# DOCUMENT RESUME

ED 238 395 IR 010 916

| | |
|---|---|
| AUTHOR | Bayman, Piraye; Mayer, Richard E. |
| TITLE | Novice Users' Misconceptions of BASIC Programming Statements. Report No. 82-1. |
| INSTITUTION | California Univ., Santa Barbara. |
| SPONS AGENCY | National Inst. of Education (ED), Washington, DC. |
| PUB DATE | [82] |
| GRANT | NIE-G-80-0118 |
| NOTE | 46p. |
| PUB TYPE | Reports - Research/Technical (143) |
| EDRS PRICE | MF01/PC02 Plus Postage. |
| DESCRIPTORS | Aptitude Tests; Cognitive Processes; College Students; *Concept Formation; Diagnostic Tests; Educational Research; Epistemology; Higher Education; Microcomputers; *Models; *Programing; *Programing Languages |
| IDENTIFIERS | *BASIC Programing Language; *Misconceptions |

ABSTRACT

The nature of novice programmers' mental models for BASIC statements following preliminary BASIC instruction was assessed with 30 undergraduates who were taught BASIC through a self-paced, mastery manual and who were simultaneously given hands-on access to an Apple II microcomputer. Following instruction, the students were tested to determine their conceptions of what goes on inside the computer during the execution of each of nine BASIC statements: (1) LET A = B + 1; (2) PRINT C; (3) LET D = O; (4) PRINT "C"; (5) IF A IS LESS THAN B GOTO 99; (6) INPUT A; (7) 20 DATA 80, 90, 99; (8) 30 READ A; and (9) 60 GOTO 30. Data gathering instruments included a questionnaire to obtain information on the subject's demographic characteristics and background in mathematics and/or computer programming; the IBM Programmer Aptitude Test; and both verbal and visual (diagram) tests on the nine statements. Results showed that despite adequate performance on program generation mastery tests, students possessed a wide range of misconceptions concerning the statements they had learned. This paper presents a catalog of the subjects' mental models for each of the statements, frequency tables showing incorrect or incomplete transactions produced for each statements, and a six-item reference list. Appendices include instructions, a sample page, and a sample protocol from each of the two tests on the BASIC statements. (Author/LMM)

# Novice Users' Misconceptions of BASIC Programming Statements

Piraye Bayman and Richard E. Mayer

Report No. 82-1

Preparation of this report was supported by Grant NIE-G80-0118 from the National Institute of Education.

Abstract

In the process of learning a computer language, beginning programmers may develop mental models for the language. A mental model refers to the user's conception of the "invisible" information processing that occurs inside the computer between input and output. In this study, thirty undergraduates learned BASIC through a self-paced, mastery manual and simultaneously had hands-on access to an Apple II computer. After instruction, the students were tested on their mental models for the execution of each of nine BASIC statements. The results show that beginning programmers--although able to perform adequately on mastery tests in program generation--possessed a wide range of misconceptions concerning the statements they had learned. For example, the majority of the beginning programmers had either incorrect conceptions for or no conceptions of statements such as INPUT A, READ A, and PRINT C. This paper presents a catalogue of beginning programmers' conceptions of "what goes on inside the computer" for each of nine BASIC statements.

# Novice Users' Misconceptions of BASIC Programming Statements

## Learning BASIC

This paper provides new information concerning how beginning programmers learn BASIC. Suppose that you were going to teach a beginning programmer how to use a programming language such as BASIC. The typical instructional sequence involves an orderly presentation of the elementary statements. For each statement, the following information is generally presented: definition, grammar, format specifications, example programs, and printouts. The apparent goal of most instructional sequences is to teach the user to perform. The typical instructional sequence occasionally includes partial descriptions of internal processes or states in the computer--such as memory locations, input stacks, etc.--but such descriptions are relatively rare and unsystematic. Thus, relatively little attention s[illegible] to be paid to the instructional goal of teaching the user to understand.

## What is Learned

The main focus of this paper concerns "what is learned" when a beginning programmer is taught a computer programming language such as BASIC, following a typical instruction sequence.

The outcome of learning--what is learned--can be viewed in two distinct ways:

Learning BASIC involves acquisition of new information. This idea asserts that learning a computer programming language is like learning any subject matter; the learner must acquire specific facts, skills, and rules. For example, the learner acquires new information such as format rules for when to use quotes in a PRINT statement or how to produce a conditional loop using an IF statement. Acquisition of specific information is generally the explicit objective of instruction.

Learning BASIC involves acquisition of a mental model. This idea asserts that, during the course of learning, a user develops a conception of the invisible actions and states that occur in the computer between input and output. For example, the learner acquires new mental models such as the idea of memory spaces for holding numbers as in a counter set LET statement. Acquisition of mental models is generally not an explicit objective of instruction.

If we conceive of the learner as an active, thinking individual, then the acquisition of new information and the acquisition of a mental model can be looked upon as complementary processes that occur in the course of learning. Most instructional effort is directed towards the acquisition of new information. However, understanding of how mental models are acquired in the process of learning the new information might be useful for the instructor as well as the designer. One hypothesis is that basic training in the essential information coupled with the opportunity to practice at a computer terminal will naturally lead to the acquisition of "useful" mental models--models that can enhance understanding of the computer language. The present study aims to explore this hypothesis.

## Mental Models for BASIC

The present study explores the idea that learning of BASIC involves more than the acquisition of specific facts, rules, and skills. Beginning programmers also develop mental models for the language in the process of learning the essentials of BASIC. Users' models, however, may not be accurate or useful ones. That is, users develop individual conceptions of "what is going on inside the computer" as well as learn specific facts.

Mayer (1979) has suggested a framework for describing the internal transformations that occur for elementary BASIC statements. In particular, any BASIC statement can be conceptualized as a list of transactions. A transaction is a simple statement asserting some action performed on some object at some location in the computer. For example, the following set of transactions describes what happens inside the computer when the statement 10 LET A = 0 is executed:

(1) Find the number in memory space A (ACTION: Find; OBJECT: Number; LOCATION: Memory).

(2) Erase the number in memory space A (ACTION: Erase; OBJECT: Number; LOCATION: Memory).

(3) Find the number indicated on the right of the equal sign (ACTION: Find; OBJECT: Number; LOCATION: Statement).

(4) Write this number in memory space A (ACTION: Write; OBJECT: Number; LOCATION: Memory).

(5) Find the next statement in the program (ACTION: Find; OBJECT: Statement; LOCATION: Program).

Experts and novices are likely to differ with respect to their mental models for a programming language. For example, an expert programmer may have developed an accurate conception for a counter set LET, such as the one given above. That is, the expert knows that the value in memory space A is replaced by 0. However, the novice may lack a coherent mental model or may possess incorrect ones for BASIC statements.

