length of each file is thus dependent upon the number of items given, the type of items given, and the testing strategy used.

APPENDIX B

Keyword Codes

Before beginning testing, the testing program requests a keyword which must be entered by the proctor. This keyword completely describes the testing assignments for a testee. It consists of 3 to 7 characters composed of 1 or 2 three-character test keys plus an optional special instructions character.

Each three-character test key has the following format:

Character 1: Code letter for test

A - Stradaptive
B - Bayesian
F - Flexilevel
L - Linear
T - Twostage
Y - Pyramid

Character 2: Numeral 1 through 9 indicating which of nine potential item subsets associated with the particular testing strategy is to be used.

Character 3: Feedback codes

N - No feedback
I - Intermediate feedback (i.e., feedback after each item)
P - Postfeedback (i.e., feedback after completion of testing)

There are two special instruction characters:

1) R: Retest - gives the testee the option of skipping instructions, if the test administration is a retest.

2) S: Skip instructions (get the identification number only). This option is used mostly by staff in trial test situations.

These two characters can be entered after a 3-character test code (between or after if two tests are given). No character is used for more than one function; each character position has its own unique code letters.

Keyword Codes (cont'd)

Examples of keywords:

1. T3I--One test, a two-stage test (T) is to be given. The reference item numbers to be used will be read from a file TWOSTG3. Intermediate feedback (I) will be given.

2. Y5NL4PR or Y5NRL4P--Two tests, a pyramidal (Y) and linear (L) will be given. The pyramidal test will get item reference numbers from a file PYRAMD5 and will present no feedback. The linear test will get item reference numbers from file LINEAR4 and will present post-feedback (P). The R designates that this is a retest situation and the testee should be given the option of reviewing instructions.

Prior to presenting a series of tests, a file (CATCODE) can be prepared containing a list of all keywords which will be valid for a period of time. Each keyword has a single associated letter which can then be entered instead of the keyword. If the proctor enters a single letter when the keyword is requested, the program refers to this file and converts the entered letter to its associated keyword. The letters used to identify keywords are chosen to minimize both memory and typing errors by the proctors.

In addition, there are three special cases where special commands other than a keyword may be entered:

RECOVER: If this is entered after an abnormal termination of testing, the program will ask for the testee's identification number, and will then recover available data and resume testing.

SAMPLE: This instructs the program to ask for a keyword again and to automatically call the display overlay upon completion of testing.

PLOT: This instructs the program to call the display overlay directly and to request information as to which testee's file should be displayed.

APPENDIX C

An Explanation of Computer Program Overlays

A comparison of computer programming with programmed learning texts might be useful in understanding the overlay concept. In programmed instruction texts the student usually proceeds through a series of frames. Each frame contains some substantive information, and the frames are usually followed sequentially. However, some frames may contain questions which, depending on the student's answer, may branch him/her to a review section, to an explanation, or to new material. The process followed here is analagous to a computer executing a program; instructions in the program are executed sequentially, although differential branching can occur at a number of points in the program.

For illustrative purposes assume that a programmed learning text is designed to teach statistics. At several points in this text it might be appropriate to lead the student, step by step, through the calculation of the standard deviation of different sets of numbers. Rather than repeating the instructional frames designed to lead the student through the calculational steps each time they are needed, it might be easier to have a section of the text called "Calculation of the Standard Deviation". At any point in the text where it was necessary to calculate a standard deviation, the student would be helped to organize the set of numbers he was using at that point and then referred to the "Standard Deviation" section for calculations. When the student finished the calculations he would be instructed to remember or record the value of the standard deviation and to return to the instructional frame which referred him to that section. The section on the calculation of the standard deviation could then be used any number of times with only a few extra instructions. This calculational section in a programmed instruction text is directly comparable to a subprogram or subroutine in computer programming.

Now, expand the programmed learning text concept into a series of texts for a complete course. Assume that at any given time a student is permitted to have in his possession a maximum of three texts--a main text, a primary text, and a secondary text. When the student first starts he is given the main text, as are all students. The main text might describe what primary and secondary texts are available to the student, give some examples of how to proceed through the text structure, determine where the student should start, and contain any frequently used or basic equations, calculations or tables. This main text might then refer the student to a primary text, for example a text on Bayesian statistics. The primary text would

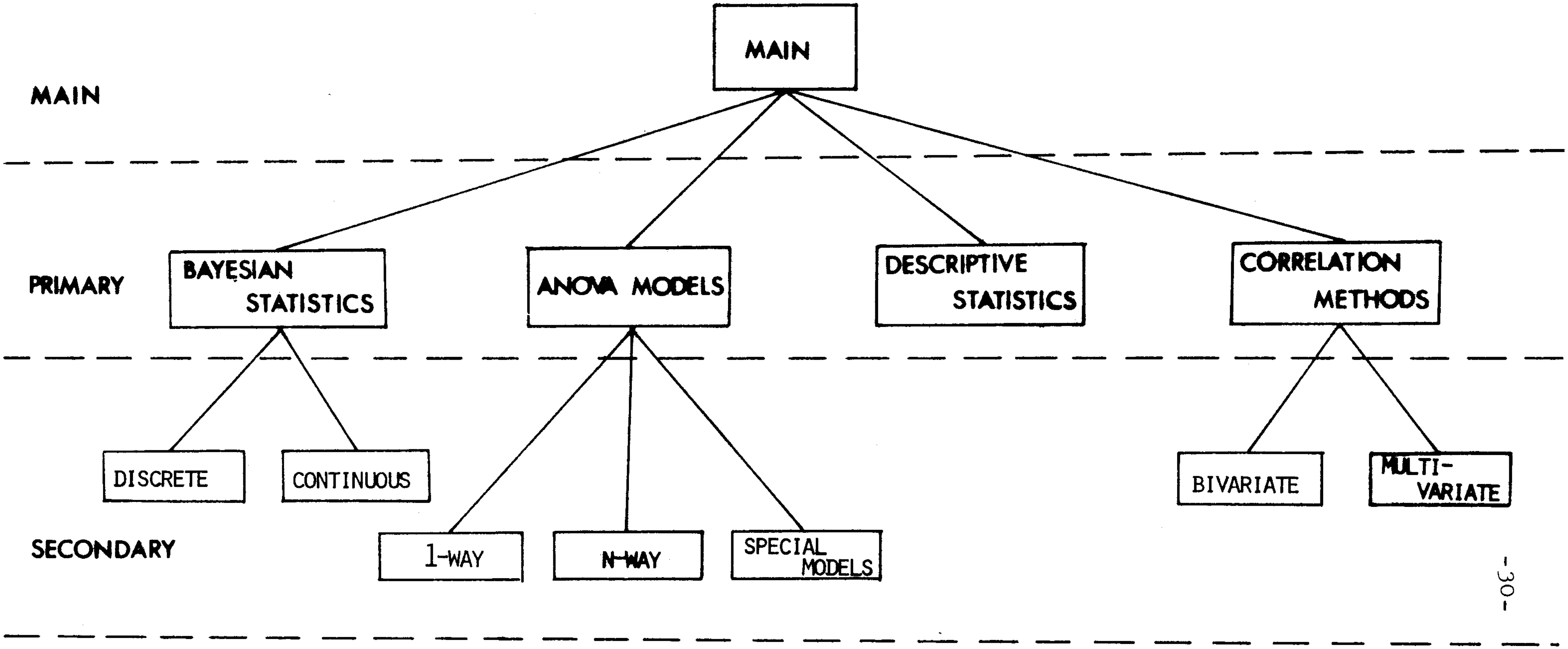


FIGURE 5. HYPOTHETICAL TEXT STRUCTURE FOR PROGRAMMED INSTRUCTION IN STATISTICS.

have its own instructional frames, equations, tables, etc. The student would now have two texts. The primary text could refer to equations or tables that it contains itself or to the more general ones in the main text. If the student is interested, the primary text might refer him to one of a number of available secondary texts, for example one on discrete Bayesian models or one on continuous Bayesian models. The secondary text would also have instructional frames, tables, equations, etc.

Figure 5 illustrates a possible text structure for the hypothetical programmed instructional sequence in statistics. The main text would give some basic information on statistics, instruct the student on use of the texts, aid the student in selecting which other texts he might want to use and contain some of the more basic, generally-used equations, formulas or tables.

Four primary texts are available: Bayesian statistics, analysis of variance models, descriptive statistics, and correlational methods. Each primary text would contain substantive information on its own special topic and general equations related to that topic and could aid the student in deciding whether or not he wanted to use any of the more specialized secondary texts.

There are seven secondary texts, each one nested within a primary text. Discrete and continuous models are nested within Bayesian statistics. One-way, N-way and special models are nested within ANOVA models. There are no secondary texts associated with the primary text on descriptive statistics. Bivariate and multivariate correlational methods are nested within the correlational methods primary text. These secondary texts contain the most specific substantive information and most specific equations for various calculations.

Figures 6a to 6f illustrate several possible configurations within the given text "system." A minimal configuration is illustrated in Figure 6a. A student who has just begun would have this configuration. If he should decide to study the text on Bayesian statistics he would then get that text; this configuration is shown in 6b. Note that he still has the main text; this enables him to refer to any of the basic equations in that text. If this student has no interest in either discrete or continuous Bayesian statistics he would not use either of the secondary texts nested within the Bayesian statistics text.

Figure 6c illustrates the next configuration. The primary text on Bayesian statistics has been returned and the text on

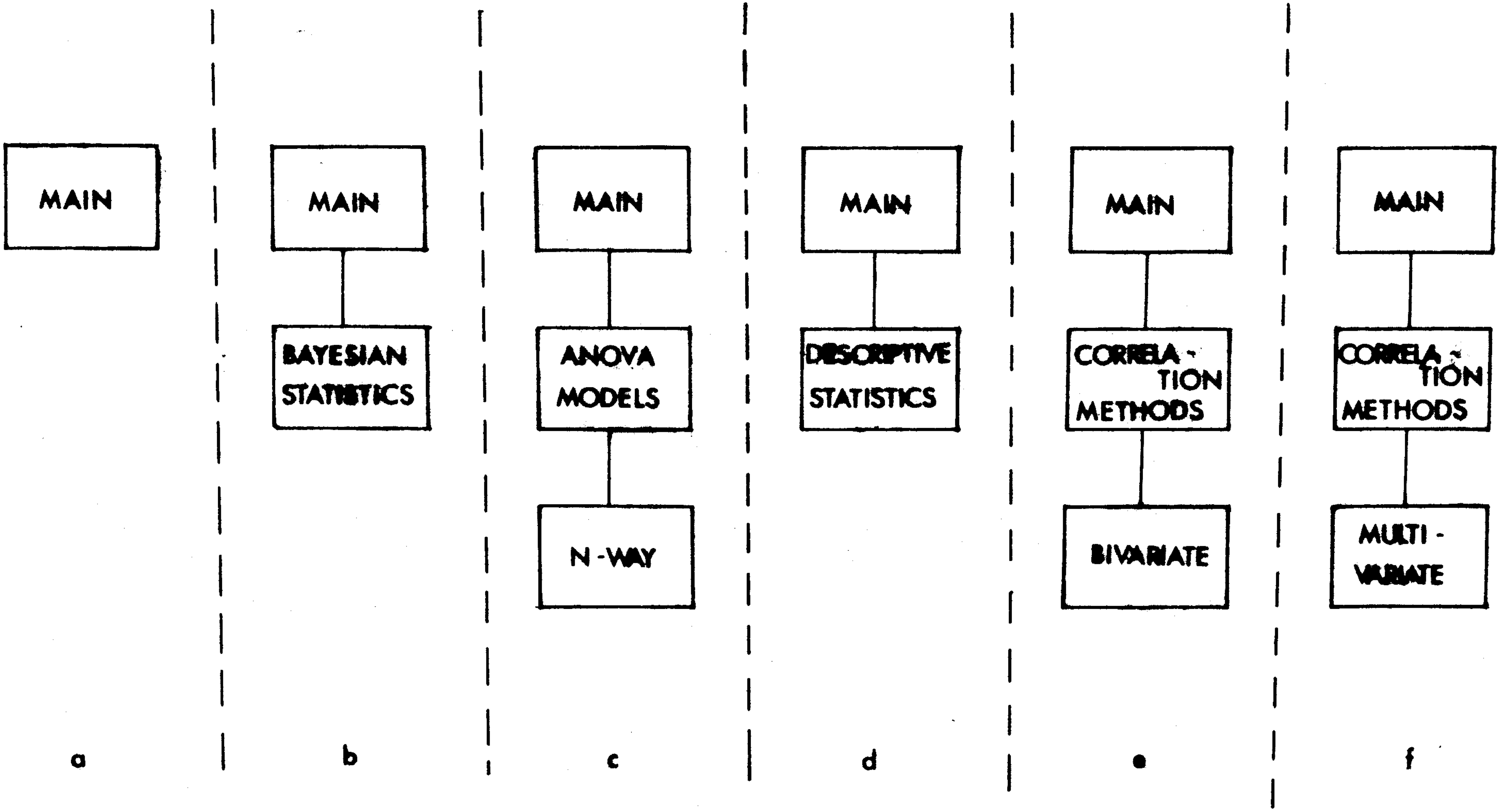


FIGURE 6. SOME POSSIBLE CONFIGURATIONS OF TEXT USAGE IN SPECIFIC APPLICATIONS OF THE PROGRAMMED INSTRUCTIONAL MATERIALS.

ANOVA models has been taken in its place. If the student is concerned only with N-way ANOVA models, he then obtains the secondary text (nested within ANOVA models) on N-way ANOVA models; he then has three integrated texts in his possession.

In Figure 6d, both the primary text on ANOVA models and the secondary text on N-way ANOVA models have been returned and the primary text on descriptive statistics has been taken in their place. Note that this text does not have any associated secondary texts. There can be no secondary text in use with this primary text.

Figures 6e and 6f illustrate two possible configurations with the primary text covering correlational methods. In 6e, the secondary text on bivariate methods is in use. In 6f, the secondary text on multivariate correlational methods is in use. Note that there is no change in the primary text on correlation methods.

As is shown in Figure 6, the student could have one, two, or three texts in his possession at any time. He must always have a main text (Fig. 6a) and can never have a secondary text without a primary text and a main text (Figs. 6c, 6e, and 6f) or a primary text without a main text (Fig. 6b and 6d).

Since a main text does not necessarily require that a primary or secondary text be used (Fig. 6a), it should be understood that a main text cannot refer to any equations in a primary or secondary text. Likewise a primary text may not have any nested secondary texts (Figs. 6b and 6d) and cannot refer to any equations in a secondary text. A secondary text does require a primary and main text to be present (Figs. 6c, 6e, 6f) and can therefore refer to equations in either one.

The structures involved in these imaginary learning texts are directly analagous to those in an overlay system of computer programs. Figure 7 illustrates an overlay configuration where a main, primary and secondary overlay are stored in central memory at one time. The arrows indicate the direction in which overlays themselves or subprograms within an overlay can be referenced or called. There is a main overlay which consists of a program and associated subprograms. It can refer to one of a number of primary overlays, each of which is another program with related subprograms. Each primary overlay can refer to any subprograms in the main overlay or to any of its own subprograms. It can also refer to its secondary overlays but to none of the subprograms in the secondary overlay. The secondary overlay can refer to subprograms in any of the three overlays within its vertical structure.

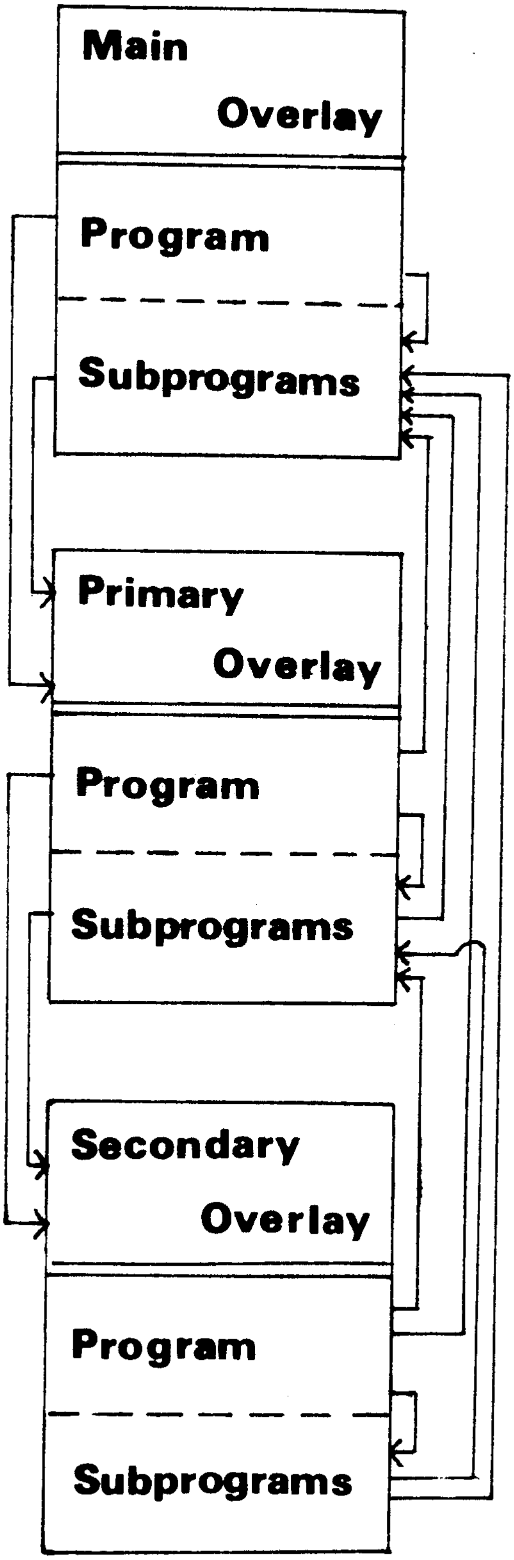


FIGURE 7. A COMPUTER OVERLAY CONFIGURATION.

If the primary overlay calls (i.e., needs to execute) a new secondary overlay, the new one replaces the old one in central memory. Similarly, if the main overlay needs to execute a new primary overlay, the old primary overlay and any of its associated secondary overlays are moved out of central memory. The main overlay is always stored in central memory. A secondary overlay cannot be referred to by more than one primary overlay (i.e., secondary overlays are "nested" within the primary overlay) although it can be referenced any number of times by that primary overlay.

