A COMPUTERIZED LABORATORY FOR THE DYNAMICS
UNIT IN HIGH SCHOOL PHYSICS

A Graduate Project
Presented
by
David C. Perkins

In Partial Fulfillment of
The Requirements for the

Master of Science Degree
in
COMPUTER SCIENCE
at

CORPUS CHRISTI STATE UNIVERSITY
Corpus Christi, Texas
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CHAPTER ONE - OVERVIEW</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>CHAPTER TWO - USE OF THE SYSTEM</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>CHAPTER THREE - SYSTEM PROGRAMS</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>ED</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>TYPE</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>CONTROL</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>ID</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>TUTORIAL</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>COLLECT DATA</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>TIMER</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>ANALYZE RESULTS</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>CHAIN</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>CHAPTER FOUR - SYSTEMS FILES</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>CLASS ROLL</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>TUTORIAL DATA</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>SHAPE n</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>APPENDIX ONE - FLOWCHARTS</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>APPENDIX TWO - DATA DICTIONARY</td>
<td>33</td>
</tr>
<tr>
<td>7</td>
<td>APPENDIX THREE - PROGRAM LISTINGS</td>
<td>37</td>
</tr>
</tbody>
</table>
CHAPTER ONE - OVERVIEW

In the Dynamics Laboratory, one is concerned with describing the forces which cause changes in the motion of a body. In order to do this, one must be able to describe the motion of the body.

The simplest motion to describe is that produced by a constant force. This is also the simplest motion to produce, as all one has to do is drop an object with a relatively high ratio of mass to surface area, and one has the desired motion.

Most elementary physics laboratories include a dynamics unit describing motion caused by a constant force, due to the ease in producing and describing constant motion. However, most high school physics laboratories use either a spark timer or a vibrating timer to time the motion. Both of these timers work by dragging a tape through a mechanism which makes marks on it. Both introduce forces into the experiment which are not constant and are not taken into consideration in the analysis of the data. Therefore, students in the laboratories usually find it very difficult, if not impossible, to produce accurate results.
This project, entitled A COMPUTERIZED LABORATORY FOR THE DYNAMICS UNIT IN HIGH SCHOOL PHYSICS, hereafter, referred to as the system, has been designed to use a laser, a photoelectric detector, and an APPLE II computer as a timer, and thus to eliminate forces introduced into the dynamics experiment by the laboratory equipment.

In addition to collecting times for the student, the system tutors the student in how to conduct the experiment and analyze the data.

The system consists of the programs ED, TYPE, CONTROL, ID, TUTORIAL, COLLECT DATA, TIMER, and ANALYZE RESULTS. The program "CHAIN" which is provided by APPLE on the Applesoft DOS 3.2.1 system diskette is used by the system.

The system uses the files CLASS ROLL, TUTORIAL DATA, and SHAPE n. (SHAPE n is a group of files.)

A structure chart showing the relations between programs and files is shown in Figure 1.

The remainder of this report is concerned with describing the use of the system, the programs, and the files. Appendices are included which contain flowcharts, listings, and an abbreviated data dictionary.
FIGURE 1 - System Structure Chart
CHAPTER TWO - USE OF THE SYSTEM

In order to use this system to conduct an experiment, the instructor must make certain preparations. He must set up the equipment and hook up the computer. He must, also, make up a diskette with the following programs and files:

Programs
CONTROL
ID
TUTORIAL
COLLECT DATA
TIMER
ANALYZE RESULTS
CHAIN

Files
CLASS ROLL
TUTORIAL DATA
SHAPE ONE  The number of shape files depends on the number of shape tables in the graphics routines.
SHAPE TWO
... SHAPE n

Before the student sits down at the computer, the instructor should have loaded the above diskette and booted DOS. He should, then, type

RUN CONTROL (CR)

to load the program CONTROL and pass control to it. At this point the student should take over.
First, the student will be taken to ID, where he will be prompted for his name. The CLASS ROLL file will be searched to see if he is an authorized user. To allow for no match caused by typing errors, the student may enter his name up to three times in the identification process. If the student fails the identification process, he will be referred to the instructor; and control will exit from the system to Applesoft.