In a recent study, Mayer & Bayman (1981) asked novice and expert calculator users to predict the answer that would be displayed for a series of problems such as 2+3+ or 2+++=. Subjects' responses were matched to transactional expressions for each button press. Results indicated that subjects differed greatly in their conceptions of "what goes on inside the calculator" for each key press, with experts possessing more sophisticated conceptions than novices. Thus, although all subjects were able to use calculators to solve basic math problems--i.e. all users had acquired the basic information for performance--they differed greatly in how sophisticated a mental model they possessed. This work with calculators suggests that learning of the basic information about how to use a language does not guarantee that a user has also acquired a useful mental model for the language.

Moran (1981) suggests that the user develops a "conceptual model" of the system as he or she learns and uses it. He defines the user's conceptual model as the knowledge that organizes how the system works and how it can be used to accomplish tasks. How much training one needs to acquire a conceptual model has not been explored yet. Studying novices' understanding of the programming statements constitutes the first step in addressing this issue. In this paper, we describe novice programmers' conceptions of elementary BASIC statements, using a transaction analysis. There is yet no empirical evidence on novices' mental models of the statements of BASIC, at this level of detail.

## Method

The goal of this study is to assess the nature of novice programmers' mental models for BASIC statements, following preliminary instruction in BASIC. In particular, this study assesses the transactions that each subject attributes to each of nine BASIC statements, following instruction in how to use these statements. Some transactions are essential for understanding a statement. We will determine the number of subjects who show evidence of having acquired these essential transactions for each statement. Some transactions are incorrect, such as thinking that READ A involves printing out a number. We will determine the number of subjects who show evidence of incorrect transactions for each statement. The final product of this study will be a frequency table for each statement, showing the number of subjects possessing each major transaction.

Subjects

The subjects were 30 undergraduates at the University of California, Santa Barbara who had no prior knowledge about computer programming. Participation in the study was partial fulfillment of the requirements for the introductory psychology course that they were taking.

Materials and Apparatus

Questionnaire. The questionnaire was an 8½x11 sheet of paper consisting of typed questions regarding: (1) the subject's mathematics background; such as geometry, algebra, and calculus courses the individual had taken in high school and in college, (2) the subject's demographic characteristics; such as academic major, age, sex, GPA and SAT-Verbal and SAT-Math scores, and (3) the subject's computer programming background, such as whether the individual knew how to program a calculator and whether the individual had ever typed at a computer terminal.

IBM Computer Programmer Aptitude Test. This pretest consisted of five 8½x11 inch sheets of paper based on a shortened version of the three-part IBM Computer Programmer Aptitude Sample Test (Luftig, 1980). Part I (PAT-1) consisted of 25 number series problems, with a 5-minute time limit. Part II (PAT-2) consisted of 18 picture analogies, with an 8-minute time limit. Part III (PAT-3) consisted of 12 arithmetic story problems, with an 8-minute time limit.

Revised Minnesota Paper Form Board Test (Series AA). This pretest consisted of [illegible] spatial visualization problems, with an 8-minute time limit.

Lessons. Both Lesson I and Lesson II were composed from a self-instruction, self-paced, mastery text called BASIC in Six Hours (Marcus, 1980) that is widely used in teaching BASIC in the Microcomputer Laboratory of the University of California, Santa Barbara. The text required that the user has hands-on access to an Apple-II Computer. The text did include mastery tests at the end of each lesson but did not provide any conceptual models to explain the statements.

Lesson I. This eleven page typewritten lesson covered statement execution in the immediate mode. The statements taught were the PRINT, LET, and IF-THEN statements. There were exercises for doing simple arithmetic calculations, for having character strings printed on the display screen, and for assigning values to numerical and character string variables. The use of the semicolons and commas for spacing on the screen, the use of the colon for statement stacking and the use of the command NEW were taught in conjunction with the statements. The lesson ended with a 10-minute self test that covered the essential information taught in the booklet.

Lesson II. This fourteen page, typed lesson covered program preparation, adding or deleting statements in a program, obtaining the list of statements on the display screen, and how to run a program. Also the lesson covered the use of the GOTO statements, the IF statement, the numerical and character string INPUT statements, and the READ and DATA statements. Several sample programs were included, and there was a 10-minute self test at the end of the lesson.

Posttests. The study involved two posttests--verbal test and visual test--with the same nine statements tested in each.[2]

Verbal Test. This test contained [illegible] inch sheets of paper, with a single statement tested on each sheet. The statements in the test were (in order of presentation):

```
LET A = B + 1
PRINT C
LET D = 0
PRINT "C"
IF A < B GOTO 99
INPUT A
20 DATA 80, 90, 99
30 READ A
60 GOTO 30
```

The immediate mode statements were presented one on a sheet, with a carriage return (indicated by <CR>) after each; the line numbered statements were presented within a simple program. For both, subjects were instructed to write, in plain English, the steps that the computer would carry out for each statement. They were instructed to write each step on a separate line of the test sheet. Subjects were instructed to work on the statements in the given order and were not allowed to skip ahead or go back to the previous page.

Visual Test. This test involved the same nine statements as in the verbal test. presented in the same order and format. In addition, this test began with a three page typed introduction, which presented a visual diagram of the computer. The diagram was based on earlier experiments (Mayer, 1981) and consisted of four parts--the display screen with keyboard, the memory, the input system, and the control system. For each of the nine statements, subjects were asked to indicate the state of each of the four components of the computer after a particular statement was executed. For each statement there was a diagram that showed the state of the computer before the statement was executed and another diagram that was left blank for the subject to fill in.

Apparatus. Apparatus consisted of four Apple II Plus computers, with 48K of memory, and with Amdek 12" Model 100 B/W monitors.

Procedure

The study consisted of six sessions. In the first session, subjects in groups of up to six filled out the questionnaire, received a brief description of the study, signed up for times for the next five sessions, and took the series of the four pretests. Tests were administered in the order, PAT-1, PAT-2, PAT-3 and Minnesota Paper Form Board Test, with time limits of 5, 8, 8, and 8 minutes, respectively. In the following three sessions each subject worked on the two lessons at his or her own pace. Subjects worked individually at the Microcomputer Laboratory,[1] using Apple II computers in conjunction with the text. Following each lesson, subjects' performance on the self test was checked, they were asked to work on the questions they missed until they passed the test. In the final two sessions subjects took the posttests at their own pace, in groups of no more than five people at one time. The verbal test was always given before the visual test.

## Results

Scoring

Each subject's protocol for each of the nine statements on the verbal test was scored by two judges. Each protocol was broken down into a list of transactions. A transaction expresses some action performed on some object at some location in the computer (see Mayer, 1979). For example, one subject's protocol for LET A = B + 1 was:

1. Prints what is typed on screen

2. Understands what the word LET means
3. Solves the equation with information given
4. Gives a solution to problem

These steps can be translated into the following transactions:

PRINT LET A=B+1 on the screen.