The major advantage of the use of overlays is that very large computer programs can be sub-divided into sub-units which, under control of a master sub-unit (main overlay) can be moved in and out of central memory as needed. Thus, very large programs can be executed with relatively small central memory requirements. This is quite important in time-shared computer systems which usually give higher program execution priority to those programs requiring small amounts of central memory. Another major advantage is that each overlay can be written and tested independently, greatly simplifying the programmer's task. The programmer developing a new overlay need not be concerned with the entire overlay structure but only with those sections directly related to the one he is developing.

APPENDIX D

Instructional Screens

There are currently 36 instructional screens. They are presented in this appendix in the order in which they are presented to a testee. Because there are a number of different testing conditions each screen is coded according to which conditions it represents. The codes follow:

N--number series items
V--vocabulary items
ER--error screen, presented if a previous screen was answered incorrectly
R--retest
S--Instructions are to be skipped and only identification number requested
HS--Screen for presentation to high-school students
AB--Screens requesting ability estimate for stradaptive and Bayesian testing strategies
F--feedback will be given
NF--no feedback will be given

Figure 8 illustrates the various paths that a single testee might follow through the instructional overlay. Each square represents an instructional screen and is numbered in accordance with the screens illustrated on the following pages and abbreviations listed above.

There are five entry points. A is the standard entry point for a first time testee or a retest testee who chooses to review the instructions. B and C are entry points for testees who are to receive a retest (R specified on keyword). If these testees elect to review the instructions, they are routed to entry point A and exit through RETURN-H. If they elect not to review the instructions, screens 5-8 may be presented to obtain name, identification number and ability estimate, and exit through RETURN-G.

D is the entry point if S is specified on the keyword. Only identification number and ability estimate (if needed) are taken and the exit is through RETURN-G.

E and F are intermediate entry points (between tests) for the situation where tests 1 and 2 use different forms of items. The testee is informed of the change in item type and given the option of going over the instructions for the new type. If he choses to skip the instructions, he exits through RETURN-G and begins Test 2. If he elects not to skip the instructions, he is presented those instructions relating to the item form on Test 2 and exits through RETURN-H.

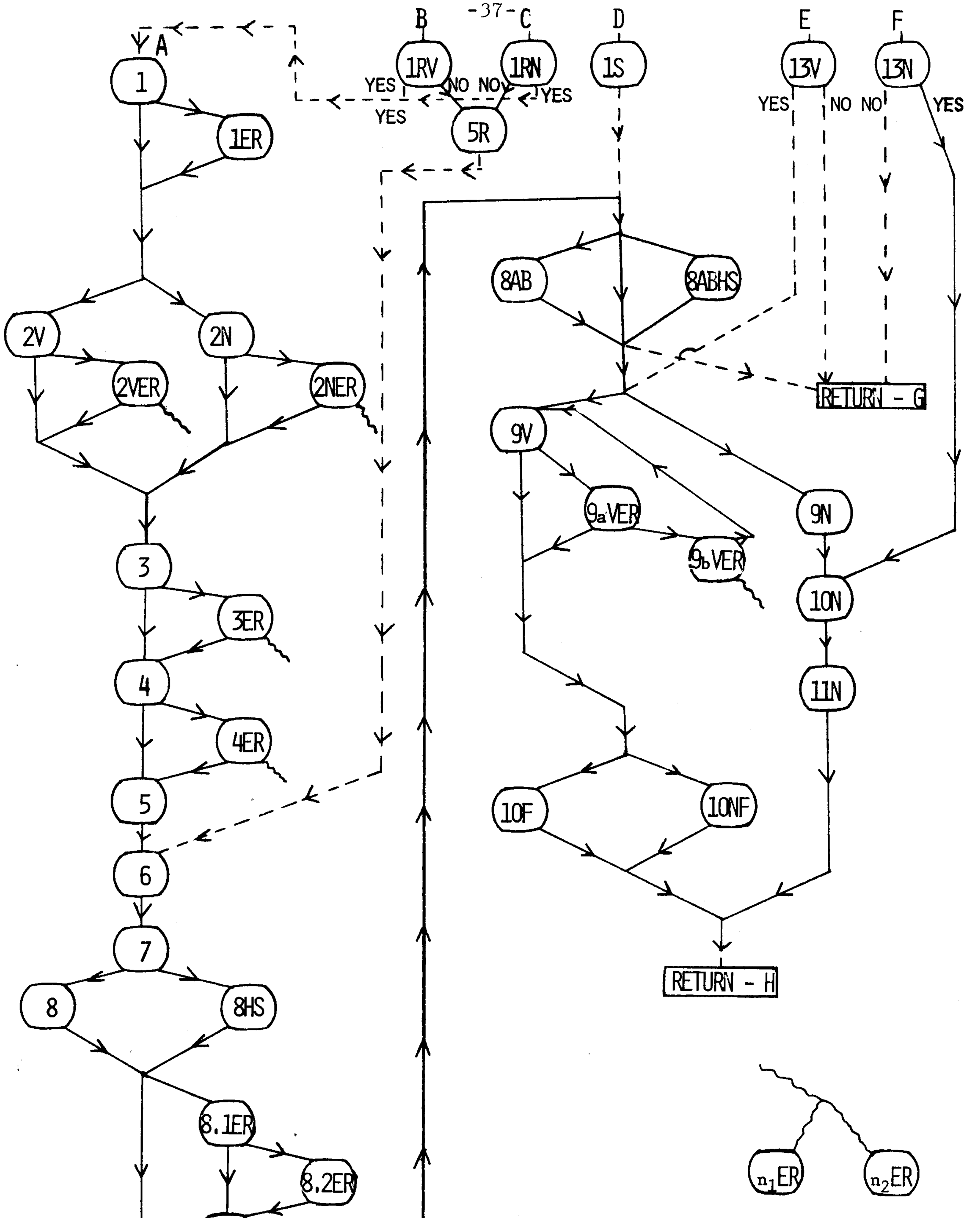
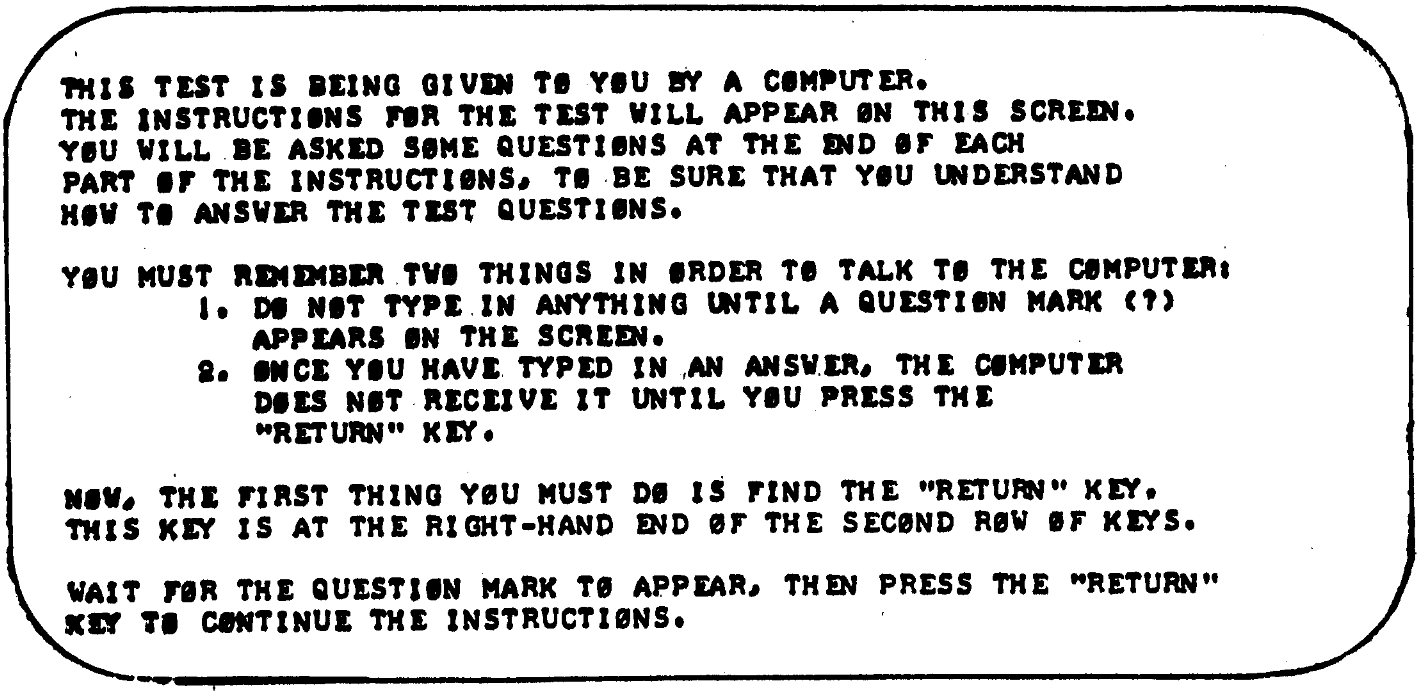


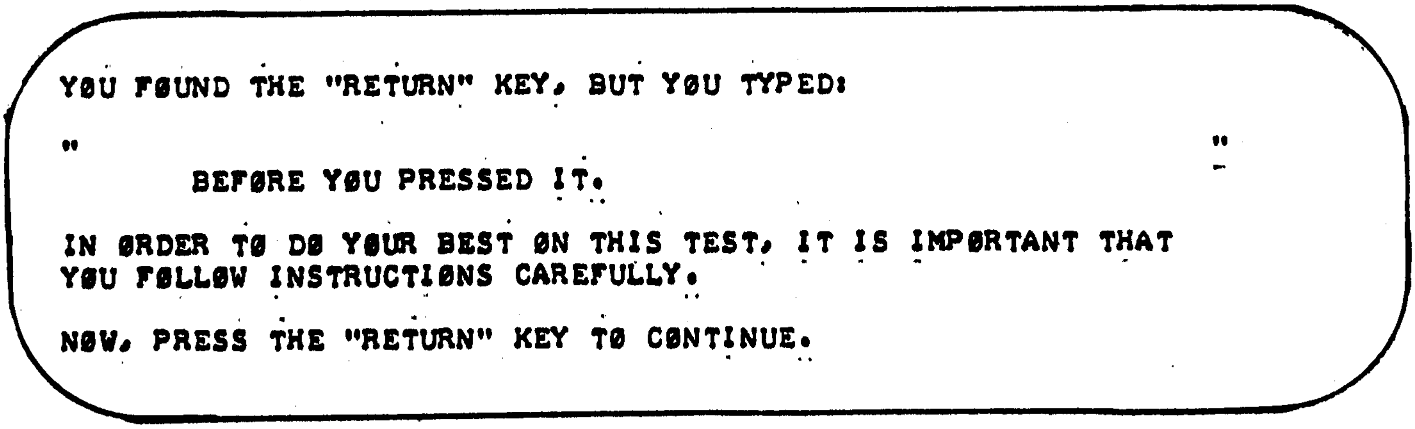
FIGURE 8. Diagram of Possible Alternative Routes Through Screens in Instructional Overlay

The wavy line (∿) indicates a jump to error screens n_1ER and n_2ER. These two screens summon the proctor to aid a testee who is having difficulty with the instructional screens. n_1ER is presented the first time the testee has difficulty; n_2ER is presented if the proctor is called again. After these screens are presented the screen on which the testee encountered difficulty is presented again to be certain that the testee understands those instructions.



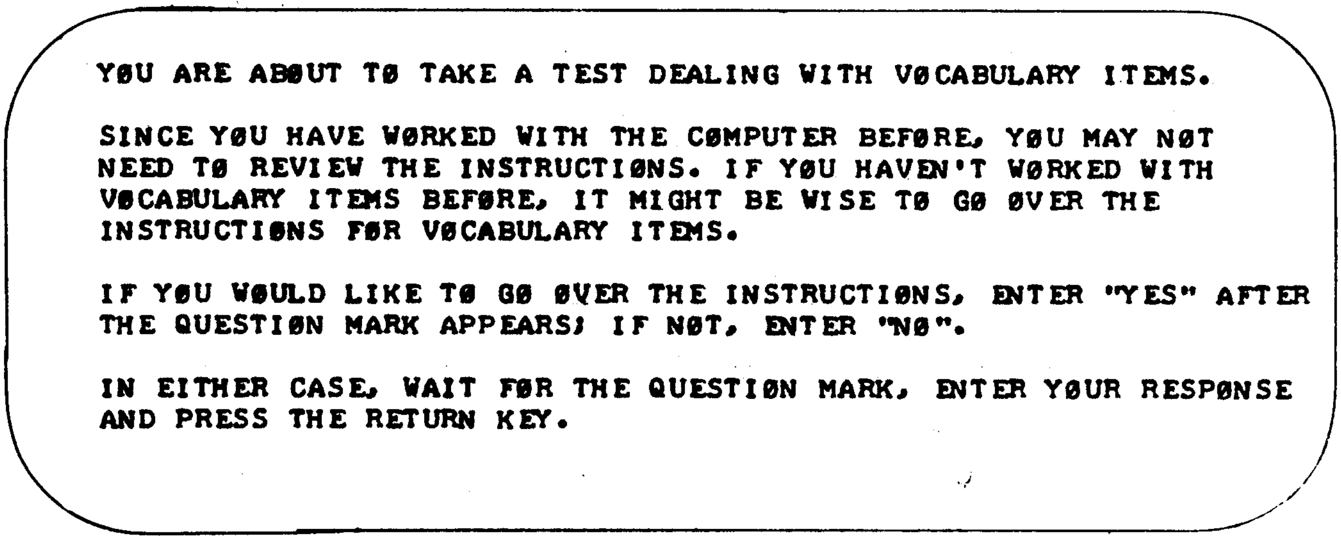
```
THIS TEST IS BEING GIVEN TØ YØU BY A CØMPUTER.
THE INSTRUCTIØNS FØR THE TEST WILL APPEAR ØN THIS SCREEN.
YØU WILL BE ASKED SØME QUESTIØNS AT THE END ØF EACH
PART ØF THE INSTRUCTIØNS, TØ BE SURE THAT YØU UNDERSTAND
HØW TØ ANSWER THE TEST QUESTIØNS.

YØU MUST REMEMBER TWØ THINGS IN ØRDER TØ TALK TØ THE CØMPUTER:
        1. DØ NØT TYPE IN ANYTHING UNTIL A QUESTIØN MARK (?)
           APPEARS ØN THE SCREEN.
        2. ØNCE YØU HAVE TYPED IN AN ANSWER, THE CØMPUTER
           DØES NØT RECEIVE IT UNTIL YØU PRESS THE
           "RETURN" KEY.

NØW, THE FIRST THING YØU MUST DØ IS FIND THE "RETURN" KEY.
THIS KEY IS AT THE RIGHT-HAND END ØF THE SECØND RØW ØF KEYS.

WAIT FØR THE QUESTIØN MARK TØ APPEAR, THEN PRESS THE "RETURN"
KEY TØ CØNTINUE THE INSTRUCTIØNS.
```

Screen 1. First instructional screen



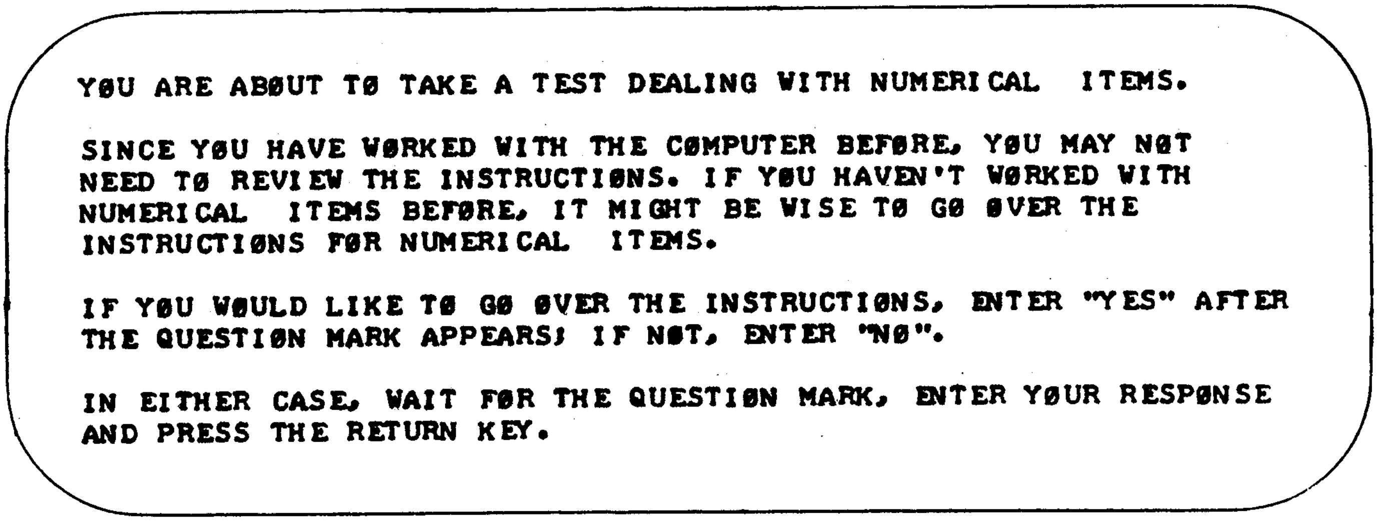
```
YØU FØUND THE "RETURN" KEY, BUT YØU TYPED:

"                                                          "
       BEFØRE YØU PRESSED IT.

IN ØRDER TØ DØ YØUR BEST ØN THIS TEST, IT IS IMPØRTANT THAT
YØU FØLLØW INSTRUCTIØNS CAREFULLY.

NØW, PRESS THE "RETURN" KEY TØ CØNTINUE.
```

Screen 1 ER. Screen presented if screen 1 is in error



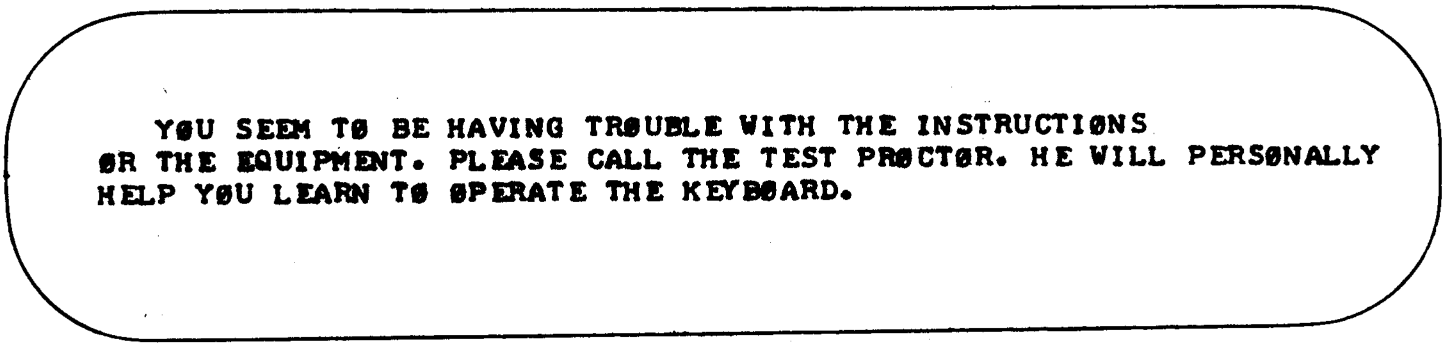
```
YØU ARE ABØUT TØ TAKE A TEST DEALING WITH VØCABULARY ITEMS.

SINCE YØU HAVE WØRKED WITH THE CØMPUTER BEFØRE, YØU MAY NØT
NEED TØ REVIEW THE INSTRUCTIØNS. IF YØU HAVEN'T WØRKED WITH
VØCABULARY ITEMS BEFØRE, IT MIGHT BE WISE TØ GØ ØVER THE
INSTRUCTIØNS FØR VØCABULARY ITEMS.

IF YØU WØULD LIKE TØ GØ ØVER THE INSTRUCTIØNS, ENTER "YES" AFTER
THE QUESTIØN MARK APPEARS; IF NØT, ENTER "NØ".

IN EITHER CASE, WAIT FØR THE QUESTIØN MARK, ENTER YØUR RESPØNSE
AND PRESS THE RETURN KEY.
```

Screen 1 RV. First screen presented to testee taking a retest for which the first test is vocabulary items. If testee responds YES entire instructional series is presented beginning with 1. If testee responds NO screens 5R through 9 are presented.