If the student passes the identification process, he will be taken to the TUTORIAL program, where he will be given instructions on how the experiment is to be run and on how the data is to be analyzed. He may go through this instruction up to three times, if he has difficulty understanding it. If he goes through the instruction three times and still indicates that he does not understand, he will be referred to the instructor and control will exit from the system to Applesoft.

After TUTORIAL, the student will be taken to COLLECT DATA, where the collection of timing data will be accomplished. COLLECT DATA will write the times to the student's record in CLASS ROLL and will, then, set the exit flag - XF% - to 1.
After COLLECT DATA, the student will leave the system to analyze the data. Typically, this is an overnight process. After the data has been analyzed by the student, he will return to the system and will again be sent through ID. CONTROL will then take the student to ANALYZE RESULTS, where the student will input his results, and they will be compared to the system's results. The student will be informed of the result of this comparison. The student's results will be written to his record in CLASS ROLL, and he will be taken back to CONTROL, where control will return to Apple-soft. If another student wishes to run an experiment, he will have to have the instructor set up the experiment for him.

For a graphical illustration of the above discussion, refer to FLOWCHART ONE - The System Flowchart in appendix one.
CHAPTER THREE - SYSTEM PROGRAMS

ED

ED was designed as a simple, general purpose text editor that affords maximum ease of use and versatility with minimum training time. ED recognizes seven commands. These commands are:

K - Delete a line of text. After the line is deleted, the editor shifts all following lines in the file up one position to fill in the empty line.

I - Insert lines of text. To end the insert, a control-Z must be entered.

+B - Move to the beginning of the file.

-B - Move to the end of the file.

+n - Move n lines toward the end of the file.

-n - Move n lines toward the beginning of the file.

E - End the edit. The user is given the option to write to the file by a prompt.

To use ED to write a text file, the user should have booted DOS and placed a diskette that has ED on it in the disk drive. The user should then type

RUN ED (CR)

to cause the ED program to be loaded and have control passed to it.
The user will be prompted for the name and type of the file to be edited. After the file name and type is given, ED looks for it on the diskette. If the file is found, it is read into an array; and the user is notified that the file is an old file. If the file is not found, ED responds with "NEW FILE".

At this point, the user is in a position to look at any portion of an old file or edit it. He is in a position to insert, if the file is a new one.

After the user has completed the editing of the file, he should enter E to end the edit. He will be prompted with

WRITE TO FILE - Y/N?

If he enters Y, the file, along with any changes he has made to it, will be written to disk. If he enters N, the file on disk will not be changed.

TYPE

TYPE was designed to allow the user to look at and list text files without having to run ED. To use TYPE, the user should have docted DOS and should have a diskette with TYPE on it in the disk drive. The user should then type
RUN TYPE (CR)
to cause the program to be loaded and have control given to it.

The user will be prompted to enter the name of the file to be typed. Then, the file will be read into an array, and the user will again be prompted with

OUTPUT TO PRINTER - Y/N?
If the user responds with Y, output will be sent to both the monitor and to slot #2, which should contain a printer card. If the user enters N, output will be sent to the monitor only.

CONTROL

CONTROL was designed to direct the flow of control between programs in the system. CONTROL chains to ID, TUTORIAL, COLLECT DATA, and ANALYZE RESULTS when their services are needed, and these programs all chain back to CONTROL.

Logically, CONTROL acts as a main program and ID, TUTORIAL, COLLECT DATA, and ANALYZE RESULTS act as subprograms. The program "CHAIN" logically acts as a subroutine call when used by CONTROL, and as a RETURN statement when used by ID, TUTORIAL, COLLECT DATA, and ANALYZE RESULTS.
When CONTROL is first entered by the instructor typing "RUN CONTROL (CR)", all real and integer variables are initialized to 0; and all string variables are initialized to the null string by Applesoft. Thus, on the first test of PN%, it is zero; and CONTROL prompts the student with:

IF YOU WISH TO RUN A PHYSICS EXPERIMENT PRESS THE SPACE BAR.