FIND the statement that was just entered.

SOLVE the equation that was just entered.

PRINT the solution on the screen.

Disagreements between judges were rare and were settled by consensus.

Each subject's diagram for each of the nine statements in the visual test was scored by one judge.[3] The judge noted the specific contents of each of the four memory spaces, the specific contents of the display screen, the contents of the input system, and the direction of the arrow in the control system. (Note that executive control of order of line execution could not be determined in this test--so key aspects of IF and GOTO statements could not be scored.) Each subject's diagram was converted into a list of transactions by comparing the initial state of each component and the state of each component following statement execution as produced by the subject. For example, one subject fills in the components of the computer diagram for LET A = B + 1 as shown in Appendix F. As can be seen, the four memory spaces contain A = B + 1, 0, 21, 15, respectively (as compared to the initial contents of 0, 21, 15, 66, respectively), the display screen is intact showing READY followed by the cursor and the command LET A=B+1, the input system contains 5, 99, 6, and 7 in each case and the control arrow points to WAIT (as compared to RUN in the initial diagram). The transactions indicated by this subject are:

For the memory spaces --- <u>Write</u> A = B + 1 in memory space A.

Move the numbers in each space to the next memory space.

For the control system --- Wait for the next statement to be entered from the keyboard.

For each of the nine statements a list of correct transactions was generated, based on an earlier analysis of Mayer (1979). The correct transactions represent the events which are essential for describing the execution of the statement. For example, the correct transactions for LET A = B + 1 are as follows:

1. Find the number in memory space A.
2. Erase that number.
3. Find the number in memory space B.
4. Add one to that number.
5. Write the obtained value in memory space A.
6. Find the next statement that is entered at the keyboard.

The key transaction(s) for each statement were determined by two judges, based on the most characteristic event(s) in the statement. For example, the key transaction for LET A = B + 1 is, "Write the obtained value in memory space A."

For each statement, a list of alternative transactions was also generated, based on the subjects' answers. Some of the transactions were unnecessary, such as printing the statement on the screen. Some of the transactions were incomplete, such as writing B + 1 in memory space A. Incomplete versions of key transactions, such as the previous example, were also identified. Some of the transactions were incorrect, such as writing A = B + 1 in memory or printing the value of A on the screen. The incorrect transactions represent students' misconceptions of the events involved for a statement.

Frequency of Misconceptions

For each of the nine statements, a frequency table was generated by tallying each subject's answer on the verbal test and on the visual test against the list of all possible correct, unnecessary, incomplete and incorrect transactions. Tables 1 through 9 presents summaries of the data for each of the nine statements, respectively. Hence, each table lists only the major transactions for a given statement,[4] and shows the proportion of subjects who produced each transaction on the verbal test and on the visual test. Proportions are reported separately for the verbal and visual tests. The missing proportions for a transaction on either test indicates that the transaction could not be detected from that test. For example, transfer of control from one line to another cannot be detected from the visual test. The transactions in the tables are stated in plain English. Key transactions are indicated by double asterisks(**).

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

Tables 1 through 9 about here

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

Note that on the verbal test subjects are asked to write the steps that the computer goes through in executing the programming statement. In contrast, on the visual test subjects are asked to fill in the four-component computer diagram indicating the changes that occur due to the execution of the statement. The following expository discussion of results is based on the subjects' answers in both of the tests but focuses on the data from the verbal test. This is done because the verbal test allows a more detailed description of subjects' answers than the visual test, the verbal test was administered before the visual test, and the visual test did not provide information on transfer of control.

LET A=B+1. The major misconceptions for this statement can be grouped into three categories: (1) That the computer writes A=B+1 in memory or in memory space A. Forty-seven percent of the subjects thought that the computer stored the equation instead of the value obtained from B+1. (2) That the computer prints the equation or A or the value of A on the screen. Twenty-three percent of the subjects answered this way. (3) That the computer solves the equation A=B+1. Thirteen percent of the subjects believed that this is the case.

Only 30% of the subjects' answers included the key transaction that the computer would store the value from B+1 for the variable A.

Table 1 gives the specific proportions produced by subjects' answers of the major incorrect and incomplete transactions as well as of the set of correct transactions.

(2) LET D=O. The major misconceptions for this statement can be grouped into three categories: (1) That the computer writes the equation in memory. Forty-seven percent of the subjects opted for this idea. (2) That the computer solves the equation. Seven percent of the subjects answered this way. (3) That the computer prints the equation on the screen. Seven percent of the subjects thought this might happen.

The key transaction that the computer writes the value zero in its memory for the variable D is given by 47 percent of the subjects.

For more detailed description of the subjects' answers refer to Table 2 which includes proportions produced for an incomplete transaction and for the set of correct transactions besides proportions for the major incorrect transactions described above.

(3) PRINT C. The major misconceptions for this statement can also be grouped into three categories: (1) That the computer prints the letter C on

the screen. Thirty-three percent of the subjects incorrectly answered this way. (2) That the computer prints either error or nothing on the screen. Seven percent of the subjects had this particular conception. (3) That the computer writes C in its memory. Seven percent of the subjects held this idea.

The key transaction that the computer would print the value of the variable C on the screen was given by 40 percent of the subject population.

See Table 3 for more detailed classification of the subjects' answers for this statement.

(4) PRINT "C". Three major misconceptions for this statement are: (1) That the computer prints the value of the variable C on the screen. Seven percent of the subjects answered this way. (2) That the computer writes C in its memory. Again 7 percent of the subjects thought this was the case. (3) That the computer finds the number in memory space C. Only one subject answered this way.

Eighty-three percent of the subject population stated the key transaction that the computer would print C on the screen.

See Table 4 for more detailed information on the subjects' answers for this statement.

(5) INPUT A. The major misconceptions for this statement are: (1) That the computer writes A in a data list or memory. Thirty percent of the subjects' answers fell in this category. (2) That the data or A will be printed on the screen. Only one subject held this conception.

There are three key transactions for the INPUT statement: (1) That the computer prints a question mark on the screen. Seven percent of the subject population stated this transaction. (2) That the computer waits for the value of A and <CR> to be entered from the keyboard. Twenty-three percent of the subjects' answers implied this conception. (3) That the computer stores the entered value in memory space A. Only one subject held this conception.

See Table 5 for the proportions produced by subjects' answers for the major transactions for this statement.

(6) IF A<B GOTO 99. The major misconceptions can be grouped into four categories: (1) That the computer prints number 99 or line 99 or an error on the screen. Twenty percent of the subject population believed this was the case. (2) That the computer finds number 99 if A is less than B. Thirteen percent of the subjects answered this way. (3) That the computer writes A or B or A is less than B in memory. Ten percent of the subjects' answers fell in this category. (4) That the computer moves to line 99 without any test of whether the condition A<B is true. Ten percent of the subjects thought that this was the case.