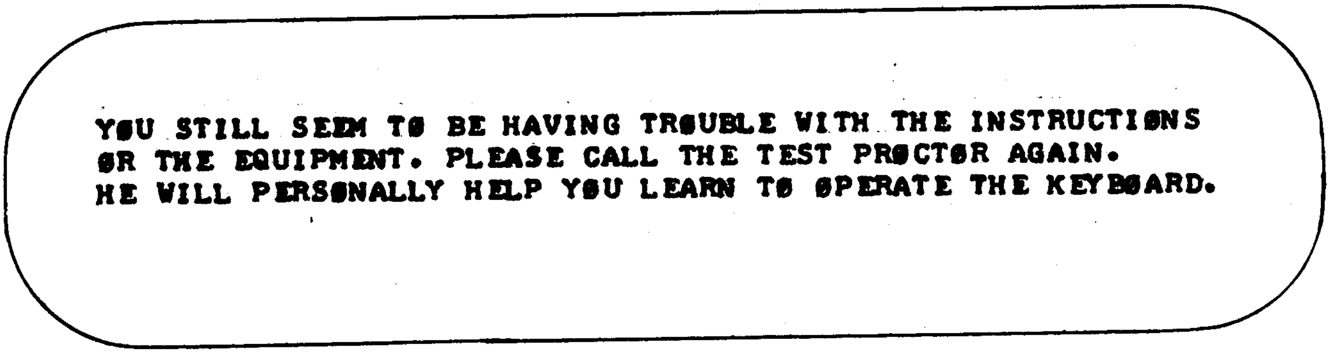
```
YØU ARE ABØUT TØ TAKE A TEST DEALING WITH NUMERICAL  ITEMS.

SINCE YØU HAVE WØRKED WITH THE CØMPUTER BEFØRE, YØU MAY NØT
NEED TØ REVIEW THE INSTRUCTIØNS. IF YØU HAVEN'T WØRKED WITH
NUMERICAL  ITEMS BEFØRE, IT MIGHT BE WISE TØ GØ ØVER THE
INSTRUCTIØNS FØR NUMERICAL  ITEMS.

IF YØU WØULD LIKE TØ GØ ØVER THE INSTRUCTIØNS, ENTER "YES" AFTER
THE QUESTIØN MARK APPEARS; IF NØT, ENTER "NØ".

IN EITHER CASE, WAIT FØR THE QUESTIØN MARK, ENTER YØUR RESPØNSE
AND PRESS THE RETURN KEY.
```

Screen 1 RN. Similar to 1RV except for number series items.



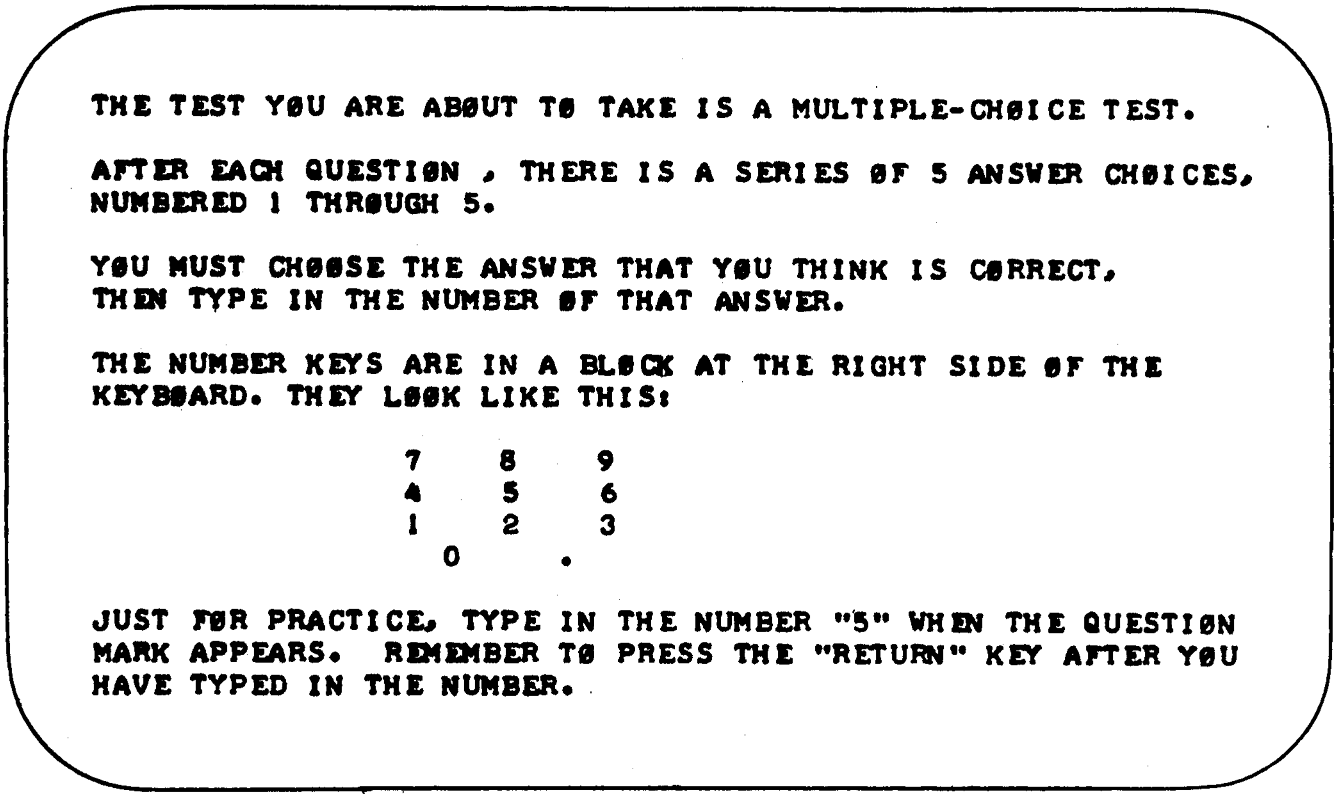
YØU SEEM TØ BE HAVING TRØUBLE WITH THE INSTRUCTIØNS
ØR THE EQUIPMENT. PLEASE CALL THE TEST PRØCTØR. HE WILL PERSØNALLY
HELP YØU LEARN TØ ØPERATE THE KEYBØARD.

Screen n$_1$ER. Error screen summoning proctor for the first time. It can be presented after any error screen is responded to incorrectly. After this screen is presented, the proctor can have the screen on which the error occured displayed again.



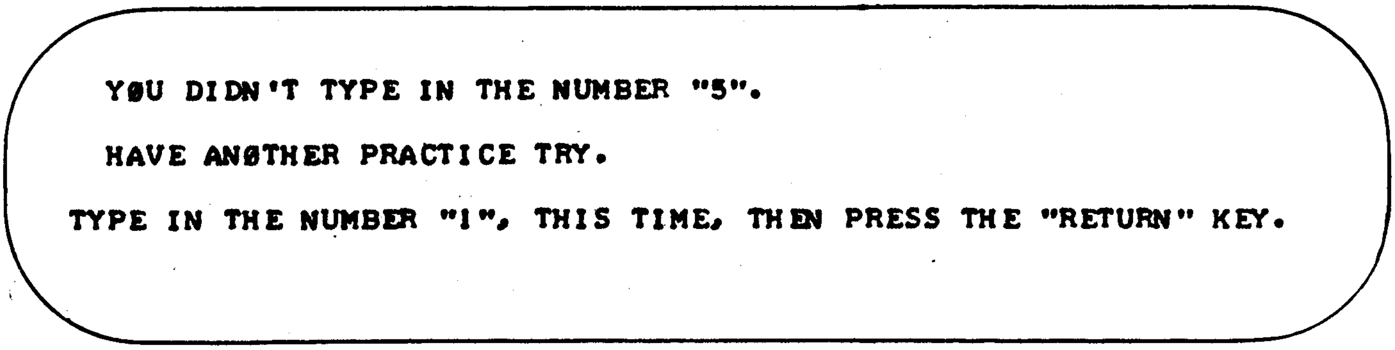
YØU STILL SEEM TØ BE HAVING TRØUBLE WITH THE INSTRUCTIØNS
ØR THE EQUIPMENT. PLEASE CALL THE TEST PRØCTØR AGAIN.
HE WILL PERSØNALLY HELP YØU LEARN TØ ØPERATE THE KEYBØARD.

Screen n$_2$ER. A variation on screen n$_1$ER. This screen is presented instead of screen n$_1$ER after that screen has been presented once.



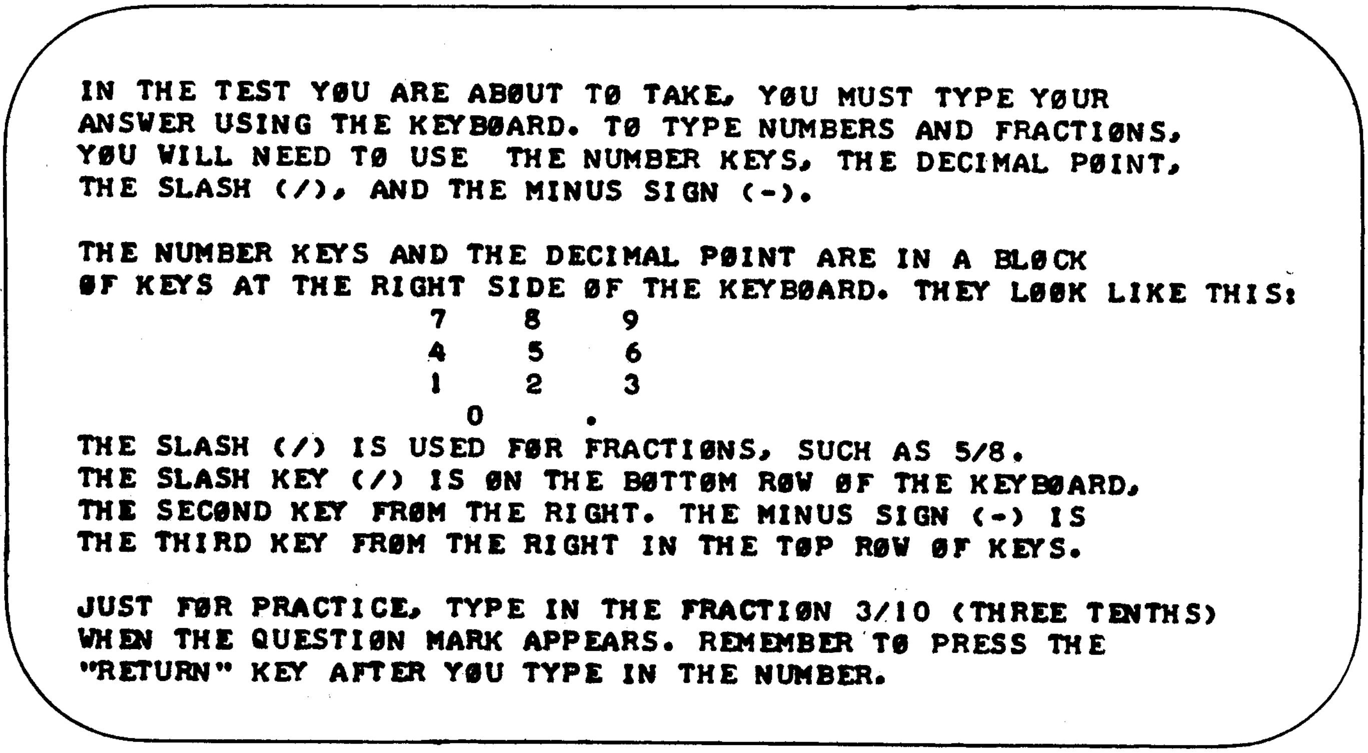
```
THE TEST YØU ARE ABØUT TØ TAKE IS A MULTIPLE-CHØICE TEST.

AFTER EACH QUESTIØN , THERE IS A SERIES ØF 5 ANSWER CHØICES,
NUMBERED 1 THRØUGH 5.

YØU MUST CHØØSE THE ANSWER THAT YØU THINK IS CØRRECT,
THEN TYPE IN THE NUMBER ØF THAT ANSWER.

THE NUMBER KEYS ARE IN A BLØCK AT THE RIGHT SIDE ØF THE
KEYBØARD. THEY LØØK LIKE THIS:

                    7    8    9
                    4    5    6
                    1    2    3
                      0     .

JUST FØR PRACTICE, TYPE IN THE NUMBER "5" WHEN THE QUESTIØN
MARK APPEARS.  REMEMBER TØ PRESS THE "RETURN" KEY AFTER YØU
HAVE TYPED IN THE NUMBER.
```

Screen 2V. Second instructional screen for vocabulary items.



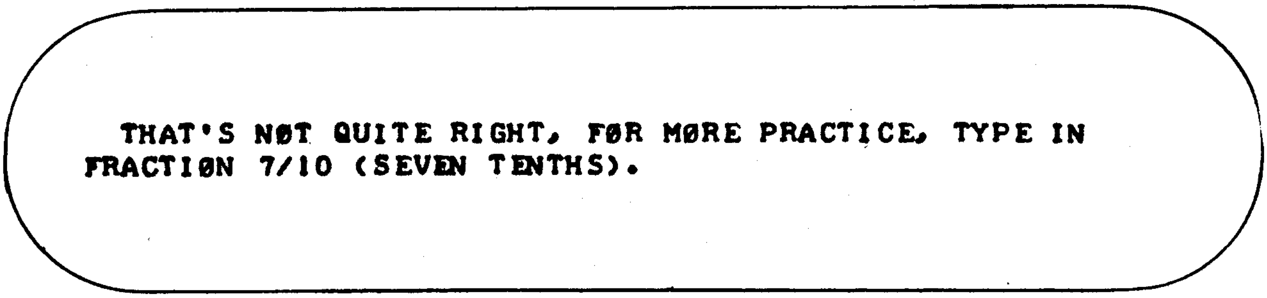
```
  YØU DIDN'T TYPE IN THE NUMBER "5".

  HAVE ANØTHER PRACTICE TRY.

TYPE IN THE NUMBER "1", THIS TIME, THEN PRESS THE "RETURN" KEY.
```

Screen 2V ER. Error screen if screen 2V responded to incorrectly.