When the student presses any key, CONTROL chains to ID.

When ID chains back to CONTROL, it will return a 1 or 4 in PN% to indicate to CONTROL that the student passed or failed, respectively, the identification process. If PN% is 1, CONTROL will chain to TUTORIAL. If PN% is 4, CONTROL will transfer control to Applesoft.

When TUTORIAL chains back to CONTROL, it will return a 2 or 4 in PN% to indicate to CONTROL that the student has understood or not understood, respectively, the tutorial process. If PN% contains a 2, CONTROL will chain to COLLECT DATA; and if it contains a 4, CONTROL will transfer control to Applesoft.

When COLLECT DATA chains back to CONTROL, it will return a 3 in PN%. It will, also, set XF%, the exit flag, to 1. When CONTROL determines that PN% contains
a 3, it will check XF% and, noting that it is 1, will print on the monitor:

    EXITING EXPERIMENT TO GIVE YOU
    TIME TO ANALYZE THE DATA,

and then transfer control to Applesoft.

When the student returns to the system, after he has analyzed the data, CONTROL will see a 0 in PN% and will, thus, chain to ID. On reading the student's record, ID will read a 3 into PN% so that when ID chains back to CONTROL it will detect a 3 in PN% and a 0 in XF%. CONTROL will, then, chain to ANALYZE RESULTS.

When ANALYZE RESULTS chains back to CONTROL, it will return a 4 in PN% so that CONTROL will transfer control to Applesoft.

ID

When ID is chained to by CONTROL, it prompts the student with a request for his name. After the student has entered his name, ID gives him a chance to correct any errors that he notices at that time. If the student responds that the name is correct, ID, then, searches the file CLASS ROLL for the student's record. If the record is
not found, ID will give the student up to two more chances to enter his name before it returns to CONTROL with a 4 in PN%. If the record is found, ID will read it and put a 1 in PN%. ID will, then, chain back to CONTROL.

In addition to its main function of identifying authorized student users, ID is also used to initialize several variables.

TUTORIAL

TUTORIAL is used by the system to instruct the student on the experimental procedure and the method that is to be used to analyze the data that is collected by the run of the experiment. When it is chained to by CONTROL, it reads the file TUTORIAL DATA into a string array.

After the file has been read, TUTORIAL looks at each string in the array one at a time. After it looks at a string, it will print it to the monitor unless it is the word "PAUSE" or the word "GRAPHICS". If "PAUSE" is encountered, then TUTORIAL will pause to allow the student time to study the material on the monitor. To continue, the student will have to press a key on the keyboard. If "GRAPHICS" is
encountered, TUTORIAL will pick up the next string, which is a parameter, and then will go to the proper graphics subroutine.

The graphics subroutines are written to aid in the tutorial process. They are used in this system to draw figures of the equipment that is used and to draw the graphs that should result from a proper analysis of the data.

All of the graphics used in this system are the Apple high resolution graphics. The figures and graphs are drawn using a combination of the H PLOT function and the DRAW and X DRAW functions.

When the end of the array is reached, TUTORIAL will prompt the student to see if he understands the material just presented. If the student does, then TUTORIAL will chain back to CONTROL after placing a 2 into PN%. If the student does not understand, he will be given up to two more passes through the material. If, after three times through the material, the student still does not understand it, a 4 will be placed in PN%. The student will be told to see the instructor, and TUTORIAL will chain back to CONTROL.
COLLECT DATA

When CONTROL chains to COLLECT DATA, the student is ready to collect times. First, COLLECT DATA initializes to 0 the memory locations used by TIMER to hold the time counts. It, then passes to TIMER the number of time intervals to be collected, and the time factor to be used.

After the initialization is complete, COLLECT DATA will prompt the student to place a steel ball under the electromagnet and to push the button on the electromagnet to allow the ball to drop and times to be collected. Then, the machine language program TIMER will be called by COLLECT DATA.

When the student pushes the button on the electromagnet, the ball will drop and times will be collected by TIMER. After the last time interval is taken, TIMER will return to COLLECT DATA.