There are two key transactions for the conditional GOTO statement: (1) That execution would move to line 99 in the program if the value of A is less than the value of B. Sixty-three of the subjects held this conception. (2) That execution would continue with the next statement in the program if the value of A is not less than the value of B. Only 33 percent of the subject population stated this conception in their protocols.

For more detailed list of the transactions and the corresponding proportions of subjects' answers see Table 6.

(7) 20 DATA 80, 90, 99. The major misconception for the DATA statement is that the computer prints the numbers on the screen. Thirteen percent of the subject population answered this way.

Only 27 percent of the subjects gave the precise answer that the numbers 80, 90, 99 would be put in memory or the input queue.

See Table 7 for more detailed categorization of the subjects' answers.

<u>(8) 30 READ A.</u> The major misconceptions for the READ statement can be grouped into three categories: (1) That the computer <u>prints</u> the value of A on the screen. Ten percent of the subjects believed in this idea. (2) That the computer <u>writes</u> A in memory. One subject stated this conception. (3) That the computer <u>waits for</u> a value to be entered from the keyboard. Again, only one subject held this wrong conception.

Only 10 percent of the subject population answered that the first date value from the DATA statement would be written in memory space A.

See Table 8 for more detailed description of the subjects' answers for the READ statement.

<u>(9) 60 GOTO 30.</u> The major misconceptions are: (1) That the computer <u>finds</u> 30 if A is not equal to some number. Seven percent of the subjects answered this way. (2) That the computer <u>prints</u> line 30 on the screen. Only one subject gave this answer.

There are two key transactions for the simple GOTO statement: (1) That the computer moves program execution to line 30 in the program. Sixty-seven percent of the subject population held this conception. (2) That the computer continues with program execution from that line. Only 37 percent of the subjects' answers implied that they had this idea.

Table 9 lists these transactions and proportions corresponding to each transaction.

<u>Differences among statements.</u> In order to make comparisons of subjects' conceptions among the nine statements, each subject's verbal protocol for each statement was categorized as correct, incomplete, empty, or incorrect. The criteria for classifying protocols were: (1) <u>correct</u>--if the subject's protocol included the key transaction(s) and no incorrect transactions, (2) <u>incomplete</u>--if the subject produced one or more incomplete versions of the

key transaction(s) and no incorrect transactions, (3) incorrect--if the subject produced one or more incorrect transactions, and (4) empty--if the subject produced no key transactions, no incomplete version of a correct transaction, and no incorrect transaction. Table 10 presents a summary of the proportion of users producing each type of conception for each of the nine BASIC statements.

Table 10 presents the nine statements in order of difficulty based on proportion correct conceptions. As can be seen, PRINT "C" is the best comprehended statement--with 80% of the subjects expressing the correct conception--while INPUT A is the worst understood statement--with only one subject indicating a correct conception. It should also be noted that substantial numbers of subjects hold "empty" conceptions for the READ, DATA, INPUT, and PRINT C statements.

---

Table 10 about here

---

A corresponding summary table based on the visual test was also prepared. Table 11 presents seven of the nine statements in order of difficulty based on proportion correct statements for this test. Data for the IF and GOTO statements has been excluded because the visual test did not allow for expressing transfer of control adequately. Although there seems to be general similarity in the patterns of performance between the two tests, making specific comparisons between them is not appropriate because of the differences in the nature of the tests. Note that the verbal test allowed subjects to exhibit incomplete and empty conceptions; in contrast, the visual test with its four-part computer diagram forced subjects to be specific.

---

Table 11 about here

---

Pretests. An additional analysis was performed in order to determine whether performance on the pretests was related to development of the correct conceptions of the BASIC statements. First, correlations were run between each pretest score (MPFBT, PAT-1, PAT-2, and PAT-3) and the total number of correct conceptions for the nine BASIC statements. Table 12 shows the correlation matrix, with significant r values marked with an asterisk(*). As can be seen, only PAT-2--a spatial reasoning test--seems to be significantly related to having correct conceptions for the BASIC statements ($r = .43$, $p < .02$). Also, the two spatial reasoning tests, MPFBT and Pat-2, show a significant correlation ($r = .49$, $p < .01$) as well as PAT-2 and PAT-3 ($r = .39$, $p < .05$). In addition, a stepwise regression analysis was performed with each of the four pretest scores as the independent variables and the total number of correct conceptions of statements as the dependent variable. The resulting equation selected only one variable--PAT-2--and accounted for 18 percent of the variance in the dependent measure. These analyses suggest that the PAT-2 test was the only test that was related to understanding of the BASIC statements, although the relationship is rather mild.

---

Table 12 about here

---

## Conclusions and Recommendations

In this study we attempted to evaluate the mental models acquired by novices following several hours of learning and practice in BASIC. It is important to note that we focused on novices' learning of mental models rather than learning of specific information. In order to evaluate mental models, we adopted a detailed level of analysis--transaction analysis (Mayer, 1979). This analysis has allowed us to identify the misconceptions and missing conceptions in novices' understanding of the conceptual models underlying elementary BASIC statements.

The results of the study indicate that mastery of the information in the instructional booklet does not guarantee that users will acquire useful mental models for how the BASIC statements function. Although all of the subjects in this study were able to adequately answer the questions on the mastery test at the end of each lesson, only about one-third of the subjects were able to produce correct conceptual descriptions of an average statement such as PRINT C or LET A=B+1 and more than one-half of the subjects generated incorrect or empty conceptual descriptions.

Not surprisingly, the nature of the misconceptions are found to differ from statement to statement; some statements being more difficult to grasp than others. The programming statements in order of difficulty (based on the verbal test) can be listed as: INPUT A, 30 READ A, IF A<B GOTO 99, LET A = B + 1, 20 DATA 80, 90, 99, 60 GOTO 30, PRINT C, LET D = 0, and PRINT "C". However, this part of the results should not be taken as conclusive since we not only used a selective number of programming statements but presented them with a specific order in the testing sessions. One needs to compose an exhaustive list of the programming statements of BASIC and use different presentation orders to be able to make conclusive remarks about the relative comprehensibility of each statement.

In this paper, we analyzed beginning programmers'as a group so that we could focus on the most common misconceptions. There seems to be sufficient evidence for specific recommendations to be made for each statement included in this study.

1. Recommendations concerning INPUT. Subjects have difficulty in conceiving where the to-be-input data comes from and how it is stored in memory. Many subjects fail to understand the nature of executive control--i.e. that the computer will "wait" for input from the keyboard. These learners need explicit training concerning the role of input terminal, the wait-run control, and the memory spaces--including visual representations, verbal descriptions of key transactions (as listed in Table 5), and role playing by the learner.

2. Recommendations concerning READ-DATA statements. Subjects have difficulty in conceiving where the to-be-read data comes from and how it is stored in memory. Subjects need explicit training concerning the data stack and the memory spaces--including visual representations, verbal descriptions of the key transactions (as listed in Tables 7 and 8), and role playing by the learner.