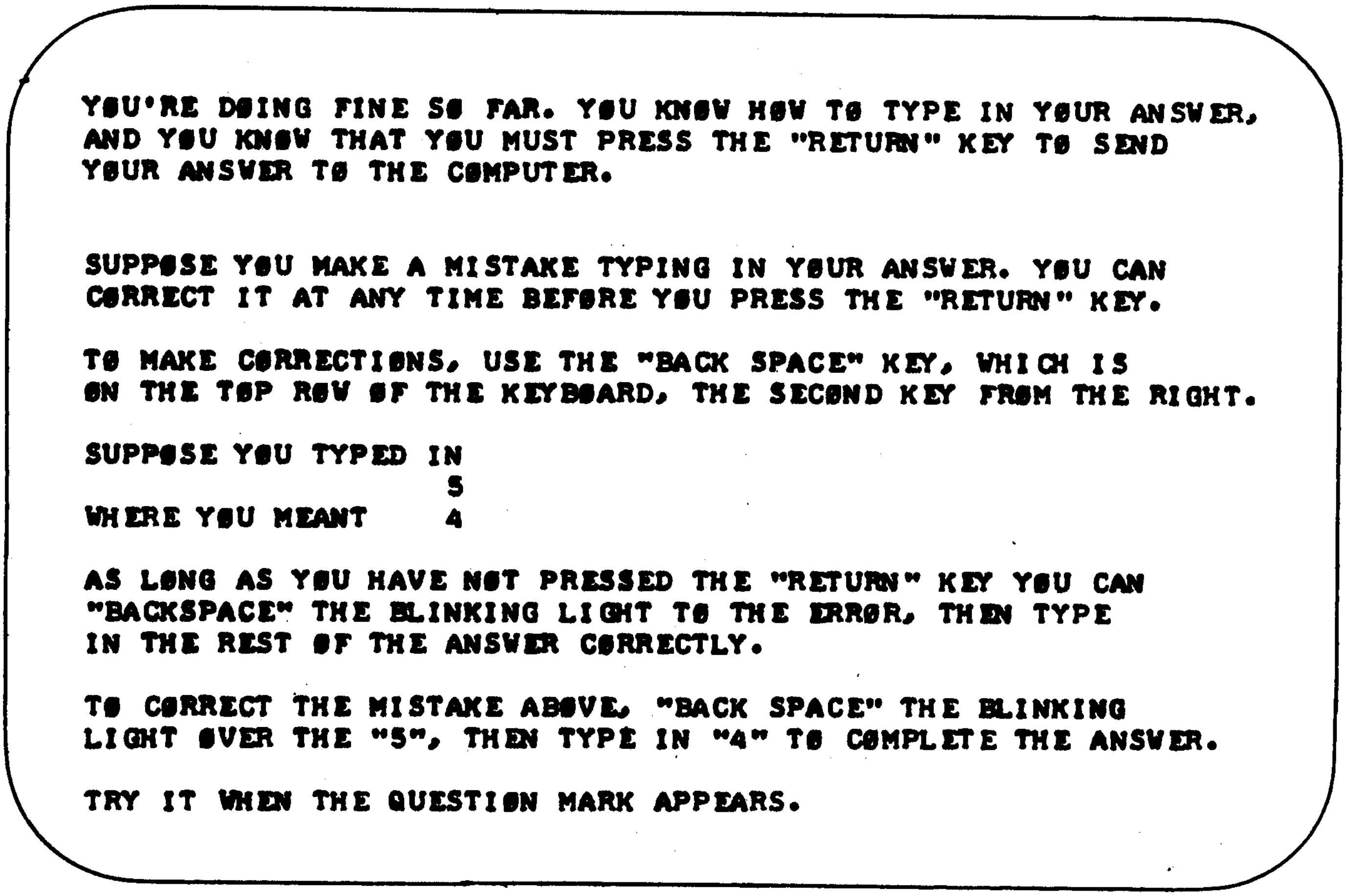
```
IN THE TEST YØU ARE ABØUT TØ TAKE, YØU MUST TYPE YØUR
ANSWER USING THE KEYBØARD. TØ TYPE NUMBERS AND FRACTIØNS,
YØU WILL NEED TØ USE  THE NUMBER KEYS, THE DECIMAL PØINT,
THE SLASH (/), AND THE MINUS SIGN (-).

THE NUMBER KEYS AND THE DECIMAL PØINT ARE IN A BLØCK
ØF KEYS AT THE RIGHT SIDE ØF THE KEYBØARD. THEY LØØK LIKE THIS:
                  7    8    9
                  4    5    6
                  1    2    3
                    0    .
THE SLASH (/) IS USED FØR FRACTIØNS, SUCH AS 5/8.
THE SLASH KEY (/) IS ØN THE BØTTØM RØW ØF THE KEYBØARD,
THE SECØND KEY FRØM THE RIGHT. THE MINUS SIGN (-) IS
THE THIRD KEY FRØM THE RIGHT IN THE TØP RØW ØF KEYS.

JUST FØR PRACTICE, TYPE IN THE FRACTIØN 3/10 (THREE TENTHS)
WHEN THE QUESTIØN MARK APPEARS. REMEMBER TØ PRESS THE
"RETURN" KEY AFTER YØU TYPE IN THE NUMBER.
```

Screen 2N. Second instructional screen for number series items.



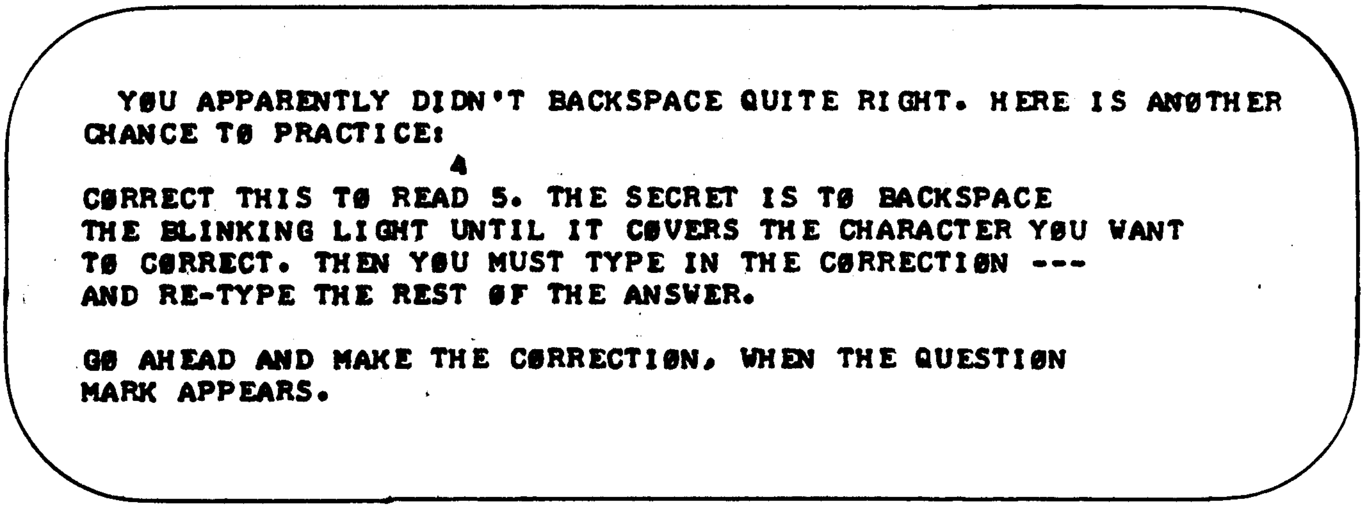
```
  THAT'S NØT QUITE RIGHT, FØR MØRE PRACTICE, TYPE IN
FRACTIØN 7/10 (SEVEN TENTHS).
```

Screen 2N ER. Error screen if screen 2N responded to incorrectly.



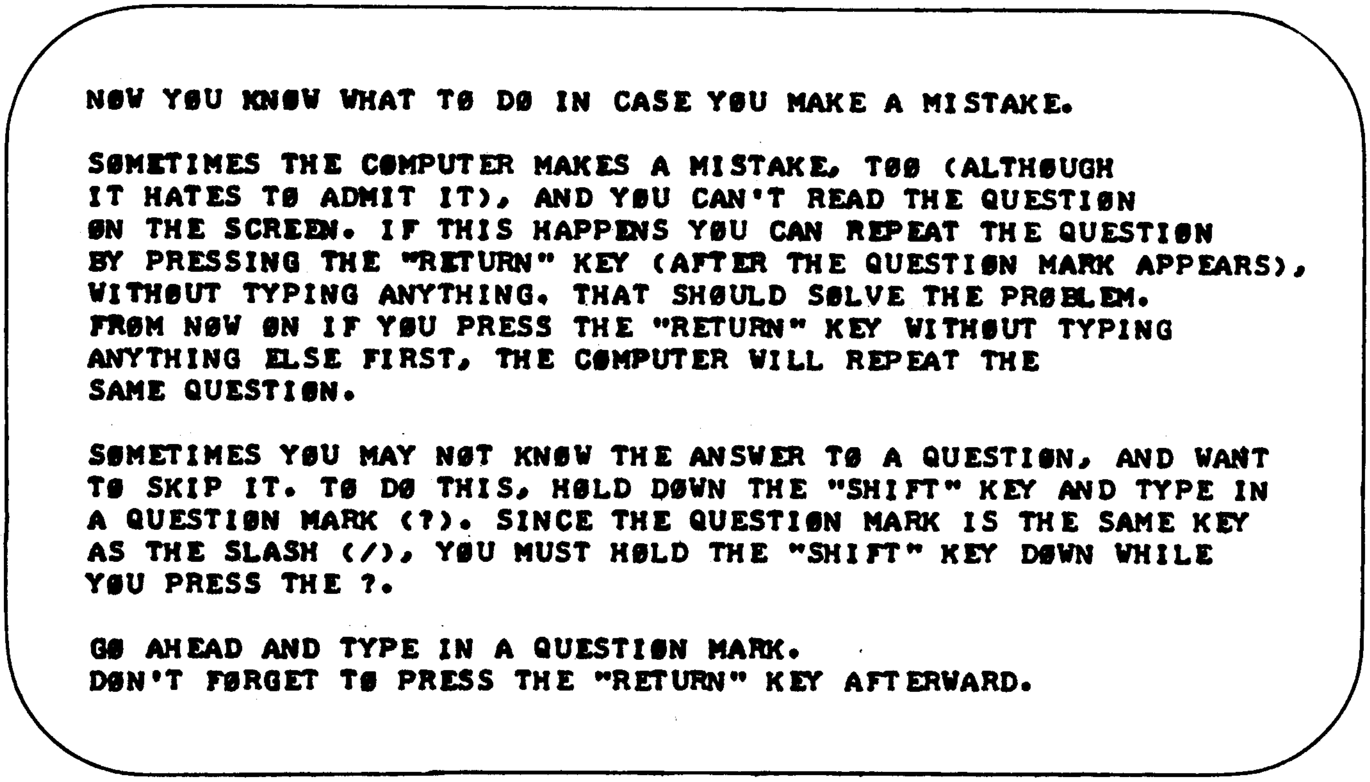
```
YØU'RE DØING FINE SØ FAR. YØU KNØW HØW TØ TYPE IN YØUR ANSWER,
AND YØU KNØW THAT YØU MUST PRESS THE "RETURN" KEY TØ SEND
YØUR ANSWER TØ THE CØMPUTER.


SUPPØSE YØU MAKE A MISTAKE TYPING IN YØUR ANSWER. YØU CAN
CØRRECT IT AT ANY TIME BEFØRE YØU PRESS THE "RETURN" KEY.

TØ MAKE CØRRECTIØNS, USE THE "BACK SPACE" KEY, WHICH IS
ØN THE TØP RØW ØF THE KEYBØARD, THE SECØND KEY FRØM THE RIGHT.

SUPPØSE YØU TYPED IN
                    5
WHERE YØU MEANT     4

AS LØNG AS YØU HAVE NØT PRESSED THE "RETURN" KEY YØU CAN
"BACKSPACE" THE BLINKING LIGHT TØ THE ERRØR, THEN TYPE
IN THE REST ØF THE ANSWER CØRRECTLY.

TØ CØRRECT THE MISTAKE ABØVE, "BACK SPACE" THE BLINKING
LIGHT ØVER THE "5", THEN TYPE IN "4" TØ CØMPLETE THE ANSWER.

TRY IT WHEN THE QUESTIØN MARK APPEARS.
```

Screen 3. Third instructional screen.



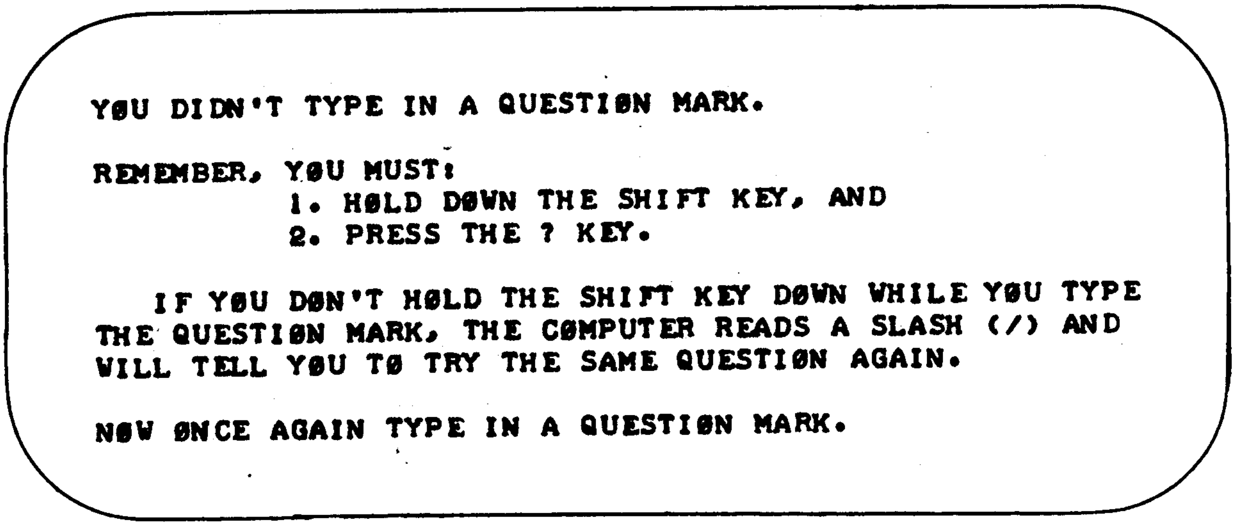
```
 YØU APPARENTLY DIDN'T BACKSPACE QUITE RIGHT. HERE IS ANØTHER
CHANCE TØ PRACTICE:
                    4
CØRRECT THIS TØ READ 5. THE SECRET IS TØ BACKSPACE
THE BLINKING LIGHT UNTIL IT CØVERS THE CHARACTER YØU WANT
TØ CØRRECT. THEN YØU MUST TYPE IN THE CØRRECTIØN ---
AND RE-TYPE THE REST ØF THE ANSWER.

GØ AHEAD AND MAKE THE CØRRECTIØN, WHEN THE QUESTIØN
MARK APPEARS.
```

Screen 3 ER. Screen presented if screen 3 is answered incorrectly.



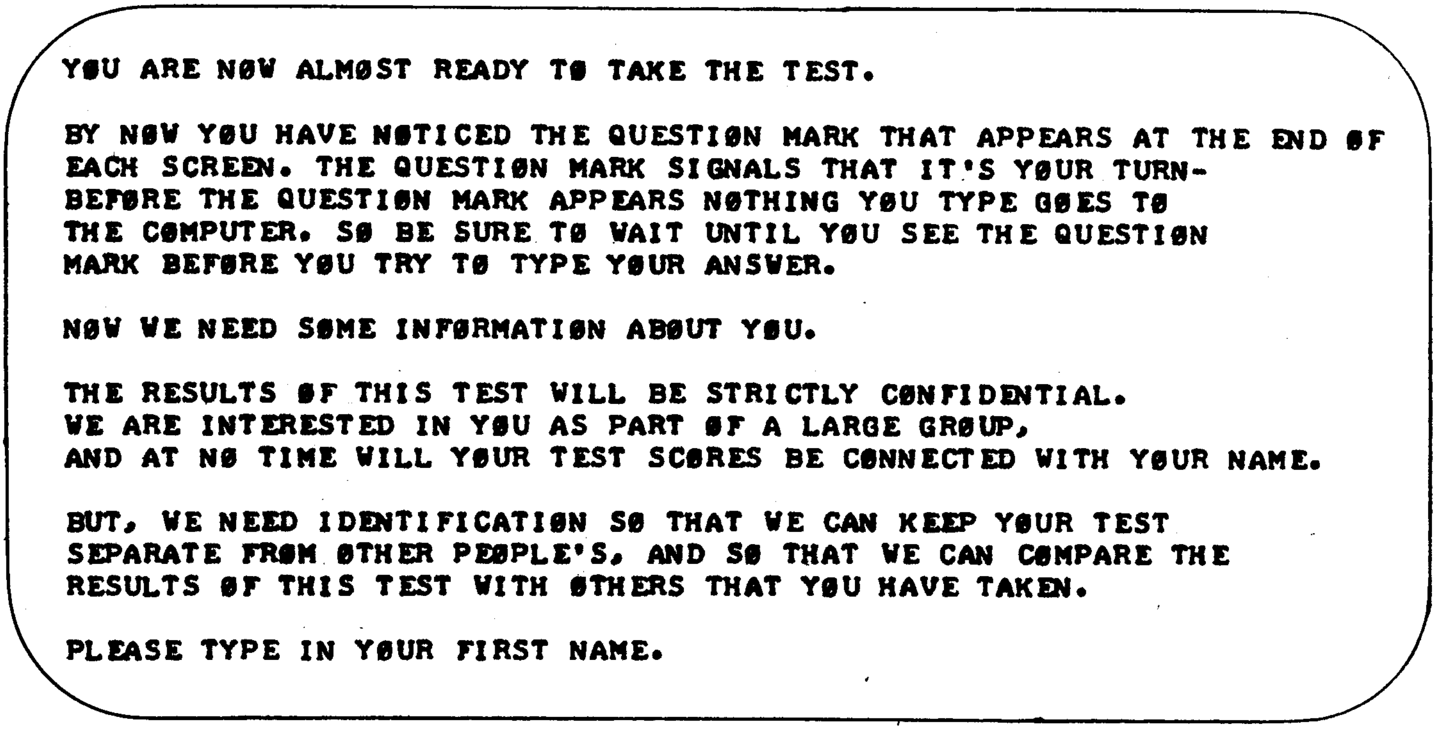
```
NØW YØU KNØW WHAT TØ DØ IN CASE YØU MAKE A MISTAKE.

SØMETIMES THE CØMPUTER MAKES A MISTAKE, TØØ (ALTHØUGH
IT HATES TØ ADMIT IT), AND YØU CAN'T READ THE QUESTIØN
ØN THE SCREEN. IF THIS HAPPENS YØU CAN REPEAT THE QUESTIØN
BY PRESSING THE "RETURN" KEY (AFTER THE QUESTIØN MARK APPEARS),
WITHØUT TYPING ANYTHING. THAT SHØULD SØLVE THE PRØBLEM.
FRØM NØW ØN IF YØU PRESS THE "RETURN" KEY WITHØUT TYPING
ANYTHING ELSE FIRST, THE CØMPUTER WILL REPEAT THE
SAME QUESTIØN.

SØMETIMES YØU MAY NØT KNØW THE ANSWER TØ A QUESTIØN, AND WANT
TØ SKIP IT. TØ DØ THIS, HØLD DØWN THE "SHIFT" KEY AND TYPE IN
A QUESTIØN MARK (?). SINCE THE QUESTIØN MARK IS THE SAME KEY
AS THE SLASH (/), YØU MUST HØLD THE "SHIFT" KEY DØWN WHILE
YØU PRESS THE ?.

GØ AHEAD AND TYPE IN A QUESTIØN MARK.
DØN'T FØRGET TØ PRESS THE "RETURN" KEY AFTERWARD.
```

Screen 4. Fourth instructional screen.