COLLECT DATA will convert the time counts returned by TIMER into times in seconds. It will, then, write the times to the student's record in CLASS ROLL, set PN% to 3, write it to the student's record, and set XF% to 1.

COLLECT DATA will, then, chain back to CONTROL.
TIMER

TIMER is a machine language program called by COLLECT DATA to time the intervals between beam interruptions caused by an object moving in a predetermined path.

It starts collecting times when the first beam interruption occurs. It does this by putting the time factor passed by COLLECT DATA into the Y-INDEX register and decrementing it to provide a time scale. After the Y-INDEX register has been decremented to zero, a memory location addressed by the X-INDEX register, is incremented by one. This process is repeated until another beam interruption occurs. Then, the value in the Y-INDEX register is stored, the X-INDEX register is incremented by one, and the process is repeated until all intervals have been timed.

After all intervals have been timed, TIMER returns control to COLLECT DATA.

ANALYZE RESULTS

When CONTROL chains to ANALYZE RESULTS, the student will be prompted to enter the numerical results he computed from his analysis of the data. ANALYZE
RESULTS will compare the student's numerical results with those that he should have determined; and if the difference is less than 10% of the system's result, it will congratulate him, write his result to his record in CLASS ROLL, set PN% to 4, and chain to CONTROL. If his results are not within 10% agreement, the student will be given the correct results. His results will be written to his record in CLASS ROLL, PN% will be set to 4, and ANALYZE RESULTS will chain back to CONTROL. If the student's results are not acceptable, he will be given a chance to run the experiment again. However, the instructor will have to reinitialize the times in the student's record to zero before he will be allowed to collect data again. Thus, the instructor will know of the student's previous attempt and its results.

CHAIN

CHAIN is provided by APPLE on the DOS 3.2.1 system diskette. It is loaded and called whenever one program of the system passes control to another program of the system. It allows chaining of the programs without loss of variables.
CHAPTER FOUR - SYSTEM FILES

CLASS ROLL

The file CLASS ROLL has two record types. The first is a general information record and has only one occurrence. It is:

\[
\begin{align*}
\text{TEACH$}, & \quad \text{TITLE$}, \quad \text{NI$}, \quad \text{TC$}, \quad \text{TF} \\
\text{DIST}(1) & \\
\text{DIST}(2) & \\
\vdots & \\
\text{DIST(NI$)}. & 
\end{align*}
\]

\text{TEACH$} is the name of the teacher. \text{TITLE$} is the teacher’s title. (i.e - MR, MRS, MISS, MS, DR). \text{NI$} is the number of intervals to be timed. (1 to 16). \text{TC$} is a timing constant used by the subroutine \text{TIMER}. (1 to 255). \text{TF} is the timing factor used to convert the count from \text{TIMER} to times in seconds. \text{DIST}(1), \text{DIST}(2), ..., \text{DIST(NI$)} are the distances corresponding to the times.

The second record type is the student record. It has an occurrence for each student that is authorized to use the program. It is:

\[
\begin{align*}
\text{LN$}, & \quad \text{GN$}, \quad \text{PN$}, \quad \text{ACC}, \quad \text{SACC} \\
\text{TIME}(1) & \\
\text{TIME}(2) & \\
\vdots & \\
\text{TIME(NI$)}. & 
\end{align*}
\]
LN$ is the student's last name. GN$ is the student's first name. PN% is the program number. ACC is the acceleration computed by the student from the data. SACC is the acceleration computed by the system from the data. TIME(1), TIME(2),... , TIME(NI%) are the times to the end of each interval.

The CLASS ROLL file must be initialized, by the instructor, using ED. PN%, ACC, SACC, and the array TIME must be initialized to zero.

TUTORIAL DATA

The file TUTORIAL DATA contains the text used by TUTORIAL. It has only one record type. It is:

ST$.

TUTORIAL DATA should be initialized using ED. In initializing TUTORIAL DATA, the instructor should be aware of the fact that to cause the flow of text to stop, so that the student has a chance to study the text, the value in ST$ should be "PAUSE". He should, also, be aware that to use graphics he should use the pair:

"GRAPHICS"
"n",

where n is the number of the graphics subroutine.
It will, also, be necessary for the instructor to code a graphics routine tailored to his needs.

