ABSTRACT

This document describes BASCHART, a computer aid designed to decipher and automatically flow chart computer program logic; it also provides the computer code necessary for this process. Developed to reduce the labor intensive manual process of producing a flow chart for an undocumented or inadequately documented program, BASCHART will automatically produce an annotated flow chart for any program that uses WANG (MVP) BASIC-2 programming language. The development of the BASCHART program is summarized, and the hardware requirements are described. A user's guide includes operating procedures and an example of a program flow chart. Three appendices provide (1) a system diagram showing the interrelationships between the program subsystems and a listing of the subroutines with brief descriptions of their purpose; (2) definitions of the BASCHART alphanumeric variables; and (3) the WANG BASIC-2 computer code for the flow charting program. (LMM)
BASIC Language Flow Charting Program (BASCHART)

Charles C. Johnson
Thomas O. Peeples
Gary W. Hodak

APPROVED BY:

Alfred F. Smode
Director

Training Analysis and Evaluation Group
Orlando, Florida 32813
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I INTRODUCTION</td>
<td>3</td>
</tr>
<tr>
<td>Background</td>
<td>3</td>
</tr>
<tr>
<td>Purpose</td>
<td>4</td>
</tr>
<tr>
<td>Organization of This Report</td>
<td>4</td>
</tr>
<tr>
<td>II SYSTEM DESCRIPTION AND USER'S GUIDE</td>
<td>5</td>
</tr>
<tr>
<td>Development History</td>
<td>5</td>
</tr>
<tr>
<td>Hardware Requirements</td>
<td>5</td>
</tr>
<tr>
<td>BASCHART Operating Procedures</td>
<td>5</td>
</tr>
<tr>
<td>BASCHART Program Flow Chart</td>
<td>12</td>
</tr>
<tr>
<td>APPENDIX A System Diagram of BASCHART Program and BASCHART Subroutine Listing</td>
<td>18</td>
</tr>
<tr>
<td>APPENDIX B BASCHART ALPHANUMERIC VARIABLES</td>
<td>22</td>
</tr>
<tr>
<td>APPENDIX C BASCHART Computer Code Listing</td>
<td>26</td>
</tr>
</tbody>
</table>

## LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1 Interrelationship of BASCHART Subroutines</td>
<td>19</td>
</tr>
</tbody>
</table>
Technical Note 3-82

SECTION I

INTRODUCTION

Proper computer program documentation is essential to the effective use, maintenance, and modification of computer software. The documentation should be comprised of both user and programmer information and be designed to fill the informational needs of the personnel ultimately responsible for using, maintaining or updating the software. The amount and quality of the software documentation produced are usually dependent on two major factors in the program development cycle-- the length of time allotted for development and the number of people assigned to the development effort. Normally, the longer the duration of a project and the larger the software development team, the greater the need for software documentation. Proper documentation (1) provides a means to efficiently monitor and control long-term projects, (2) provides an understandable, transferrable method of communication between the project team members and future users or program modifiers, and (3) reduces the time consuming, labor intensive process of duplicating the development effort if key project members leave or the program is moved to another site and/or the programs require modification or updating. If the software documentation is adequate, only minimal problems arise when it is modified or transferred to new locations. If the formal documentation is inadequate, or nonexistent, the new user is faced with the formidable and tedious process of manually flow charting the program to determine program logic and defining the program variables prior to proceeding with any modifications. One solution to this approach is to develop a computer aid with the capability to accurately and expeditiously decipher the program logic and automatically provide a flow chart.

BACKGROUND

The Chief of Naval Education and Training (CNET) tasked the Training Analysis and Evaluation Group (TAEG) to document selected software components of the ADP portion of the Foreign Military Sales (FMS) financial management system. The specific objectives of the tasking were to:

- describe the structure of the FMS system and its elements, along with the logic for completing summary billing
- provide an operator's guide that would improve understanding of the program logic and procedures necessary to use the system
- document the specific system programming details of the FMS FY 80 costing/billing programs.