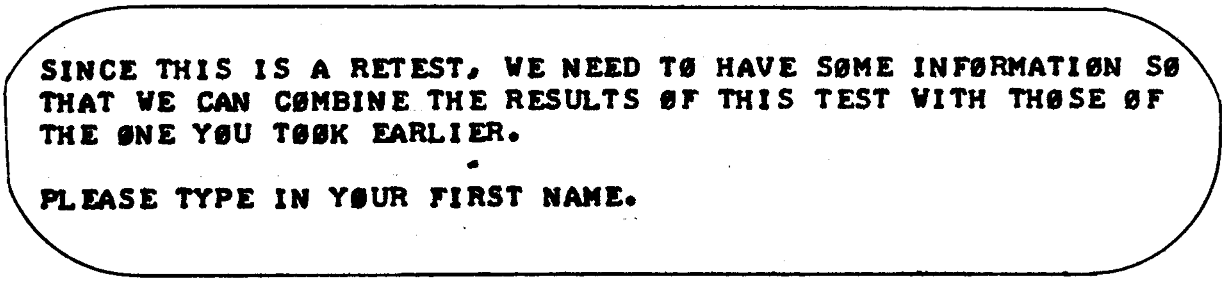
```
YØU DIDN'T TYPE IN A QUESTIØN MARK.

REMEMBER, YØU MUST:
            1. HØLD DØWN THE SHIFT KEY, AND
            2. PRESS THE ? KEY.

   IF YØU DØN'T HØLD THE SHIFT KEY DØWN WHILE YØU TYPE
THE QUESTIØN MARK, THE CØMPUTER READS A SLASH (/) AND
WILL TELL YØU TØ TRY THE SAME QUESTIØN AGAIN.

NØW ØNCE AGAIN TYPE IN A QUESTIØN MARK.
```

Screen 4 ER. Screen presented if screen 4 is answered incorrectly.



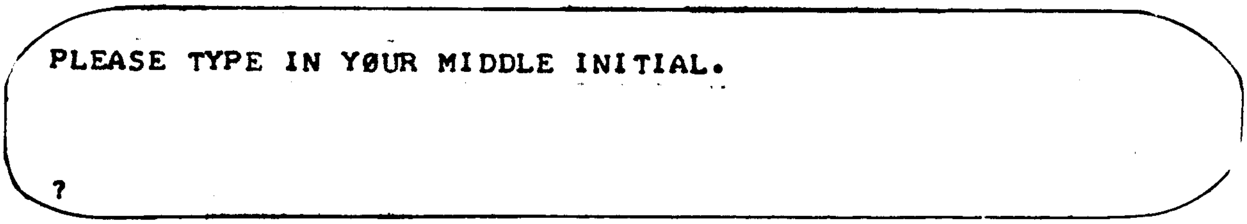
```
YØU ARE NØW ALMØST READY TØ TAKE THE TEST.

BY NØW YØU HAVE NØTICED THE QUESTIØN MARK THAT APPEARS AT THE END ØF
EACH SCREEN. THE QUESTIØN MARK SIGNALS THAT IT'S YØUR TURN-
BEFØRE THE QUESTIØN MARK APPEARS NØTHING YØU TYPE GØES TØ
THE CØMPUTER. SØ BE SURE TØ WAIT UNTIL YØU SEE THE QUESTIØN
MARK BEFØRE YØU TRY TØ TYPE YØUR ANSWER.

NØW WE NEED SØME INFØRMATIØN ABØUT YØU.

THE RESULTS ØF THIS TEST WILL BE STRICTLY CØNFIDENTIAL.
WE ARE INTERESTED IN YØU AS PART ØF A LARGE GRØUP,
AND AT NØ TIME WILL YØUR TEST SCØRES BE CØNNECTED WITH YØUR NAME.

BUT, WE NEED IDENTIFICATIØN SØ THAT WE CAN KEEP YØUR TEST
SEPARATE FRØM ØTHER PEØPLE'S, AND SØ THAT WE CAN CØMPARE THE
RESULTS ØF THIS TEST WITH ØTHERS THAT YØU HAVE TAKEN.

PLEASE TYPE IN YØUR FIRST NAME.
```

Screen 5. Fifth instructional screen.



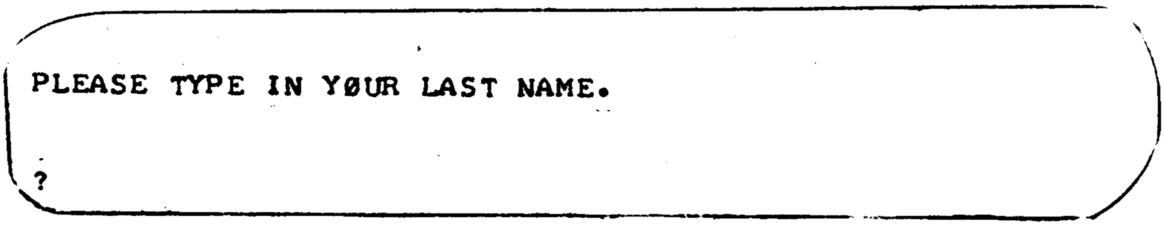
```
SINCE THIS IS A RETEST, WE NEED TØ HAVE SØME INFØRMATIØN SØ
THAT WE CAN CØMBINE THE RESULTS ØF THIS TEST WITH THØSE ØF
THE ØNE YØU TØØK EARLIER.

PLEASE TYPE IN YØUR FIRST NAME.
```

Screen 5R. Screen presented if retest is being taken, and testee did not request instructions.



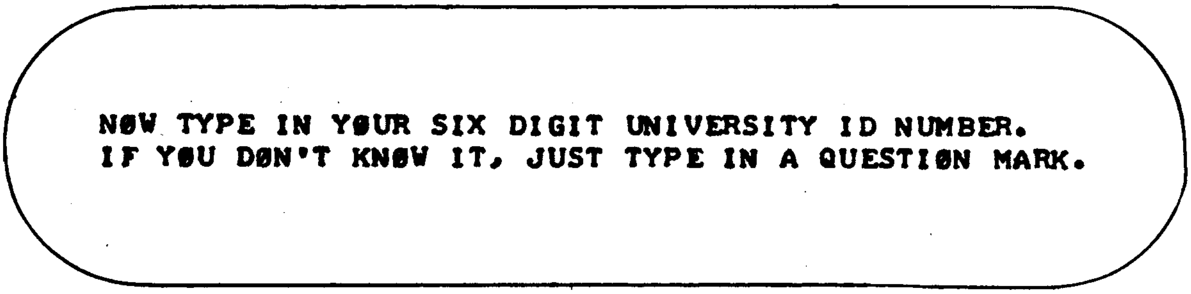
```
PLEASE TYPE IN YØUR MIDDLE INITIAL.


?
```

Screen 6. Sixth instructional screen.



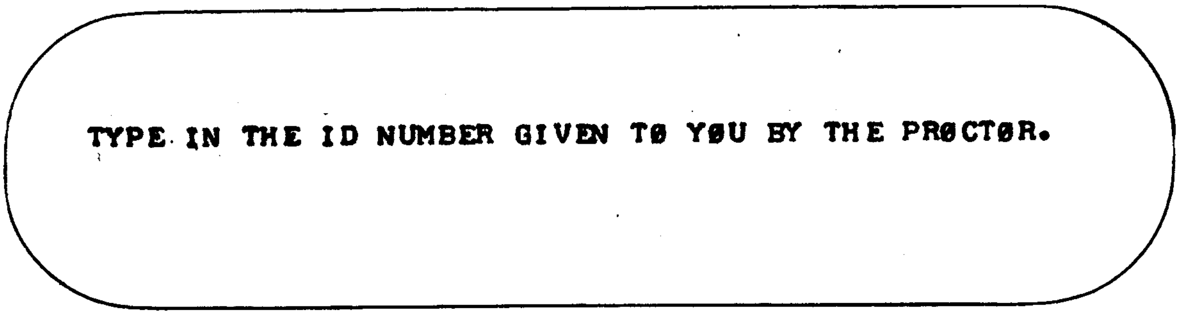
```
PLEASE TYPE IN YØUR LAST NAME.


?
```

Screen 7. Seventh instructional screen.



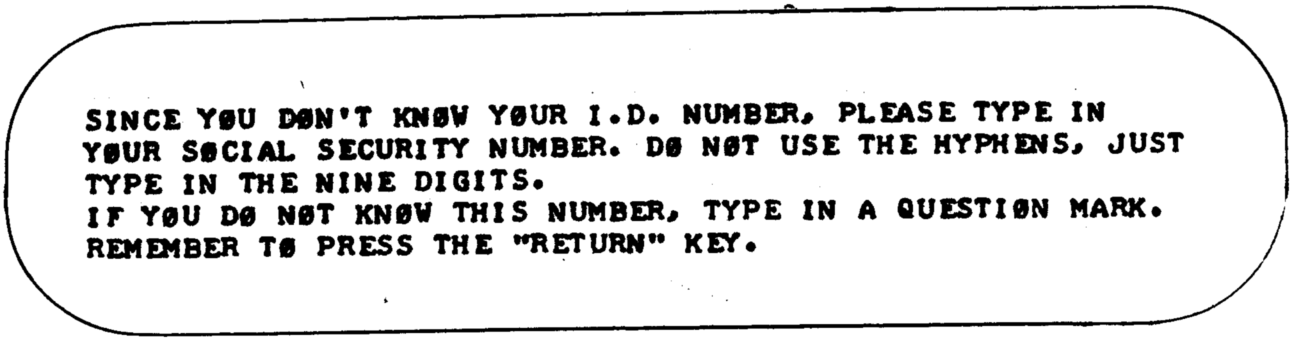
NØW TYPE IN YØUR SIX DIGIT UNIVERSITY ID NUMBER.
IF YØU DØN'T KNØW IT, JUST TYPE IN A QUESTIØN MARK.

Screen 8. Eighth instructional screen.



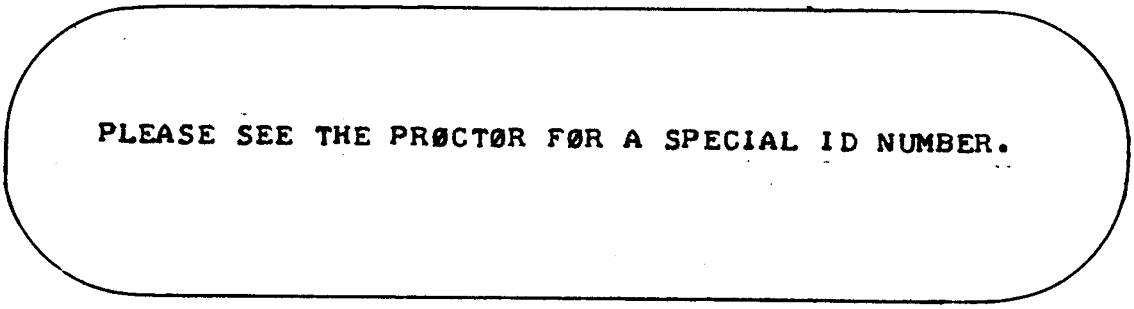
TYPE IN THE ID NUMBER GIVEN TØ YØU BY THE PRØCTØR.

Screen 8 HS. Version of instructional screen 8, presented to high school students.



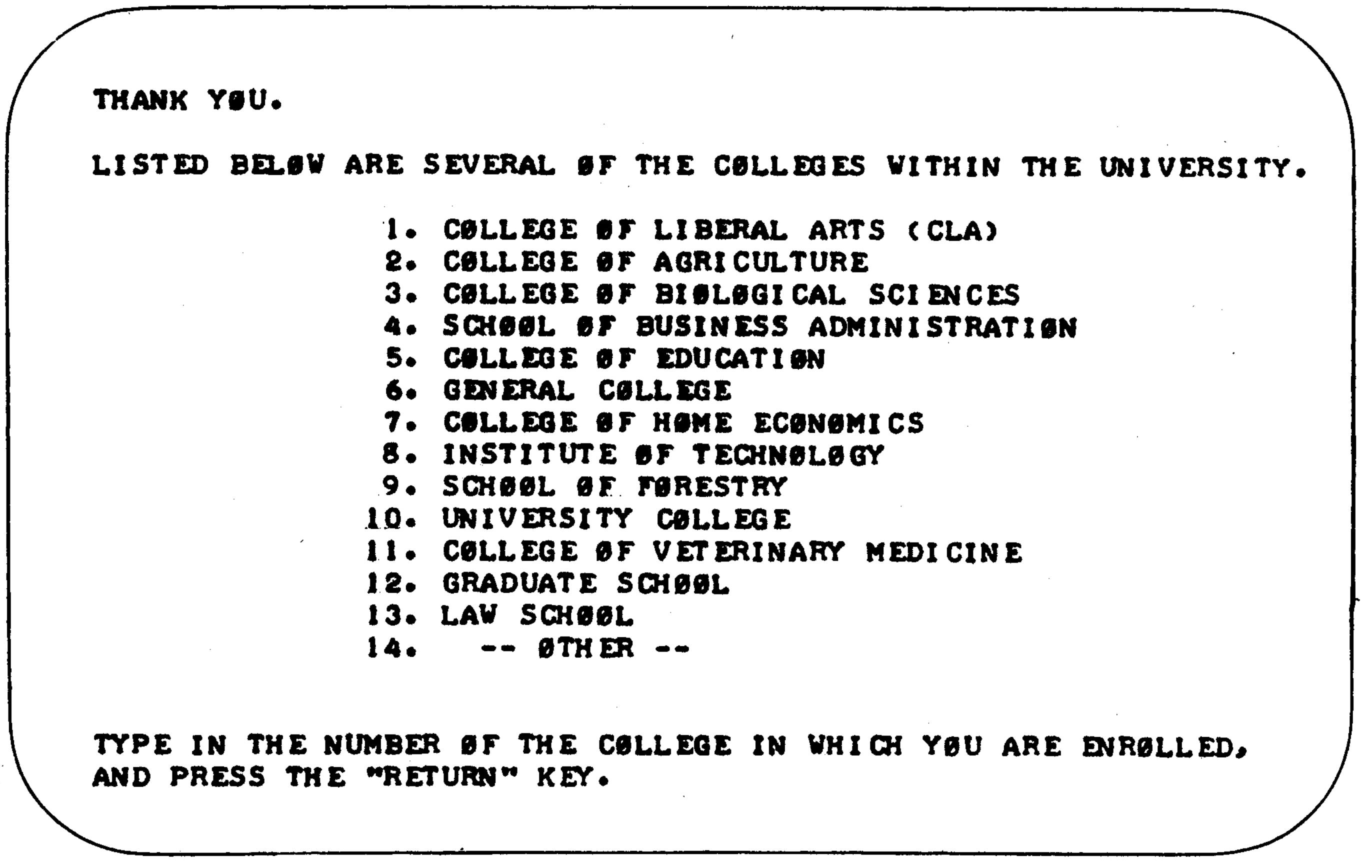
SINCE YØU DØN'T KNØW YØUR I.D. NUMBER, PLEASE TYPE IN
YØUR SØCIAL SECURITY NUMBER. DØ NØT USE THE HYPHENS, JUST
TYPE IN THE NINE DIGITS.
IF YØU DØ NØT KNØW THIS NUMBER, TYPE IN A QUESTIØN MARK.
REMEMBER TØ PRESS THE "RETURN" KEY.

Screen 8.1 ER. Screen presented if subject does not know I.D. number in screen 8.



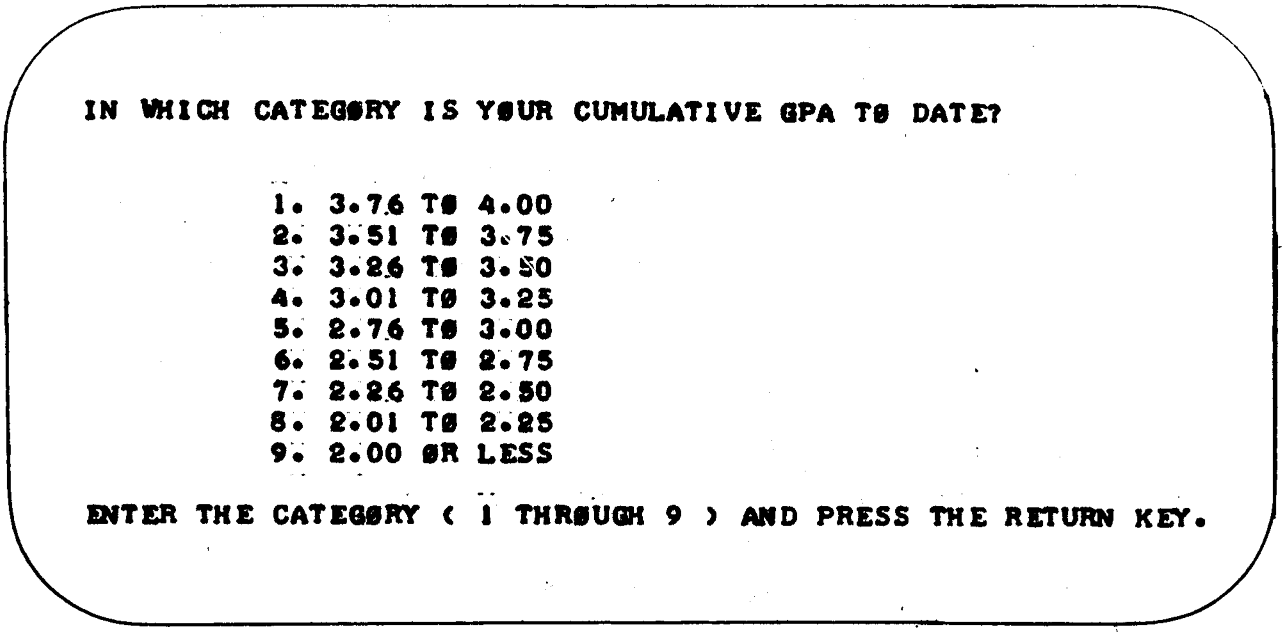
```
PLEASE SEE THE PRØCTØR FØR A SPECIAL ID NUMBER.
```

Screen 8.2 ER. Screen presented if testee knows neither I.D. nor Social Security number.