SHAPE n

The group of files named SHAPE n are binary data files. They contain the shape tables used by the graphics subroutines of the TUTORIAL program. A shape table contains information about the table and shape definitions. The shape definitions are binary code, describing a series of horizontal and vertical unit vectors. The vectors have codes to cause a point to be plotted and a move made or to just make a move.
APPENDIX ONE - FLOWCHARTS

1. SYSTEM --------- 21
2. ED -------------- 22
3. TYPE ----------- 23
4. CONTROL ------ 24
5. ID --------------- 25
6. TUTORIAL ------- 26
7. COLLECT DATA -- 28
8. ANALYZE RESULTS --- 30
9. TIMER --------- 31
FLOWCHART ONE - System Flowchart
FLOWCHART TWO - ED
START -> INPUT FILE NAME -> INITIALIZE -> READ FILE INTO ARRAY

NO: OUTPUT TO PRINTER? -> PR ≠ ∅

YES: PRINT ARRAY -> PR ≠ ∅

NO: ANOTHER FILE?

YES: PR # 2

NO: END
FLOWCHART FOUR - CONTROL
START
ID

INITIALIZE

A

IF
J > 3

THEN

 ELSE

ENTR (MESSAGE); GN$
EUTF (MESSAGE); LN$

IF
NAME IS CORRECT

THEN

READY CLASS
ROLL TO READ
STUDENT
RECORDS.

C

ELSE

PN% ← 1

B

IF

ELSE

IS NM%

THEN

ELSE

J = 3

THEN

PRINT GN$; "I
COULD NOT FIND
YOUR RECORD.
PRESS SPACE
BAR TO TRY
AGAIN."

PRINT
"PRESS
SPACE BAR."

I ← I + 1

GET B$ 3 ← I + 1

A

C

READ
STUDENT
RECORD.

IF
FLG=LN$ AND
GN$=GN$

THEN

ELSE

PN% = φ

THEN

PN% = 1

B

END ID

FLOWCHART FIVE - ID
FLOWCHART SIX - Continued
START
COLLECT DATA

PN% ← 3
XF% ← 1

LOAD THE
SUBROUTINE
TIMER

INITIALIZE DATA
AREAS IN TIMER

INITIALIZE TIME
CONSTANT AND NO.
OF INTERVALS IN TIMER

RING THE BELL
AND SEE IF
STUDENT IS READY.

IF
STUDENT IS
READY.

THEN
TIMER

READY CLASS
ROLL FILE TO
READ STUDENT

DIMENSION ARRAYS
TO HOLD STUDENT
RECORDS.

1

FLOWCHART SEVEN - COLLECT DATA
1

READ STUDENT RECORDS INTO ARRAYS

FIND RECORD FOR CURRENT STUDENT USR

IF

ELSE

STUDENT TIMES ≤ 0

THEN

CONVERT TIMER COUNTS TO TIMES AND PUT IN STUDENT RECORD IN ARRAYS

READY CLASS ROLL FILE TO WRITE.

WRITE CLASS ROLL FILE

CLOSE CLASS ROLL FILE

PRINT TIMES AND DISTANCES TO MONITOR

PRINT MSG: "YOU MUST SEE TITLE: " "TEACH BEFORE YOU ATTEMPT TO ROV. EXPERIMEMT AGAIN"

END COLLECT DATA

FLOWCHART SEVEN - Continued
FLOWCHART EIGHT - ANALYZE RESULTS
FLOWCHART NINE - TIMER
1

\[ M_{2^x} \leftarrow y \]

\[ x \leftarrow x + 1 \]

\[ x = N1? \]

\[ \text{YES} \rightarrow \text{RETURN FROM TIMER} \]