3. Recommendations concerning the conditional GOTO and simple GOTO. Subjects' major difficulty with the GOTO is that they cannot conceive of what will happen next after program execution moves on to the desired line. Also, with the conditional GOTO, they seem to have difficulty in conceiving what would happen next if the condition is false. Hence, beginners need training at the terminal to observe execution control, explicit training with verbal descriptions of the key transactions (as listed in Tables 6 and 9) and role playing to get a feel for execution control.

4. Recommendations concerning LET statements. Subjects seem to get confused between solving an equation (i.e. treating the equal sign as an equality) and making an assignment. Those who seem to understand the assignment property in the statement still have difficulties in conceiving where to store the assigned values. Beginning learners need explicit

training concerning the specific memory locations and under what conditions values stored in those locations get replaced. Visual representations, verbal descriptions consisting of the set of correct transactions (as listed in Tables 1 and 2), and role playing by the learner are ways to overcome beginners difficulties.

5. Recommendations concerning the PRINT statements. Subjects seem to confuse the function of PRINT C and PRINT "C". Also, subjects have difficulty in conceiving that these statements simply display on the screen what is asked to be printed; they incorrectly assume that the computer keeps a record of what is printed somewhere in its memory. Beginning programmers need explicit comparative training for the two types of PRINT statements. Training can be at the terminal so that learners observe the differences in output; training can include visual representations for memory spaces showing no change after statement execution, verbal descriptions of the key transactions (as listed in Tables 3 and 4), and role playing by the learner.

The present study has focused on diagnosis of bugs in novices' mental models for BASIC statements. The specific diagnosis of what users do not know--at the level of missing or incorrect transactions--allows us to develop individual instructional techniques for remediation. Thus, the next logical step in this project is to determine whether instructional techniques can be developed to correct users' mental models for BASIC.

# References

Luftig, M. <u>Computer programmer aptitude tests</u>. New York: Arco, 1980.

Marcus, J. <u>Basic in six hours: A self-instruction text</u>. Santa Barbara, CA: Microcomputer Laboratory, 1980.

Mayer, R. E. A psychology of learning BASIC. <u>Communications of the ACM</u>, 1979, <u>11</u>, 589-593.

Mayer, R. E. The psychology of how novices learn computer programming. <u>Computing Surveys</u>, 1981, <u>1</u>, 121-141.

Mayer, R. E., and Bayman, P. Psychology of calculator languages: A framework for describing differences in users' knowledge. <u>Communications of the ACM</u>, 1981, <u>24</u>, 511-520.

Moran, T. P. An applied psychology of the user. <u>Computing Surveys</u>, 1981, <u>1</u>, 1-11.

Footnotes

This project was supported by Grant NIE-G80-0118 from the National Institute of Education, Program in Teaching and Learning. Richard Welker assisted in data tabulation. Requests for reprints should be sent to: Piraye Bayman or Richard E. Mayer, Department of Psychology, University of California, Santa Barbara, CA 93106.

[1] We wish to thank Jeffrey Marcus and the staff of the Microcomputer Laboratory at the University of California, Santa Barbara, for their assistance on this project.

[2] The test booklets used in this study included additional questions that were not analyzed, and do not influence the present results. Appendices include instructions for the posttests as well as a sample page and a sample protocol from either posttest.

[3] Since scoring was literal, there was no need for an additional judge.

[4] Some unnecessary, incomplete and incorrect transactions are excluded for purpose of clarity. However, detailed versions of these tables can be obtained from the authors upon request.

Table 1

Proportions of Users Who Performed Each Transaction for 10 A=B+1

| Initial Test | Final Test | |
|---|---|---|
| | | Incorrect Transactions |
| .43 | .17 | Write A=B+1 in memory or in memory space A. |
| .23 | .17 | Print A=B+1 or A or the value of A on the screen. |
| .13 | .17 | Solve the equation in the statement. |
| | | Incomplete Transaction |
| .33 | .20 | Write B+1 in memory space A. |
| | | Set of Correct Transactions |
| .2[illegible] | --- | Find the number in memory space A. |
| .03 | --- | Erase the number in memory space A. |
| .05 | --- | Find the number in memory space B. |
| .03 | --- | Add 1 to the number in memory space B. |
| .30 | .33 | ** Write the obtained value in memory space A. |
| .27 | .60 | Find (wait for) the next statement to be entered in the keyboard. |

*Note.* - Double asterisk(**) indicates the key correct transaction.
Dash (---) indicates transaction could not be evaluated from the test.

Table 2

Proportion of Users Who Produced Each Transaction for LET D=0

| Verbal Test | Visual Test | |
|---|---|---|
| | | Incorrect Transactions |
| .47 | .10 | Write the equation in memory. |
| .07 | --- | Solve the equation in the statement. |
| .07 | .13 | Print the equation on the screen. |
| | | Incomplete Transaction |
| .13 | .03 | Write D or 0 in memory. |
| | | Set of Correct Transactions |
| .07 | --- | Find the number in memory space D. |
| .03 | --- | Erase the number in memory space D. |
| .47 | .77 | **Write 0 in memory space D. |
| .17 | --- | Find the next statement to be entered in the keyboard. |

Note. --Double asterisk(**) indicates the key correct transaction.

Dash (---) indicates transaction could not be evaluated from the test.

Table 3

Proportion of Users Who Produced Each Transaction for PRINT C

| Verbal Test | Visual Test | |
|---|---|---|
| | | Incorrect Transactions |
| .33 | .10 | Print the letter C on the screen. |
| .07 | --- | Print either error or nothing on the screen. |
| .07 | .17 | Write C in memory. |
| | | Incomplete Transaction |
| .03 | --- | Print |
| | | Set of Correct Transactions |
| .17 | --- | Find the number in memory space C. |
| .30 | .60 | ** Print the number or zero on the screen. |
| .10 | --- | Find the next statement to be entered in the keyboard. |

Note.-- Double asterisk (**) indicates the key correct transaction.
Dash (---) indicates transaction could not be evaluated from the test.

Table 4

Proportion of Users Who Produced Each Transaction for PRINT "C"

| Verbal Test | Visual Test | |
|---|---|---|
| | | Incorrect Transactions |
| .07 | .17 | Print the value of C on the screen |
| .07 | .03 | Write C in memory |
| .03 | --- | Find the number in memory space C |
| | | Incomplete Transaction |
| .07 | --- | Do not print the value of C on the screen. |
| | | Set of Correct Transactions |
| .07 | --- | Find the letter C in quotes in the statement. |
| .83 | .47 | **Print C on the screen. |
| .10 | --- | Find the next statement to be entered in the keyboard. |

Note. -- Double asterisk (**) indicates the key correct transaction. Dash (---) indicates transaction could not be evaluated from the test.