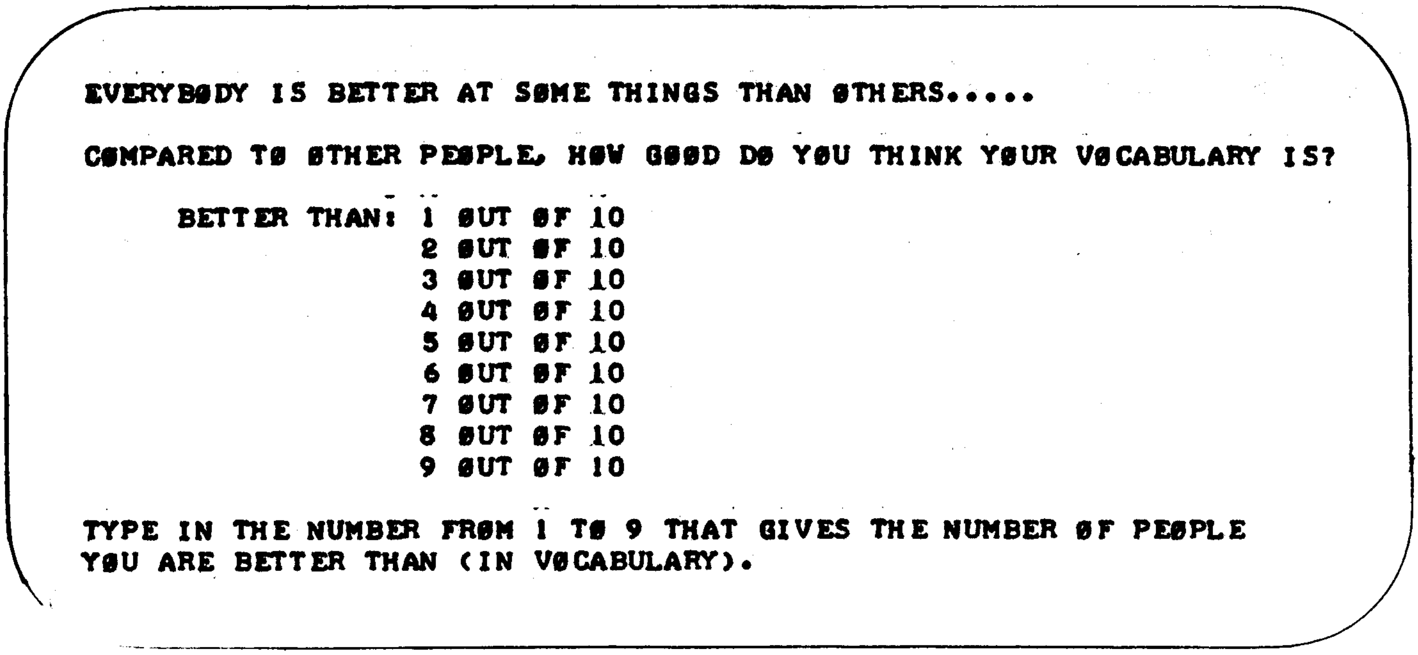
```
THANK YØU.

LISTED BELØW ARE SEVERAL ØF THE CØLLEGES WITHIN THE UNIVERSITY.

                 1. CØLLEGE ØF LIBERAL ARTS (CLA)
                 2. CØLLEGE ØF AGRICULTURE
                 3. CØLLEGE ØF BIØLØGICAL SCIENCES
                 4. SCHØØL ØF BUSINESS ADMINISTRATIØN
                 5. CØLLEGE ØF EDUCATIØN
                 6. GENERAL CØLLEGE
                 7. CØLLEGE ØF HØME ECØNØMICS
                 8. INSTITUTE ØF TECHNØLØGY
                 9. SCHØØL ØF FØRESTRY
                10. UNIVERSITY CØLLEGE
                11. CØLLEGE ØF VETERINARY MEDICINE
                12. GRADUATE SCHØØL
                13. LAW SCHØØL
                14.   -- ØTHER --


TYPE IN THE NUMBER ØF THE CØLLEGE IN WHICH YØU ARE ENRØLLED,
AND PRESS THE "RETURN" KEY.
```

Screen 8.3 ER. Screen requesting student's college after social security number or proctor assigned number is obtained following unsuccessful request for I.D. number.



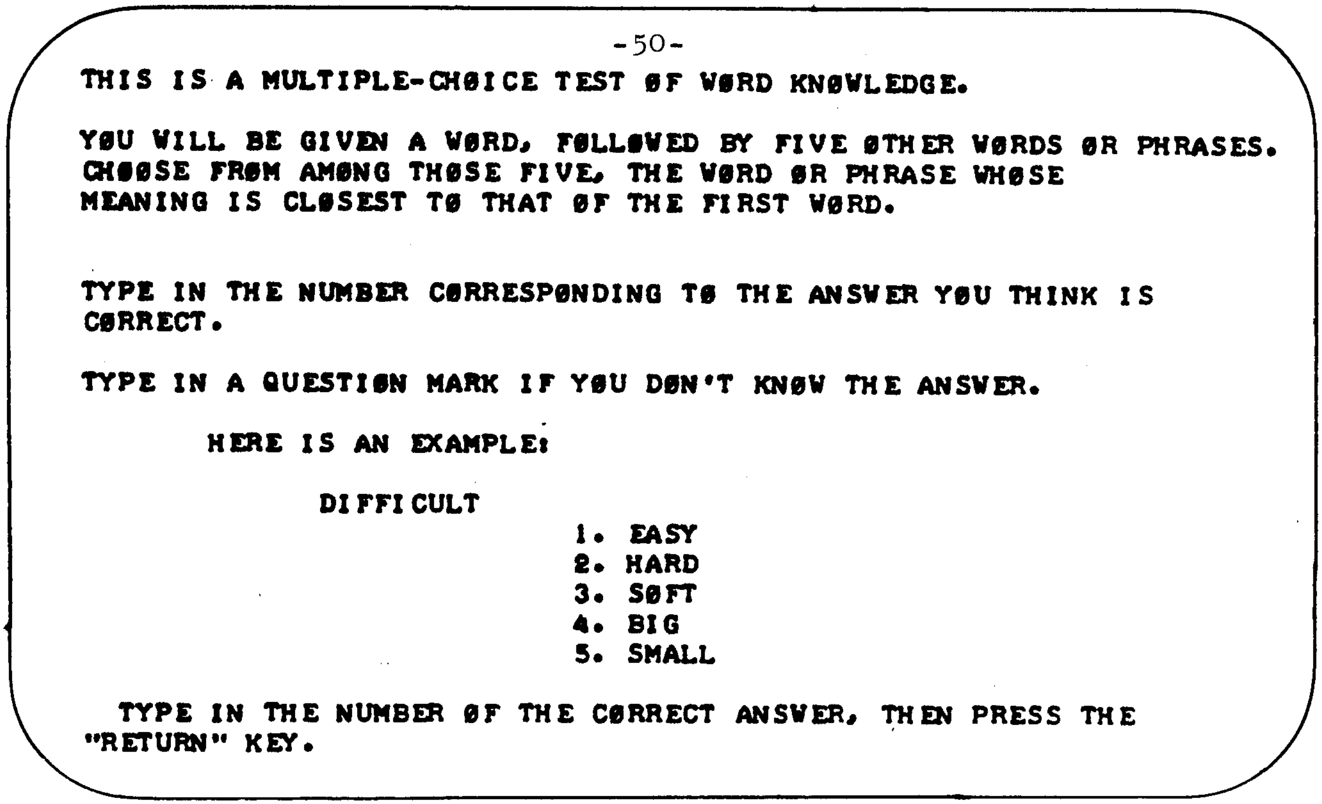
```
IN WHICH CATEGØRY IS YØUR CUMULATIVE GPA TØ DATE?

              1. 3.76 TØ 4.00
              2. 3.51 TØ 3.75
              3. 3.26 TØ 3.50
              4. 3.01 TØ 3.25
              5. 2.76 TØ 3.00
              6. 2.51 TØ 2.75
              7. 2.26 TØ 2.50
              8. 2.01 TØ 2.25
              9. 2.00 ØR LESS

ENTER THE CATEGØRY ( 1 THRØUGH 9 ) AND PRESS THE RETURN KEY.
```

Screen 8AB. Screen presented to university students requesting an ability estimate if a stradaptive or Bayesian test is to be given.



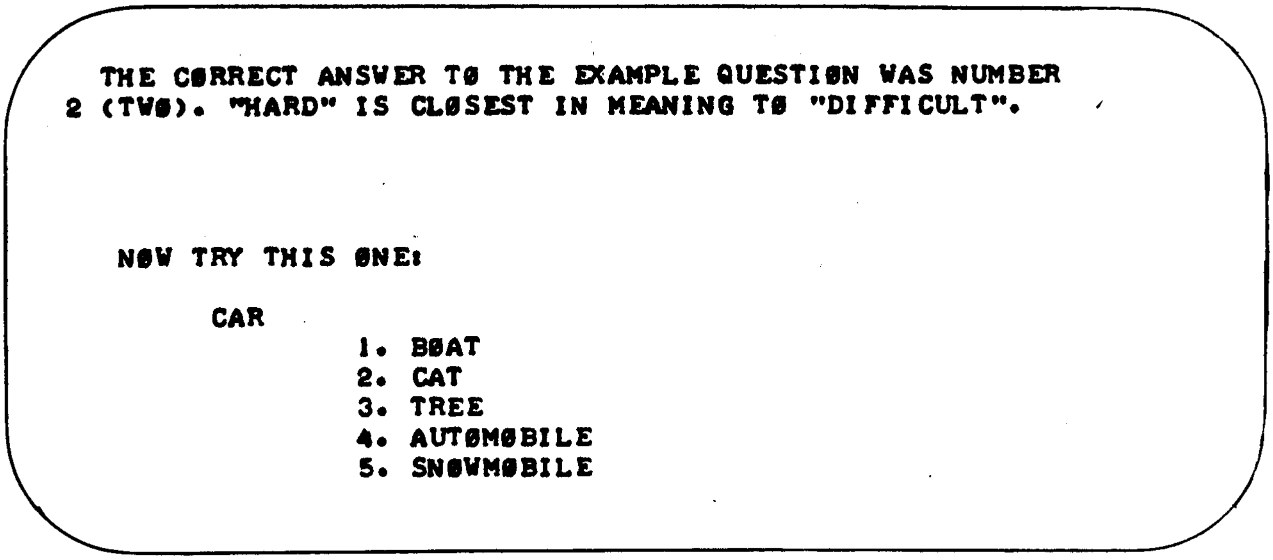
```
EVERYBØDY IS BETTER AT SØME THINGS THAN ØTHERS.....

CØMPARED TØ ØTHER PEØPLE, HØW GØØD DØ YØU THINK YØUR VØCABULARY IS?

     BETTER THAN: 1 ØUT ØF 10
                  2 ØUT ØF 10
                  3 ØUT ØF 10
                  4 ØUT ØF 10
                  5 ØUT ØF 10
                  6 ØUT ØF 10
                  7 ØUT ØF 10
                  8 ØUT ØF 10
                  9 ØUT ØF 10

TYPE IN THE NUMBER FRØM 1 TØ 9 THAT GIVES THE NUMBER ØF PEØPLE
YØU ARE BETTER THAN (IN VØCABULARY).
```

Screen 8AB HS. Screen presented to high school students requesting an ability estimate if a stradaptive or Bayesian test is to be given.



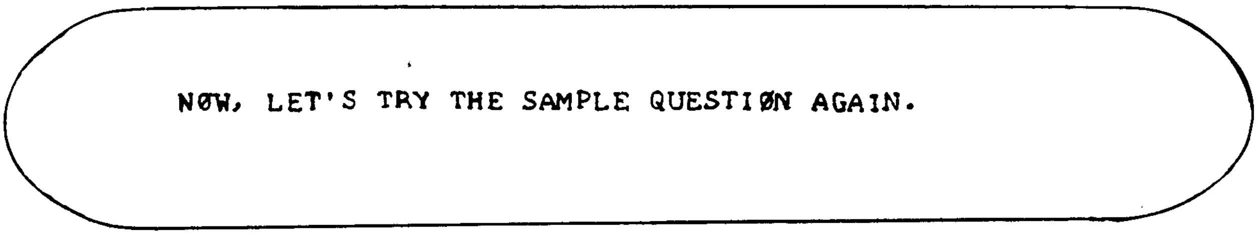
```
THIS IS A MULTIPLE-CHØICE TEST ØF WØRD KNØWLEDGE.

YØU WILL BE GIVEN A WØRD, FØLLØWED BY FIVE ØTHER WØRDS ØR PHRASES.
CHØØSE FRØM AMØNG THØSE FIVE, THE WØRD ØR PHRASE WHØSE
MEANING IS CLØSEST TØ THAT ØF THE FIRST WØRD.


TYPE IN THE NUMBER CØRRESPØNDING TØ THE ANSWER YØU THINK IS
CØRRECT.

TYPE IN A QUESTIØN MARK IF YØU DØN'T KNØW THE ANSWER.

        HERE IS AN EXAMPLE:

               DIFFICULT
                                  1. EASY
                                  2. HARD
                                  3. SØFT
                                  4. BIG
                                  5. SMALL

  TYPE IN THE NUMBER ØF THE CØRRECT ANSWER, THEN PRESS THE
"RETURN" KEY.
```

Screen 9V. Ninth instructional screen for vocabulary items.



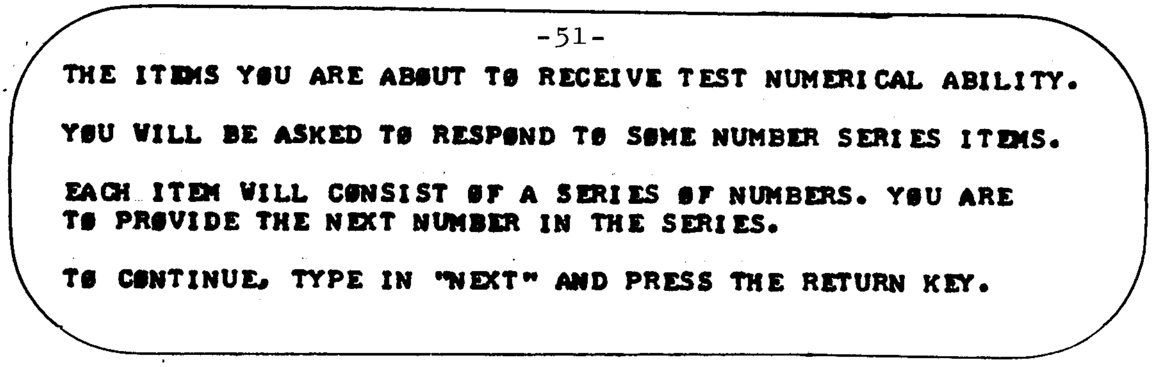
```
  THE CØRRECT ANSWER TØ THE EXAMPLE QUESTIØN WAS NUMBER
2 (TWØ). "HARD" IS CLØSEST IN MEANING TØ "DIFFICULT".



   NØW TRY THIS ØNE:

       CAR
              1. BØAT
              2. CAT
              3. TREE
              4. AUTØMØBILE
              5. SNØWMØBILE
```

Screen 9a VER. Screen presented if screen 9V was answered incorrectly.



```
       NØW, LET'S TRY THE SAMPLE QUESTIØN AGAIN.
```

Screen 9b VER. Screen presented following 9a VER. This screen continues with screen 9V.



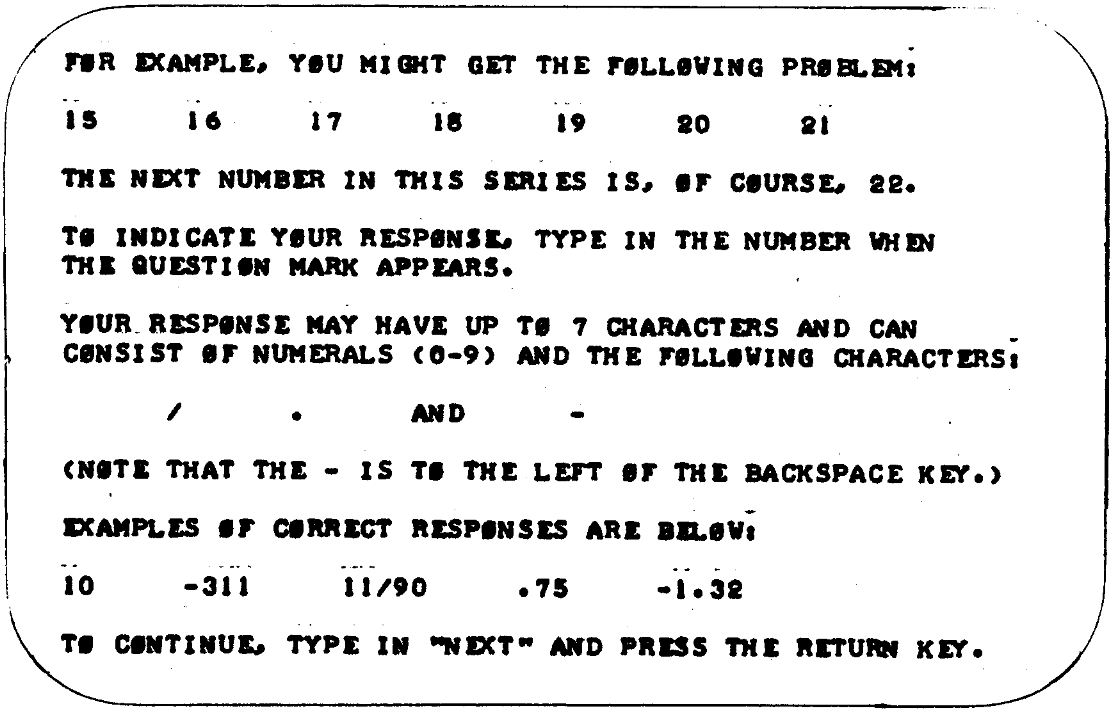
THE ITEMS YØU ARE ABØUT TØ RECEIVE TEST NUMERICAL ABILITY.

YØU WILL BE ASKED TØ RESPØND TØ SØME NUMBER SERIES ITEMS.

EACH ITEM WILL CØNSIST ØF A SERIES ØF NUMBERS. YØU ARE TØ PRØVIDE THE NEXT NUMBER IN THE SERIES.

TØ CØNTINUE, TYPE IN "NEXT" AND PRESS THE RETURN KEY.

Screen 9N. Ninth instructional screen for numeric items.



FØR EXAMPLE, YØU MIGHT GET THE FØLLØWING PRØBLEM:

15 16 17 18 19 20 21

THE NEXT NUMBER IN THIS SERIES IS, ØF CØURSE, 22.

TØ INDICATE YØUR RESPØNSE, TYPE IN THE NUMBER WHEN THE QUESTIØN MARK APPEARS.