\[ \text{NO} \rightarrow 2 \]
APPENDIX TWO - DATA DICTIONARY
### DATA DICTIONARY

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DESCRIPTION</th>
<th>TYPE AND SPECIAL VALUES</th>
<th>IN PROGRAM*</th>
</tr>
</thead>
<tbody>
<tr>
<td>A$</td>
<td>indicator</td>
<td>character, (Y,N)</td>
<td>1, 2, 5</td>
</tr>
<tr>
<td>ACC</td>
<td>acceleration computed by student</td>
<td>real</td>
<td>4, 6</td>
</tr>
<tr>
<td>ACC( )</td>
<td>array to hold acc's computed by student</td>
<td>real</td>
<td>6, 7</td>
</tr>
<tr>
<td>B$</td>
<td>indicator or pause</td>
<td>character, (Y,N)</td>
<td>3, 4, 5, 6, 7</td>
</tr>
<tr>
<td>BS$</td>
<td>one character, used to build up a string</td>
<td>character</td>
<td>1, 2</td>
</tr>
<tr>
<td>BL$</td>
<td>bell character</td>
<td>CHR$(7)</td>
<td>4, 5, 6, 7</td>
</tr>
<tr>
<td>BS$</td>
<td>back space</td>
<td>CHR$(8)</td>
<td>1</td>
</tr>
<tr>
<td>COUNT</td>
<td>index of FOR-NEXT</td>
<td>real</td>
<td>5</td>
</tr>
<tr>
<td>D$</td>
<td>used to implement DOS</td>
<td>CHR$(4)</td>
<td>1, 2, 3, 4, 5, 6, 7</td>
</tr>
<tr>
<td>DIST</td>
<td>dummy variable used to read first record in CLASS ROLL.</td>
<td>real</td>
<td>4</td>
</tr>
<tr>
<td>DIST( )</td>
<td>array to hold dist</td>
<td>real</td>
<td>6, 7</td>
</tr>
<tr>
<td>DT$</td>
<td>file type indicator</td>
<td>character, (Y,N)</td>
<td>1</td>
</tr>
<tr>
<td>FGN$</td>
<td>first name read from file</td>
<td>character</td>
<td>4</td>
</tr>
<tr>
<td>FILE$</td>
<td>name of file</td>
<td>character</td>
<td>1, 2</td>
</tr>
<tr>
<td>FLN$</td>
<td>last name read from file</td>
<td>character</td>
<td>4</td>
</tr>
<tr>
<td>GN$</td>
<td>first name input by student</td>
<td>character</td>
<td>4, 5, 6, 7</td>
</tr>
<tr>
<td>GN$( )</td>
<td>array of first names</td>
<td>character</td>
<td>6, 7</td>
</tr>
<tr>
<td>VARIABLE</td>
<td>DESCRIPTION</td>
<td>TYPE AND SPECIAL VALUES</td>
<td>IN PROGRAM*</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------</td>
<td>-------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>I</td>
<td>total number of file lines</td>
<td>real</td>
<td>1, 2</td>
</tr>
<tr>
<td>I</td>
<td>index in FOR-NEXT</td>
<td>real</td>
<td>5, 6, 7</td>
</tr>
<tr>
<td>J</td>
<td>current line number</td>
<td>real</td>
<td>1, 2</td>
</tr>
<tr>
<td>J%</td>
<td>number of students in CLASS ROLL file</td>
<td>integer</td>
<td>6, 7</td>
</tr>
<tr>
<td>K</td>
<td>instruction number</td>
<td>real, (1,2,3,4,5,6)</td>
<td>1</td>
</tr>
<tr>
<td>K</td>
<td>index in FOR-NEXT</td>
<td>real</td>
<td>2, 4</td>
</tr>
<tr>
<td>L</td>
<td>index in FOR-NEXT</td>
<td>real</td>
<td>1, 4, 6, 7</td>
</tr>
<tr>
<td>LC</td>
<td>line count</td>
<td>real</td>
<td>2</td>
</tr>
<tr>
<td>LN$</td>
<td>last name input by student</td>
<td>character</td>
<td>4, 6, 7</td>
</tr>
<tr>
<td>LN$( )</td>
<td>array of last names</td>