Table 5

Proportion of Users who Produced Each Transaction for INPUT A

| Verbal Test | Visual Test | |
|---|---|---|
| | | Incorrect Transaction |
| .30 | .17 | Write A in memory or data list. |
| .03 | .20 | Print the data or A on the screen. |
| | | Incomplete Transactions |
| .13 | --- | Wait for some data or number or A. |
| .07 | .10 | Write a number. |
| | | Set of Correct Transactions |
| .07 | .10 | ** Print ? on the screen |
| .23 | --- | ** Wait for the number and the <CR> to be entered from the keyboard. |
| .00 | --- | Find the number entered in the keyboard. |
| .03 | --- | Find the number in memory space A. |
| .03 | --- | Erase the number in memory space A. |
| .03 | --- | ** Write the number just entered in memory space A. |
| .10 | --- | Find the next statement to be entered in the keyboard. |

Note. - Double asterisk (**) indicates the key correct transaction.
Dash (---) indicates transaction could not be evaluated from the test.

Table 6

Proportion of Users Who Produced Each Transaction for IF A<B GOTO 99

| Verbal Test | Visual Test | |
|---|---|---|
| | | Incorrect Transactions |
| .20 | .13 | Print number 99, line 99, or error or the screen. |
| .13 | --- | If A is less than B, then find number 99 |
| .10 | .03 | Write A or B or A is less than B in memory. |
| .10 | --- | Find line 99 in the program |
| | | Set of Correct Transactions |
| .33 | --- | Find the number in memory space A. |
| .33 | --- | Find the number in memory space B. |
| .43 | --- | Test if the number in memory space A is less than the number in memory space B. |
| .63 | --- | ** If the value of A is less than the value in B, then move to line 99 in the program. |
| .53 | --- | ** If the value of A is not less than the value of B, then move on to the next statement in the program. |
| .33 | --- | Continue with the execution of the program from there. |

Note. -- Double asterisk (**) indicates the key correct transaction.
Dash (---) indicates transaction could not be evaluated from the test.

Table 7

Proportion of Users Who Produced Each Transaction for 20 DATA 80, 90, 99

| Verbal Test | Visual Test | |
|---|---|---|
| | | Incorrect Transaction |
| .13 | .13 | Print the numbers on the screen. |
| | | Incomplete Transaction |
| .27 | .30 | Put the data in memory space A or in memory |
| | | Set of Correct Transactions |
| .03 | --- | Find the numbers in the statement |
| .27 | .60 | **Put the numbers in memory or in the input queue. |
| .00 | --- | Find the next statement in the program. |

Note. --Double asterisk (**) indicates the key correct transaction.

Dash (---) indicates transaction could not be evaluated from the test.

Table 8

Proportion of Users Who Produced Each Transaction for READ A

| Verbal Test | Visual Test | |
|---|---|---|
| | | Incorrect Transactions |
| .10 | .07 | Print the value of A on the screen. |
| .03 | .07 | Write A in memory. |
| .03 | --- | Wait for a value to be entered from the keyboard. |
| | | Set of Correct Transactions |
| .07 | --- | Find the DATA statement in the program. |
| .07 | --- | Find the number in memory space A. |
| .00 | --- | Erase the number in memory space A. |
| .10 | .13 | **Write the first number from the DATA statement in memory space A. |
| .10 | --- | Find the next statement in the program |

Note. -- Double asterisk (**) indicates the key correct transaction. Dash (---) indicates transaction could not be evaluated from the test.

Table 9

Proportion of Users Who Produced Each Transaction for 60 GOTO 30

| Verbal Test | Visual Test | |
|---|---|---|
| | | Incorrect Transactions |
| .07 | --- | If A does not equal to X or 99 find 30. |
| .03 | .10 | Print line 30 on the screen. |
| | | Set of Correct Transactions |
| .00 | --- | Find the line number in the statement. |
| .6[illegible] | --- | **Move to line 30 in the program. |
| .37 | --- | **Continue with program execution from that line. |

Note. --Double asterisk (**) indicates the key correct transaction.

Dash (---) indicates transaction could not be evaluated from the test.

Table 10

Proportion of Users with Correct, Incomplete, Incorrect, and Empty

Conceptions for the Nine BASIC Statements from the Verbal Test

| | Conceptions | | | |
|---|---|---|---|---|
| Statements | Correct | Incomplete | Incorrect | Empty |
| INPUT A | .03 | .30 | .30 | .37 |
| 30 READ A | .10 | .27 | .17 | .47 |
| IF A < B GOTO 99 | .27 | .27 | .40 | .07 |
| LET A = B + 1 | .27 | .10 | .60 | .03 |
| 20 DATA 80, 90, 99 | .27 | .17 | .13 | .43 |
| 60 GOTO 30 | .27 | .56 | .10 | .07 |
| PRINT C | .33 | .00 | .47 | .20 |
| LET D = 0 | .43 | .03 | .53 | .00 |
| PRINT "C" | .80 | .00 | .13 | .07 |

Table 11

Proportion of Users with Correct, Incomplete, Incorrect, and Empty Conceptions [illegible] the Seven BASIC Statements from the Visual Test

| | Conceptions | | | |
|---|---|---|---|---|
| Statements | Correct | Incomplete | Incorrect | Empty |
| INPUT A | .00 | .00 | .77 | .23 |
| 30 READ A | .13 | .07 | .43 | .37 |
| LET A = B + 1 | .30 | .20 | .43 | .07 |
| PRINT "C" | .37 | .00 | .53 | .10 |
| PRINT C | .50 | .00 | .47 | .03 |
| LET D = 0 | .60 | .07 | .30 | .03 |
| 20 DATA 80, 90, 99 | .60 | .17 | .17 | .07 |

Table 12

The Correlation Matrix for Pretest Scores and the Number of Correct Conceptions for the Nine BASIC Statements

| | MIEBT | PAT-1 | PAT-2 | PAT-3 | Total Correct |
|---|---|---|---|---|---|
| MIEBT | 1.00 | | | | |
| PAT-1 | 0.33 | 1.00 | | | |
| PAT-2 | 0.49* | 0.22 | 1.00 | | |
| PAT-3 | 0.05 | -0.03 | 0.39* | 1.00 | |
| Total Correct | 0.24 | 0.14 | 0.43* | 0.04 | 1.00 |

Note. Asterisk (*) indicates $p < .05$.

# Appendices

# Appendix A

## TEST 1

Instructions

Suppose that you type in a one line command at the computer keyboard, and then pressed the RETURN key, such as:

PRINT 3 + 3 <CR>

Can you list, in English, all of the steps that the computer must go through in order to carry out your command? Your job is to translate the BASIC command into a list of steps written in plain English. List the steps in English sentences, as if you were telling the computer what to do, step by step. Try to break the command down so that each step involves just one simple action.