YØUR RESPØNSE MAY HAVE UP TØ 7 CHARACTERS AND CAN CØNSIST ØF NUMERALS (0-9) AND THE FØLLØWING CHARACTERS:

/ . AND -

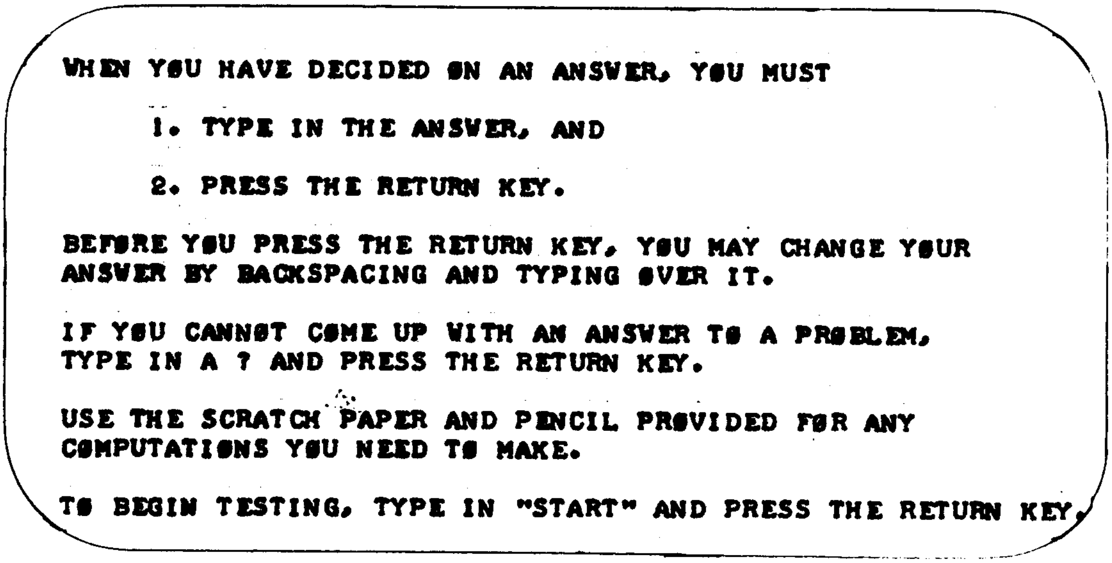
(NØTE THAT THE - IS TØ THE LEFT ØF THE BACKSPACE KEY.)

EXAMPLES ØF CØRRECT RESPØNSES ARE BELØW:

10 -311 11/90 .75 -1.32

TØ CØNTINUE, TYPE IN "NEXT" AND PRESS THE RETURN KEY.

Screen 10N. Tenth instructional screen for numeric items.



WHEN YØU HAVE DECIDED ØN AN ANSWER, YØU MUST

1. TYPE IN THE ANSWER, AND

2. PRESS THE RETURN KEY.

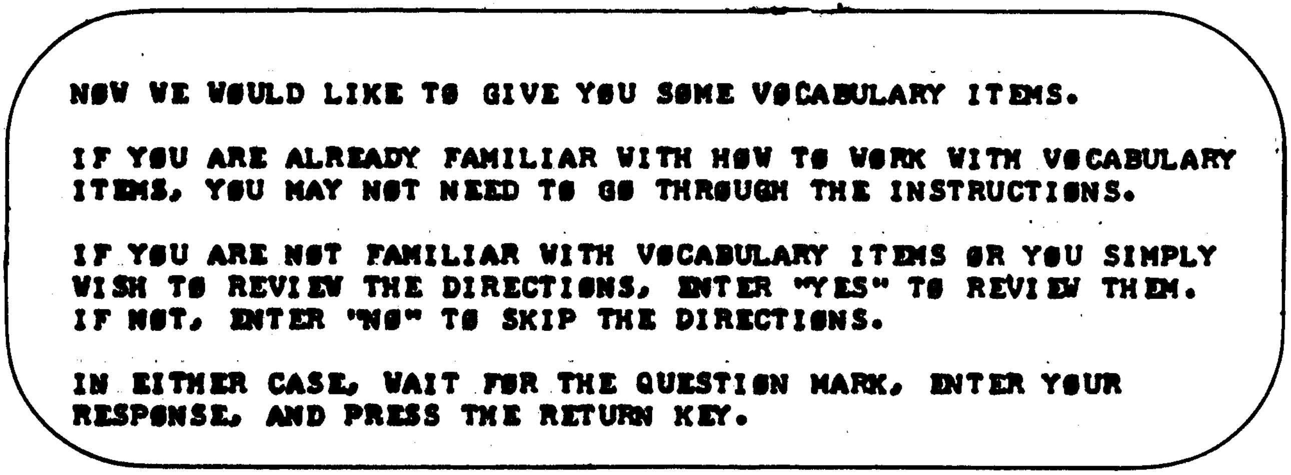
BEFØRE YØU PRESS THE RETURN KEY, YØU MAY CHANGE YØUR ANSWER BY BACKSPACING AND TYPING ØVER IT.

IF YØU CANNØT CØME UP WITH AN ANSWER TØ A PRØBLEM, TYPE IN A ? AND PRESS THE RETURN KEY.

USE THE SCRATCH PAPER AND PENCIL PRØVIDED FØR ANY CØMPUTATIØNS YØU NEED TØ MAKE.

TØ BEGIN TESTING, TYPE IN "START" AND PRESS THE RETURN KEY.

Screen 11N. Eleventh instructional screen for numeric items.



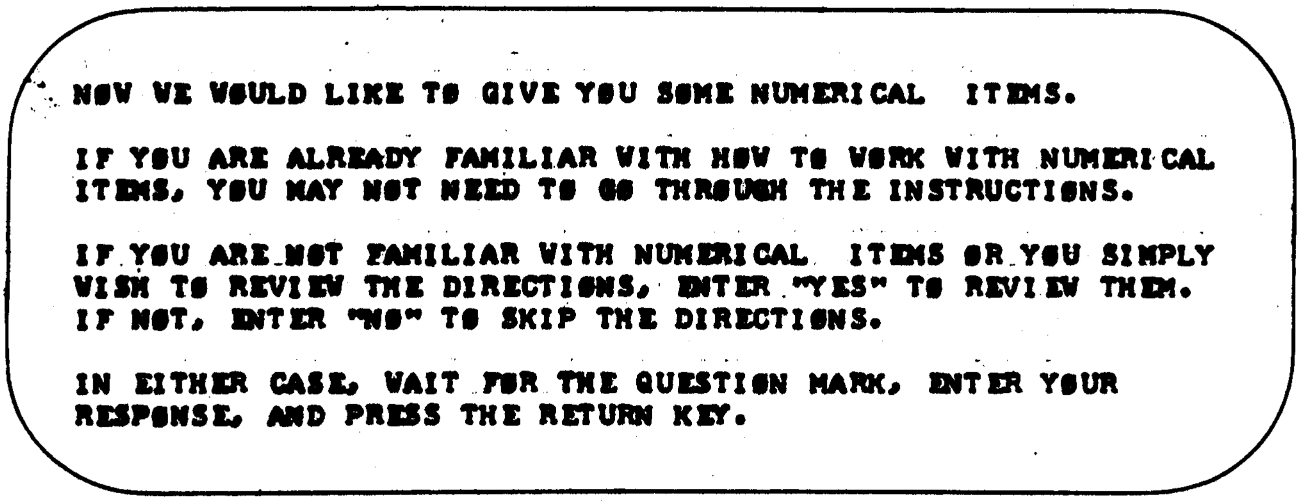
NOW WE WOULD LIKE TO GIVE YOU SOME VOCABULARY ITEMS.

IF YOU ARE ALREADY FAMILIAR WITH HOW TO WORK WITH VOCABULARY ITEMS, YOU MAY NOT NEED TO GO THROUGH THE INSTRUCTIONS.

IF YOU ARE NOT FAMILIAR WITH VOCABULARY ITEMS OR YOU SIMPLY WISH TO REVIEW THE DIRECTIONS, ENTER "YES" TO REVIEW THEM. IF NOT, ENTER "NO" TO SKIP THE DIRECTIONS.

IN EITHER CASE, WAIT FOR THE QUESTION MARK, ENTER YOUR RESPONSE, AND PRESS THE RETURN KEY.

Screen 12V. Screen presented between tests where test 1 is number series and test 2 is vocabulary. If testee responds NO, testing resumes. If testee responds YES, screen 9 is presented before testing resumes.



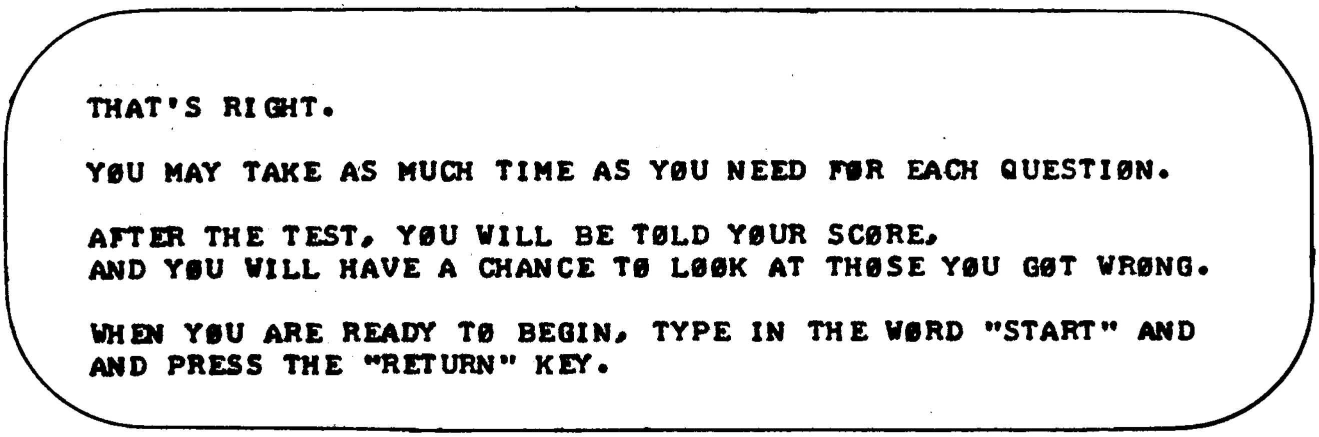
NOW WE WOULD LIKE TO GIVE YOU SOME NUMERICAL ITEMS.

IF YOU ARE ALREADY FAMILIAR WITH HOW TO WORK WITH NUMERICAL ITEMS, YOU MAY NOT NEED TO GO THROUGH THE INSTRUCTIONS.

IF YOU ARE NOT FAMILIAR WITH NUMERICAL ITEMS OR YOU SIMPLY WISH TO REVIEW THE DIRECTIONS, ENTER "YES" TO REVIEW THEM. IF NOT, ENTER "NO" TO SKIP THE DIRECTIONS.

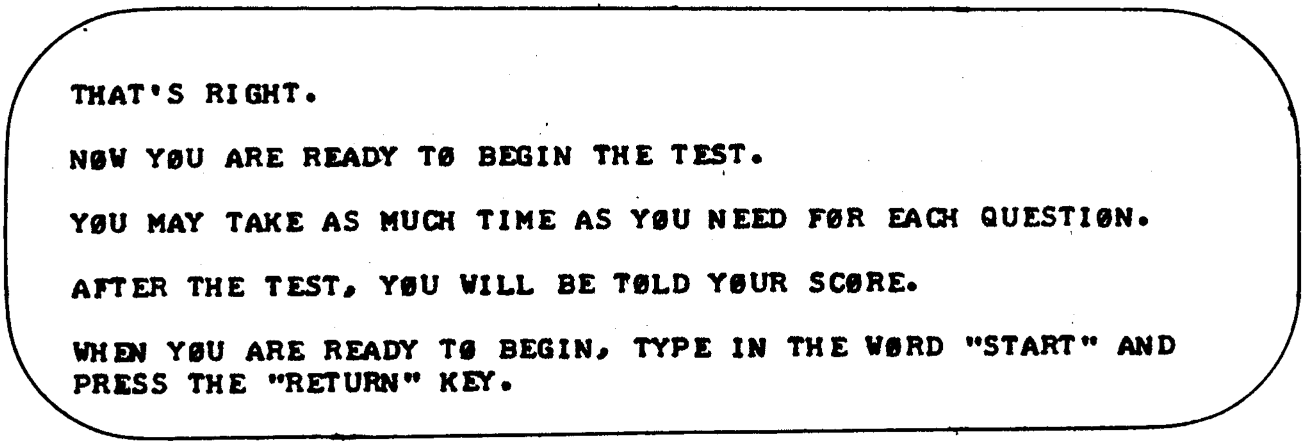
IN EITHER CASE, WAIT FOR THE QUESTION MARK, ENTER YOUR RESPONSE, AND PRESS THE RETURN KEY.

Screen 12N. Screen presented between tests where test 1 is vocabulary and test 2 is number series. If testee responds NO, testing continues. If testee responds YES, screens 10N and 11N are presented before testing resumes.



```
THAT'S RIGHT.

YØU MAY TAKE AS MUCH TIME AS YØU NEED FØR EACH QUESTIØN.

AFTER THE TEST, YØU WILL BE TØLD YØUR SCØRE,
AND YØU WILL HAVE A CHANCE TØ LØØK AT THØSE YØU GØT WRØNG.

WHEN YØU ARE READY TØ BEGIN, TYPE IN THE WØRD "START" AND
AND PRESS THE "RETURN" KEY.
```

Screen 10 F. Tenth instructional screen if post-feedback is to be presented.



```
THAT'S RIGHT.

NØW YØU ARE READY TØ BEGIN THE TEST.

YØU MAY TAKE AS MUCH TIME AS YØU NEED FØR EACH QUESTIØN.

AFTER THE TEST, YØU WILL BE TØLD YØUR SCØRE.

WHEN YØU ARE READY TØ BEGIN, TYPE IN THE WØRD "START" AND
PRESS THE "RETURN" KEY.
```

Screen 10 NF. Tenth instructional screen if no feedback is to be given.

APPENDIX E

Procedure File for Executing Testing Program

Large computers are controlled by a software supervisory system (also referred to as a monitor, executive system or operating system). This "system" is essentially a program which controls the flow of programs through the computer as efficiently as possible. The functions of this operating system include the allocation of memory space, the allocation and manipulation of files, the loading and execution of programs and the monitoring of time limits.

There are two methods for giving commands to the Kronos operating system on Control Data Corporation's 6400 computer: 1) In an interactive mode where each command is entered on a terminal (CRT or teletype) and executed before the next command can be sent or 2) In a "batch" mode where a file (called a procedure file) containing an entire sequence of commands is prepared by the user and called at some later time: the user can issue no new commands until all of the commands on this procedure file have been executed.

A combination of both of these types of commands is used to execute the testing program. After connecting the phone and logging on, the proctor must enter:

OLD,CAT — This attaches a procedure file.

The system responds with a / (slash). The proctor then enters:

CALL,CAT — This command gives control to the procedure file CAT.

The procedure file CAT consists of the 7 commands listed and explained below.

RFL,32500. — This requests a field length or central memory storage space of 32500 words.

GET,SUPTEST/UN=xxxxxxx

This attaches a file called SUPTEST from account number xxxxxxx. The testing system currently operates on several different account numbers and program files are kept under one account number since duplicate copies are not needed. This procedure also protects the testing programs since they cannot be destroyed or altered by testees logged onto the system on different account numbers.

Procedure file for executing testing program (cont'd)

OFFSW,2. Switch 2 is an electronic switch which can be set on or off by commands and read by a program. Switch 2 is read by the testing program. If it is off, instructions developed for University students are presented. If it is on, instructions for high-school students are presented. This command turns switch 2 off.

SUPTEST. The file SUPTEST contains the main overlay for the testing system. This command causes the file to be loaded and executed, hence beginning the program.

RETURN,SUPTEST,INPUT,OUTPUT,INSTR. This command is not executed until testing is finished. It serves the function of returning several of the files to the system; in essence, giving up the disc space they occupied.

RETURN,TAPE7,TESTS,TAPE8. This command is similar to that above.

EXIT. This returns control to the CRT or proctor who can determine if the testing program should be executed again.

A second procedure file, CHS, is usable by entering OLD,CHS and CALL,CHS.

This file is identical to the procedure file CAT except for the third command, which is ONSW,2. instead of OFFSW,2. This serves to turn on switch 2 which, when read by the testing program, causes instructions for high-school students to be presented.

APPENDIX F

Response Codes Returned to Testing Strategy Programs by Subroutine ITEM

-3 - testing strategy program has requested a valid but unavailable item number

-2 - testing strategy program has requested an invalid item number

-1 - testing is terminated by proctor

0 - item answered incorrectly

1 - item answered correctly

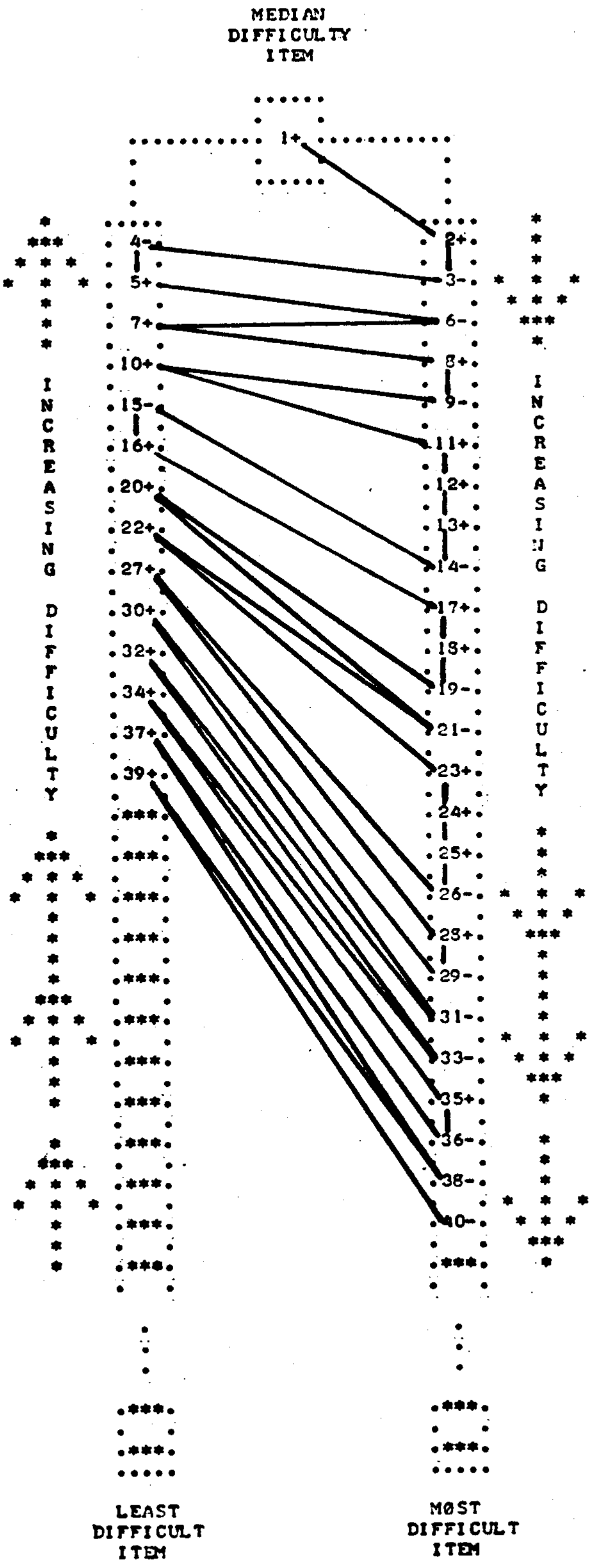
2 - item answered with a ?