At the outset of this effort, little formal documentation was available other than comments contained in the program software concerning program structure and use. In order to understand how the programs worked, it was necessary to determine program logic and define software variables and their interrelationships. The need to accomplish this formidable labor intensive effort as the necessary prelude to achieving the FMS study objectives served as the impetus for automating this process.
PURPOSE

This document describes a computer aid designed to decipher and automatically flow chart computer program logic and provides the computer code necessary for this process. This computer aid reduces the labor intensive manual process of producing a flow chart for an undocumented or inadequately documented program. This TAEG developed program (hereafter called BASCHART) automatically produces an annotated flow chart for any program that uses WANG (MVP) BASIC-2 programming language.

ORGANIZATION OF THIS REPORT

In addition to this introduction, one other section and three appendices are provided. Section II describes the BASCHART flow charting program and its development and provides a detailed user's guide. Appendix A provides a system diagram of the BASCHART program showing the interrelationships between the subsystems contained in the program. It also contains a listing of the subroutines with brief descriptions of their purpose. Appendix B contains definitions of the BASCHART alphanumeric variables. Appendix C provides the WANG BASIC-2 computer code for the flow charting program.
SECTION II
SYSTEM DESCRIPTION AND USER'S GUIDE

This section briefly summarizes the development of the BASCHART program, describes the hardware requirements, and contains a user's guide for the BASCHART.

DEVELOPMENT HISTORY

The WANG MVP flow charting program is an extensive modification of an existing program developed at the University of Central Florida, Orlando. In the conversion, many of the unique WANG functions were implemented to provide as complete a documentation system as possible. The original program, operated in a batch environment, was designed to work in Harris BASIC on relatively short and noncomplex programs. BASCHART is an online interactive program that can be used on any WANG BASIC-2 program regardless of program length or programmers' writing technique. The program can also isolate a given segment of a program and flow chart only that portion. The user is provided substantial flexibility because BASCHART has been modified to interact with the WANG operating system and disk catalog structure.

HARDWARE REQUIREMENTS

BASCHART is written using WANG BASIC-2 and can be operated on a WANG 2200 MVP within a 56K partition. The user should have a line printer available and must supply the BASCHART program with the program to be flow charted from either a floppy disk or a fixed disk storage medium.

BASCHART OPERATING PROCEDURES

It is assumed that the required computer hardware (CRT, DISK DRIVE, and LINE PRINTER) is available to the user intending to use the BASCHART program. Initializing the equipment is an extremely easy task. However, because of the many possible equipment configurations, it is desirable that personnel knowledgeable in WANG equipment set up the system for subsequent use. After loading and starting the execution of program BASCHART, the following screen will appear.

`888888  A  SSSSSS  CCCCCC  H  H  A  RRRRRR  TTTTTT
 8  8  A  A  S  C  H  H  A  A  R  R  T
888BBB  A  A  SSSSSS  C  HHHHHHH  A  A  RRRRRR  T
 8  8  AAAAAAA  S  C  H  H  AAAAAAA  R  R  T
 8  8  A  A  S  C  H  H  A  A  R  R  T
888BBB  A  A  SSSSSS  CCCCCC  H  H  A  R  R  T`

This program is designed to create a logic flow chart of any active program on disk in the BASIC-2 language. Is it your intention to run the program 'BASCHART'? (Y/N)
Entering an N will return the user to the WANG operating system. If the user enters a Y, the following screen will be displayed:

```
***** DISK ADDRESS YOUR PROGRAM *****

DISK ADDRESSES AVAILABLE: 325 825 365 D10 D11 D13

ENTER DISK ADDRESS __ __ __
```