<td>character</td>
<td>6, 7</td>
</tr>
<tr>
<td>M</td>
<td>graphics parameter</td>
<td>real</td>
<td>5</td>
</tr>
<tr>
<td>NI$( )</td>
<td>string array which holds file in memory</td>
<td>character</td>
<td>1</td>
</tr>
<tr>
<td>NI%</td>
<td>number of intervals</td>
<td>integer</td>
<td>4, 6, 7</td>
</tr>
<tr>
<td>NM%</td>
<td>number of student records</td>
<td>integer</td>
<td>4</td>
</tr>
<tr>
<td>NP$</td>
<td>new page</td>
<td>CHR$(12)</td>
<td>2</td>
</tr>
<tr>
<td>NR%</td>
<td>number of lines in CLASS ROLL file</td>
<td>integer</td>
<td>6, 7</td>
</tr>
<tr>
<td>NREC</td>
<td>number of lines in TUTORIAL DATA file</td>
<td>real</td>
<td>5</td>
</tr>
<tr>
<td>PA</td>
<td>graphics flag</td>
<td>real, (0,1)</td>
<td>5</td>
</tr>
<tr>
<td>PARA$</td>
<td>graphics parameter</td>
<td>character</td>
<td>5</td>
</tr>
<tr>
<td>PC</td>
<td>page count</td>
<td>real</td>
<td>2</td>
</tr>
<tr>
<td>VARIABLE</td>
<td>DESCRIPTION</td>
<td>TYPE AND SPECIAL VALUES</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------</td>
<td>-------------------------</td>
<td></td>
</tr>
<tr>
<td>PN$</td>
<td>program number</td>
<td>integer, (0, 1, 2, 3, 4), 3, 4, 5, 6, 7</td>
<td></td>
</tr>
<tr>
<td>PN$( )</td>
<td>array of program no's</td>
<td>integer</td>
<td>6, 7</td>
</tr>
<tr>
<td>RS$</td>
<td>return</td>
<td>CHR$(13)</td>
<td>1, 2</td>
</tr>
<tr>
<td>SACC</td>
<td>acceleration computed by system</td>
<td>real</td>
<td>4</td>
</tr>
<tr>
<td>SACC( )</td>
<td>array for system computed accelerations</td>
<td>real</td>
<td>6, 7</td>
</tr>
<tr>
<td>ST$:</td>
<td>instruction variable</td>
<td>character</td>
<td>1</td>
</tr>
<tr>
<td>ST$( )</td>
<td>string array to hold file in memory</td>
<td>character</td>
<td>2, 5</td>
</tr>
<tr>
<td>S$:</td>
<td>flag - indicates if TUTORIAL DATA has been read</td>
<td>integer, (0, 1)</td>
<td>5</td>
</tr>
<tr>
<td>T:</td>
<td>time constant</td>
<td>integer(1 - 255)</td>
<td>4, 6, 7</td>
</tr>
<tr>
<td>TEACHER$:</td>
<td>name of teacher</td>
<td>character</td>
<td>4, 6, 7</td>
</tr>
<tr>
<td>TF</td>
<td>time factor</td>
<td>real</td>
<td>4, 6, 7</td>
</tr>
<tr>
<td>TIME$</td>
<td>cumulative time</td>
<td>real</td>
<td>7</td>
</tr>
<tr>
<td>TIME$( )</td>
<td>array to hold times</td>
<td>real</td>
<td>6, 7</td>
</tr>
<tr>
<td>TITLE$:</td>
<td>teacher's title</td>
<td>character</td>
<td>4, 6, 7</td>
</tr>
<tr>
<td>X$:</td>
<td>exit flag</td>
<td>integer</td>
<td>3, 6</td>
</tr>
</tbody>
</table>

*In the IN PROGRAM column above the following matching is used:

1 - ED; 2 - TYPE; 3 - CONTROL; 4 - ID; 5 - TUTORIAL; 6 - COLLECT DATA; 7 - ANALYZE RESULTS
APPENDIX THREE - PROGRAM LISTINGS*

1. ED
2. TYPE
3. CONTROL
4. ID
5. TUTORIAL
6. COLLECT DATA
7. ANALYZE RESULTS
8. TIMER

*All program listings are located in the folder pocket located on the back cover of the folder.