For example, try to list the steps for the command PRINT 3 + 3, by filling in as many of the following lines as you need:

1. ______________________________________
2. ______________________________________
3. ______________________________________
4. ______________________________________
5. ______________________________________

Write each step as an English sentence. Write the first step as a sentence on line one; write the next step as a sentence on line two, and so on. Use as many lines as you need. If you need more lines, just add some to the sheet; if you need less, just ignore the extra lines.

In a moment, you will be given a booklet of problems. Each page contains one BASIC command. Assume that you type in that command followed by a RETURN key. Your job is to list all of the steps that the computer must go through in order to carry out the command. List the steps in plain English, with each step giving one simple action, as if you were telling the computer what to do. Feel free to use less or more lines than are given for each problem. When you finish listing the steps for one command, turn the page and go on to the next. Assume that you are starting fresh on each command and that the computer has no memory for earlier pages in the booklet. Once you go on to a new problem you may not go back to any previous problem. There is no time limit. Please take as much time as you need to do a complete and accurate job. Do you have any questions?

Appendix B

Assume that you type in this command, followed by RETURN.

LET A = B + 1 <CR>

The steps carried out for this command are:

1. ______________________________

2. ______________________________

3. ______________________________

4. ______________________________

5. ______________________________

6. ______________________________

7. ______________________________

8. ______________________________

(If you need more lines just add them to this sheet; if you need less, just ignore the extra lines.)

WHEN YOU FINISH THIS PROBLEM GO ON TO THE NEXT PAGE.

YOU MAY NOT SKIP AHEAD OR GO BACK TO PREVIOUS PAGES.

[illegible] that you type in this command, followed by RETURN.

LET A = B + 1 <CR>

The steps carried out for this command are:

1. Prints what is typed on screen.
2. Understands what the word LET means
3. Solves the equation with information given
4. Gives a solution to problem.
5.
6.
7.
8.

(If you need more lines just add them to this sheet; if you need less, just ignore the extra lines.)

WHEN YOU FINISH THIS PROBLEM GO ON TO THE NEXT PAGE.

YOU MAY NOT SKIP AHEAD OR GO BACK TO PREVIOUS PAGES.

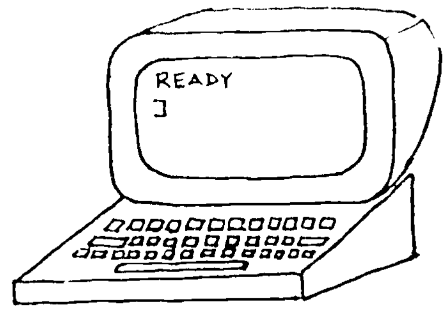
Appendix D

TEST 2

Instructions

For the pur[illegible] test, we want you to study four components of the computer.

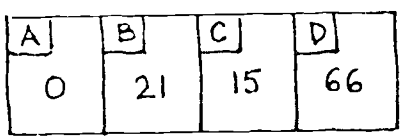
1. The display screen and the keyboard:




The display screen may look like the above, where "READY" indicates that the computer is ready to accept any command or information that you might type and the cursor "]" indicates where you are on the screen. Whatever you type in and whatever the computer replies with get displayed through this screen.

The keyboard is like a typewriter. It consists of keys for letters; digits; special characters, such as *, +, =, and so on; and special commands, such as RETURN and CONTROL.

2. The memory:

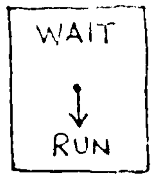



Suppose that the memory of the computer consists of boxes labeled A, B, C, D, and so on, with a number or some other information stored in each. In the above diagram, box labeled A contains the number 0, box labeled B contains the number 21, box C contains the number 15, and box D contains the number 66.

3. The input system:


Numbers wait in line to be processed in the input system. When a number is processed it drops down to the finished area. In the above diagram, the numbers 5, 99, 6 are waiting to be processed, whereas the number 7 has already been processed. The next number in line to be processed is 6.

4. The control system:

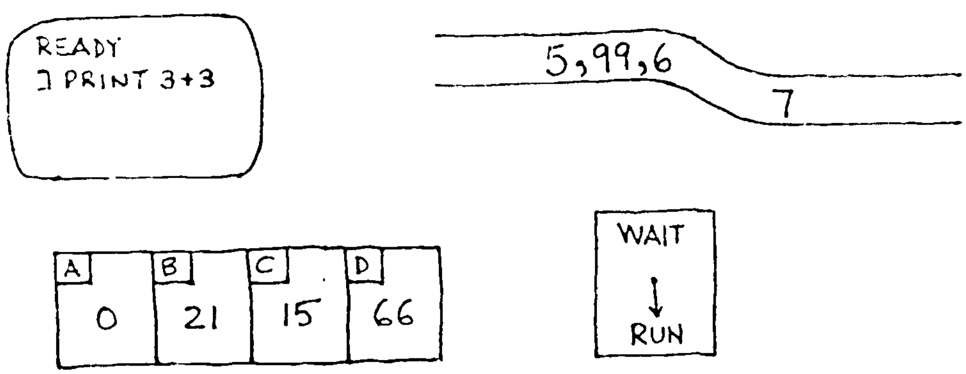



When the computer is waiting for you to type in something, the arrow points to WAIT and when the computer is busy carrying out commands, the arrow points to RUN. In the above diagram, the computer is "running".

Now that you have been introduced to four parts of the computer, you are ready to try a problem. Suppose that you typed in a one line command at the computer keyboard, and then pressed the RETURN key, such as,

PRINT 3 + 3 <CR>

Before the computer carries out this command (i.e., right after you press RETURN), the parts of the computer look like this:

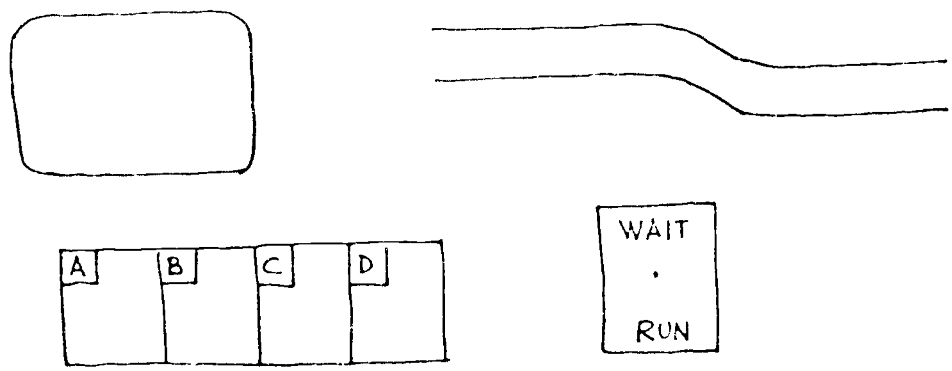



Can you indicate how the parts of the computer will look like after this command has been carried out? Just fill in the diagram on the next page.