APPENDIX G

Displays of Test Results from all Testing Strategies except Linear Testing

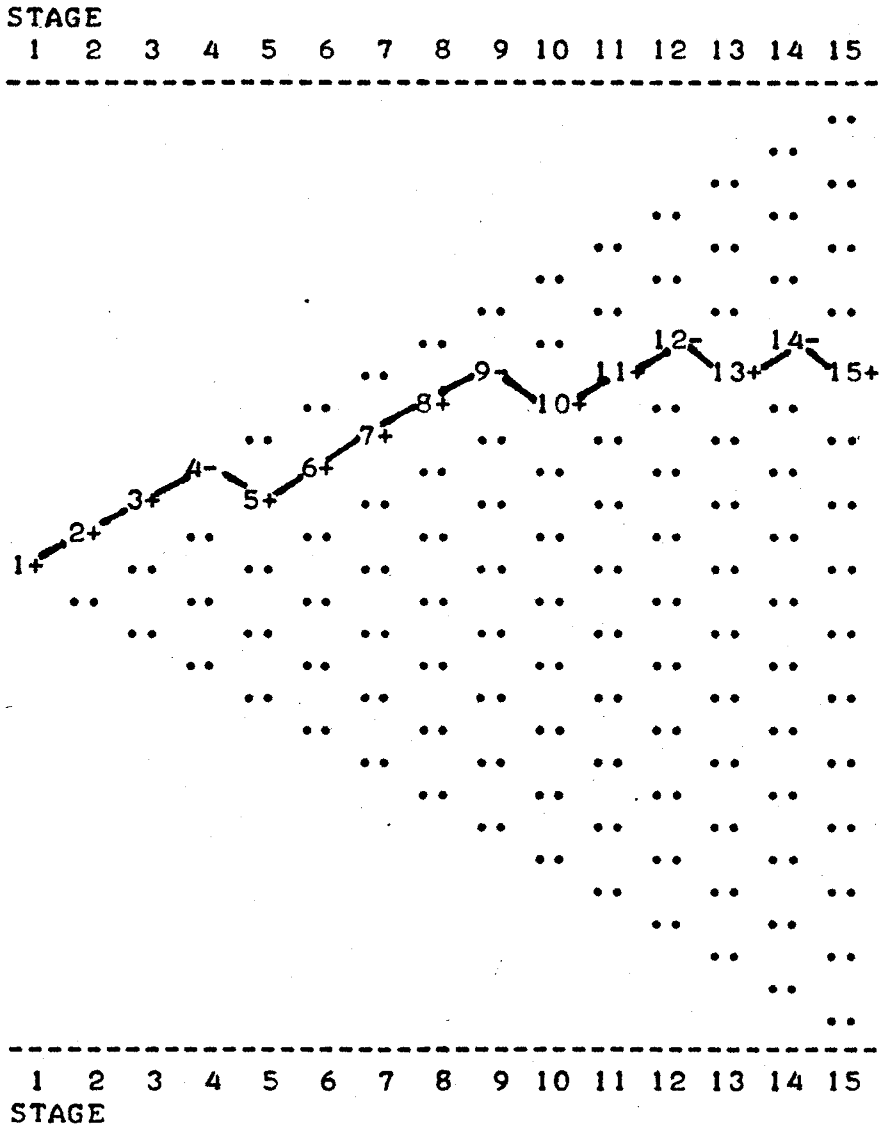
The display programs for Two-stage, Flexilevel, Stradaptive and Bayesian strategies were written by David Vale; the display program for Pyramidal testing was written by Robert Swisher.

```
                    REPØRT ØN BAYESIAN TEST

ID NUMBER:      2067                         DATE TESTED:  73/05/03

            X=CØRRECT  O=INCØRRECT  ?=NØ RESPØNSE
       ERRØR BAND PLØTTED IS + AND - STANDARD DEVIATIØN

                         ABILITY LEVEL                          ABILITY
   -2.5 -2.0 -1.5 -1.0 -0.5  0.0  0.5  1.0  1.5  2.0  2.5     EST     SD
I----I----I----I----I----I----I----I----I----I----I----I----I

                 ..............E..............                  .17  1.41
           .........O.........                                 -.93   .98
                  .......X.......                              -.42   .78
                      ......X......                            -.16   .68
                   ......O......                               -.42   .61
                       .....X.....                             -.12   .55
                      .....O.....                              -.29   .51
                     ....O....                                 -.45   .48
                    ....O....                                  -.60   .46
                     ....X....                                 -.50   .44
                   ....O....                                   -.61   .42
                  ....O....                                    -.72   .41
                  ...O...                                      -.88   .39
                ...O...                                       -1.01   .37
               ...O...                                        -1.16   .35
             ...O...                                          -1.33   .32
             ...?...                                          -1.33   .32
              ...X...                                         -1.25   .32
                ..X..                                         -1.19   .30
                ..X..                                         -1.13   .30
                 ..X..                                        -1.07   .28

                    20 ITEMS WERE ADMINISTERED
                    REPØRT ØN FLEXILEVEL TEST

ID NUMBER:    690134                        DATE TESTED:  73/07/12
---------------------------------------------------------------------

   + =CØRRECT ANSWER    - =INCØRRECT ANSWER    ? =NØ ANSWER
             XXX =ITEM NØT ADMINISTERED
```



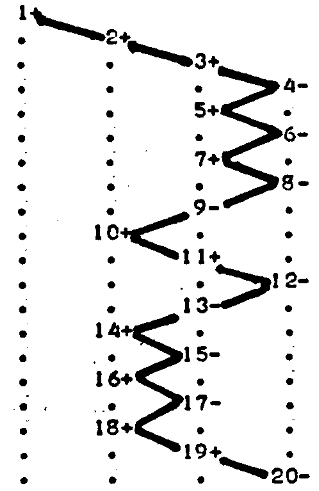
```
TØTAL NUMBER ØF ITEMS IN TEST= 79
NUMBER ØF ITEMS ADMINISTERED= 40
NUMBER ØF ITEMS CØRRECT= 25
SCØRE= 25.50
```

```
                    REPØRT ØN PYRAMIDAL TEST

 ID NUMBER:    554911                       DATE TESTED:  73/04/09
```



```
THERE WERE 11 CØRRECT ANSWERS.

THESE HAD A MEAN DIFFICULTY ØF    .63

THERE WERE 15 ANSWERS IN TØTAL.

THESE HAD A MEAN DIFFICULTY ØF    .75

THE FINAL ITEM HAD A DIFFICULTY ØF  1.01

THE DIFFICULTY ØF THE (N+1)TH ITEM IS  1.33
```

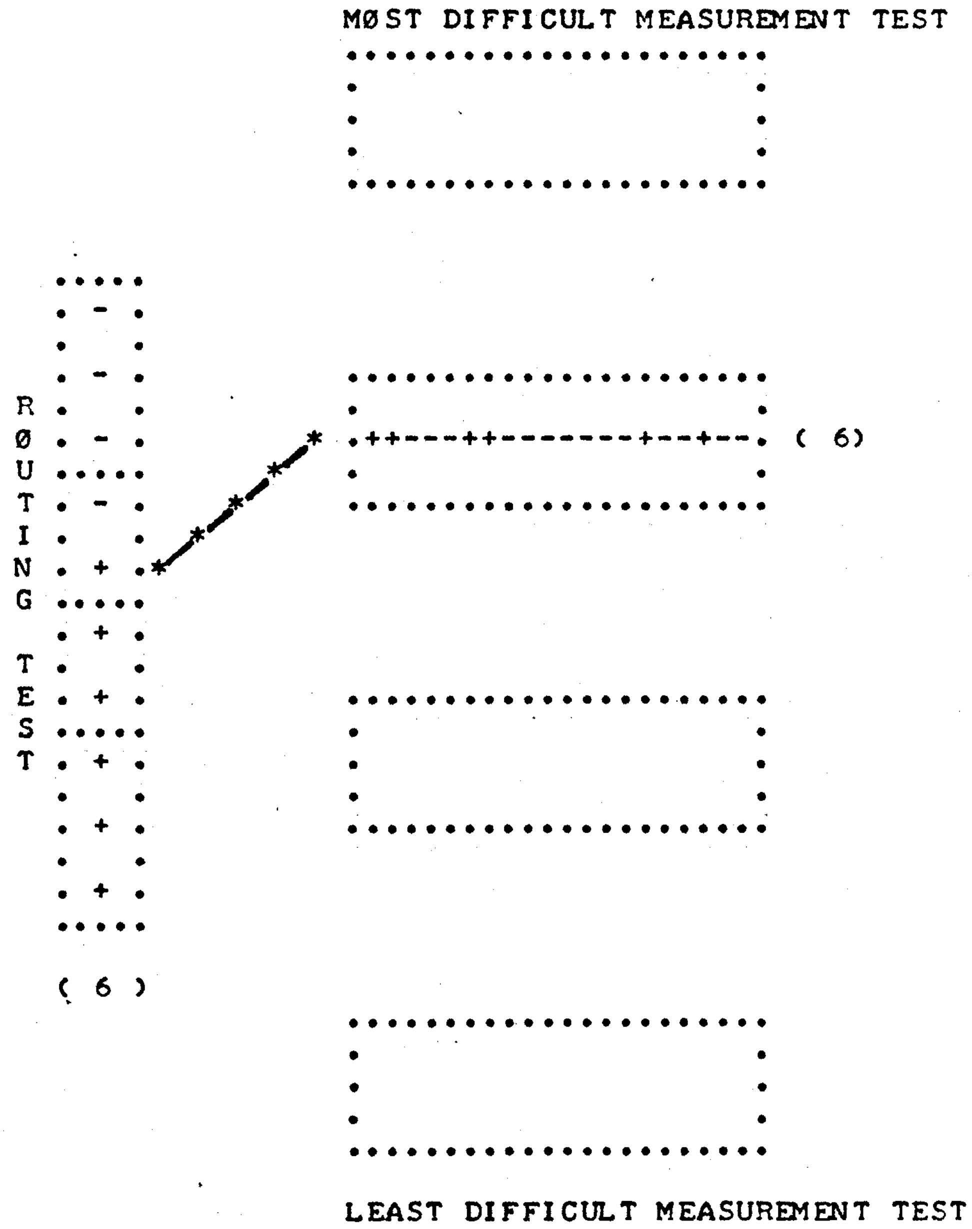
```
                              REPØRT ØN STRADAPTIVE TEST

ID NUMBER: 469625594                                      DATE TESTED:  73/07/12

------------------------------------------------------------------------------------

                     (EASY)                                              (DIFFICULT)
STRATUM:               1       2       3       4       5       6       7       8       9
```



```
PRØP.CØRR:                                            1.00    1.00     .56    0.00

                          TØTAL PRØPØRTIØN CØRRECT=  .550


              SCØRES ØN STRADAPTIVE TEST

              1. DIFFICULTY ØF MØST DIFFICULT ITEM CØRRECT=  1.49

              2. DIFFICULTY ØF THE N+1 TH ITEM=  1.44

              3. DIFFICULTY ØF HIGHEST NØN-CHANCE ITEM CØRRECT=  1.49

              4. DIFFICULTY ØF HIGHEST STRATUM
                 WITH A CØRRECT ANSWER=  1.33

              5. DIFFICULTY ØF THE N+1 TH STRATUM=  1.33

              6. DIFFICULTY ØF HIGHEST NØN-CHANCE STRATUM=  1.33

              7. INTERPØLATED STRATUM DIFFICULTY=  1.37

              8. MEAN DIFFICULTY ØF ALL CØRRECT ITEMS=   .88

              9. MEAN DIFFICULTY ØF CØRRECT ITEMS
                 BETWEEN CEILING AND BASAL STRATA=  1.28

             10. MEAN DIFFICULTY ØF ITEMS CØRRECT
                 AT HIGHEST NØN-CHANCE STRATUM=  1.28
             11. SD ØF ITEM DIFFICULTIES ENCØUNTERED=   .59

             12. SD ØF DIFFICULTIES ØF
                 ITEMS ANSWERED CØRRECTLY=   .46

             13. SD ØF DIFFICULTIES ØF ITEMS ANSWERED
                 CØRRECTLY BETWEEN CEILING AND BASAL STRATA=   .18

             14. DIFFERENCE IN DIFFICULTIES
                 BETWEEN CEILING AND BASAL STRATA=  1.36

             15. NUMBER ØF STRATA BETWEEN
                 CEILING AND BASAL STRATA=  1
```

REPØRT ØN TWØ-STAGE TEST

```
ID NUMBER:        2100                          DATE TESTED:   73/05/09
------------------------------------------------------------------------
```


MØST DIFFICULT MEASUREMENT TEST
R
Ø
U
T
I
N
G

T
E
S
T
(6)
(6)
LEAST DIFFICULT MEASUREMENT TEST

ABILITY ESTIMATE IN STANDARD SCØRE IS:-2.145

ABILITY ESTIMATE IN PERCENTILE IS: 1.599

APPENDIX H

Technical Requirements for a Computer-Assisted Testing System

The current testing system was developed and is currently operational on MERITSS (Minnesota Educational Regional Interactive Time-Sharing System). MERITSS is based on a Control Data 6400 computer system which can handle more than 70 ports simultaneously. Approximately 75 million characters of disc storage are available as well as approximately 65,000 words (10 characters or 60 bits per word) of central memory, making this a large-scale computer system.

Disc or Central Memory Requirements

The present testing system requires approximately 13,000 words (130,000 characters) of central memory storage for execution. Slightly more is required for compilation of the programs involved--approximately 18,000 words (180,000 characters).

Storage, on disc, of the different overlays used takes up approximately 264,000 characters. The random access file for vocabulary items uses 119,000 characters of storage; that for number series items uses 64,000 characters. In addition to these minimal requirements, each subject's data file takes up from 640 to 6000 characters of storage. Transferring the testees' data files to magnetic tape at frequent intervals will cut down total disc storage requirements but often over 200 testee data files are on the system representing as much as a million characters of disc space. The addition of other small auxiliary files such as CATCODE (for valid keywords) and ABILS (for ability estimates on retest) brings total disc capacity needed to approximately two million characters.

Software Requirements

The entire testing system was written in Fortran and compiled using the Control Data 6000 series Fortran IV compiler. This is a very powerful compiler containing almost all Fortran IV conventions. There are no system subroutines used in the testing system which are not supplied by Control Data (i.e., there are no local installation routines used).

Control Data 6000 series computers are based on a 60-bit (10-character) fixed word length. This fixed word length was an important consideration in writing several of the routines and will seriously affect any transfer of the testing system to other computer installations.

Transportability

The present testing system connects with user ports over standard voice grade telephone lines--one line per port. Thus, transferring the terminals to any location where phone lines are available is possible.

Transferability

Two aspects must be considered with respect to the transferability of the testing system: 1) use of CRT's from different manufacturers as terminals and 2) transfer of the program to another computer.

CRT's. The current testing system was written for Computer Terminal Corporation (CTC) Datapoint 3000 CRT terminals. The testing program uses special codes, which are not standardized, to cause the CRT to move the cursor to the top of the screen, erase the screen and backspace the cursor. Use of other manufacturers' CRT's might require changing these special codes within the testing program.

The Datapoint CRT's have a screen which contains 24 72-character lines. Other CRT's might have substantially smaller screens--usually fewer lines. Many of the instructional screens use the full 24 lines and would have to be rewritten for a smaller CRT.

Programs. Transferring the testing system to another computer presents four levels of increasing difficulty of modification. All but the last assume that a Fortran IV compiler is available and all assume sufficient disc storage space.

1. The easiest transfer would be to a similar Control Data 6000 series computer. Some very minor modifications might be necessary to conform to local installation parameters but no major changes should be necessary.

2. A more difficult transfer assumes that the new system has sufficient central memory capacity but is not a Control Data 6000 series machine. The Control Data Fortran IV Series compiler permits the use of star fields (*) in format statements for displaying strings of alphanumeric characters. This feature is not available with most other compilers so almost every displayed screen in the program would require rewriting in Hollerith fields. This would affect the instructional and display overlays significantly.

The second, and more severe, problem hinges on the fact that the CDC 6000 series computers use a 10-character (60-bit) word of fixed length. Some sections of the program are nearly independent of word length considerations. Subroutine ITEM, however, is very dependent on this fact. Program sections dealing with item retrieval from disc, item presentation and storage of data in the subject's data file would have to be rewritten. The instructional overlay saves some collected data which is dependent on word length. The instructional screens themselves are nearly independent of word length. The feedback overlay's presentation section would require rewriting. The display overlay is somewhat dependent on word length but would probably not require major revision. In addition to these two major problems there are probably a number of Fortran statements unique to the CDC 6000 series Fortran IV Compiler. These would have to be found and changed to a form acceptable to the compiler at the new installation.

3. The still more difficult transfer would be to a system lacking sufficient central memory storage to store the program in its present operating configuration. Assuming all necessary modifications described in (2) above have been completed, the difficulty of rewriting for less storage varies inversely with the amount of central memory space available. Rewriting in this situation would require breaking the program into smaller and smaller overlays, segments, or chains until the largest configuration in use at one time would fit within the given central memory constraints. This is entirely feasible, although it must be noted that as the size of each overlay, segment, or chain decreases the number required increases and a significant amount of time would be used locating each one on disc and loading it into memory as needed.

4. The most difficult transfer would be to a system where no Fortran IV compiler was available. In this situation only the logic or structure of the system could be used and would serve as a structure around which a new system could be rewritten.