You should feel free to write in numbers in the memory part and the input part, to move the direction of the arrow in the control part, and write in numbers and/or words in the display screen. Your job is to describe what you would see in each part of the computer after the command has been carried out. Of course, some parts may not change even though the command has been executed. Also, feel free to add any comments.

After the command has been executed, the parts of the computer look like this.

Note that you typed in PRINT 3 + 3 and then pressed the RETURN key.




In a moment, you will be given a booklet of problems. Each page contains one BASIC command. Assume that you type in the command followed by RETURN each time. Your job is to fill in the diagram so that it corresponds to the way the parts of the computer will look like after the command is executed. When you finish filling in the diagram, turn the page and go on to the next problem. Assume that you are starting out fresh on each command and that the computer has no memory for earlier pages in the booklet. Once you go on to a new problem you may not go back to any previous problem. There is no time limit. Please take as much time as you need to do a complete and accurate job. Do you have any questions?

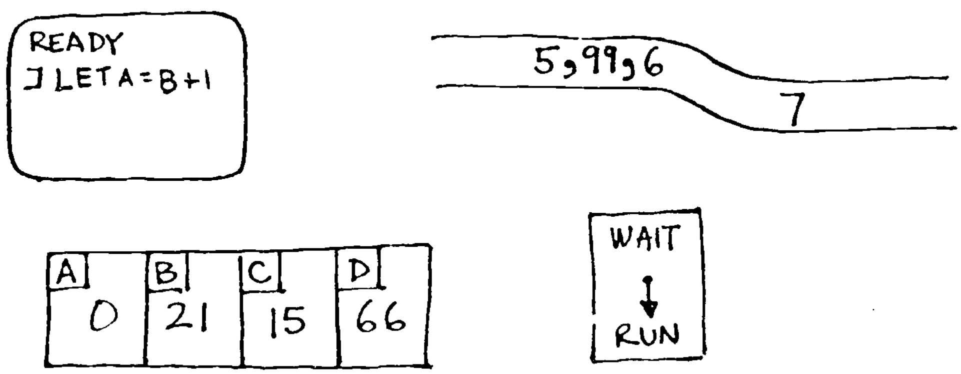
Appendix E

Assume that you type in this command, followed by RETURN.

LET A = B + 1 <CR>

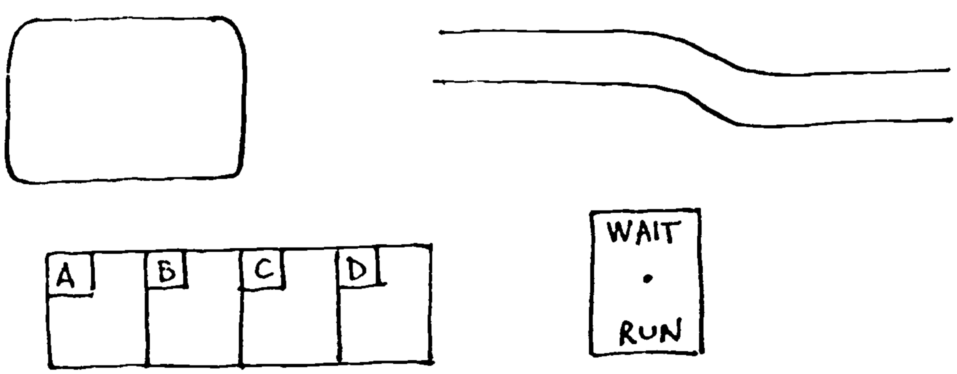
---

Before this command is executed (right after you press RETURN), the parts of the computer look like this:




---

After this command has been executed, the parts of the computer look like this:




---

WHEN YOU FINISH THIS PROBLEM GO ON TO THE NEXT PAGE.

YOU MAY NOT SKIP AHEAD OR GO BACK TO PREVIOUS PAGES.

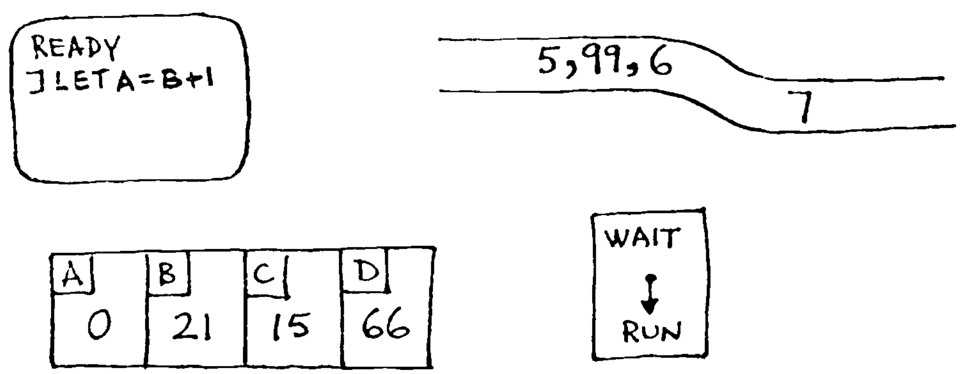
Appendix F

Assume that you type in this command, followed by RETURN.

LET A = B + 1 <CR>

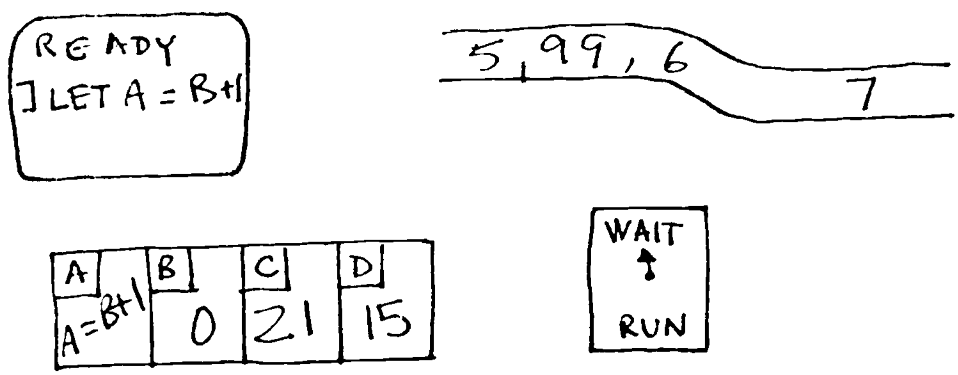
---

Before this command is executed (right after you press RETURN), the parts of the computer look like this:




---

After this command has been executed, the parts of the computer look like this:




---

WHEN YOU FINISH THIS PROBLEM GO ON TO THE NEXT PAGE.

YOU MAY NOT SKIP AHEAD OR GO BACK TO PREVIOUS PAGES.