The user must now enter the disk address where the program to be flowcharted is located. Entering a valid disk address will enable the program to scan the specified disk directory and automatically present the following display:

```
DISK D11 CATALOG OF PROGRAMS

10. 20. 30. 40. 50.
1. 11. 21. 31. 41.
2. 12. 22. 32. 42.
3. 13. 23. 33. 43.
4. 14. 24. 34. 44.
5. 15. 25. 35. 45.
6. 16. 26. 36. 46.
7. 17. 27. 37. 47. BASCHART
8. 18. 28. 38. 48. BASICout
9. 19. 29. 39. 49. BASchart

FN '4 = END PROGRAM / FN '5 = SEARCH / FN '6 = BASCHART PROGRAM

510 TOTAL PROGRAMS PRESS RTN/FN?
```

This display lists the programs contained at the disk address previously specified. This initial display contains the first 49 programs, after that the programs are presented in groups of 50. Pressing RETURN will cause the next group of 50 programs to appear. Repeating this process will present succeeding groups of programs until all the programs available at that disk address have been displayed. The total number of programs available on the disk is continuously displayed at the bottom of the screen.
Selecting FN '4 = END PROGRAM will return the user to the WAdG operating system.

If the user does not know the entire name of the program or its correct spelling, use the search capability of BASCHART, special function Key '5. Selecting FN '5 = SEARCH will cause the screen to display the following:

<table>
<thead>
<tr>
<th>DISK</th>
<th>O11</th>
<th>CATALOG OF PROGRAMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.</td>
<td>20.</td>
<td>30. 40.</td>
</tr>
<tr>
<td>11.</td>
<td>21.</td>
<td>31. 41.</td>
</tr>
<tr>
<td>12.</td>
<td>22.</td>
<td>32. 42.</td>
</tr>
<tr>
<td>13.</td>
<td>23.</td>
<td>33. 43.</td>
</tr>
<tr>
<td>14.</td>
<td>24.</td>
<td>34. 44.</td>
</tr>
<tr>
<td>15.</td>
<td>25.</td>
<td>35. 45.</td>
</tr>
<tr>
<td>16.</td>
<td>26.</td>
<td>36. 46.</td>
</tr>
<tr>
<td>17.</td>
<td>27.</td>
<td>37. 47. BASCHART</td>
</tr>
<tr>
<td>18.</td>
<td>28.</td>
<td>38. 48. BASCout</td>
</tr>
<tr>
<td>19.</td>
<td>29.</td>
<td>39. 49. BASchart</td>
</tr>
</tbody>
</table>

INPUT CHARACTER STRING: BAS

The user is required to input a character string that approximates the program name. Using this character string, a search of the existing catalog index will be made and all the programs contained on the disk with that character string will be displayed as indicated below:

<table>
<thead>
<tr>
<th>DISK</th>
<th>O11</th>
<th>SEARCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>47.</td>
<td>BASCHART</td>
<td></td>
</tr>
<tr>
<td>48.</td>
<td>BASCout</td>
<td></td>
</tr>
<tr>
<td>49.</td>
<td>BASchart</td>
<td></td>
</tr>
</tbody>
</table>

Three files were searched out with string BAS.

FN '4 = END PROGRAM / FN '5 = SEARCH / FN '6 = BASCHART PROGRAM

510 TOTAL PROGRAMS PRESS RTN/FN?
Technical Note 3-82

The user should now be able to identify the specific program to be flow charted.

Once the user has identified the desired program (either by entering program name or by using the search function, FN '5) and located the program number, special function key '6, FN '6 = BASCHART PROGRAM, is used to initiate the subroutine by which the number of the program (in this instance 47) to be flow charted is entered.

Pressing FN '6 will cause one of the following screens to be displayed. If the user has previously employed the search function (FN '5), the following will appear:

```
***** DISK D11 SEARCH *****

47. BASCHART
48. BASICout
49. BASchart

Three files were searched out with string BAS.
WHAT IS THE NUMBER OF THE PROGRAM? _____
```

If the user knew the program name and consequently pressed FN '6, the following will be displayed:

```
***** DISK D11 CATALOG OF PROGRAMS *****

10. 20. 30. 40. 47. BASCHART
11. 21. 31. 41. 48. BASICout
12. 22. 32. 42. 49. BASchart
13. 23. 33. 43. 47. BASCHART
14. 24. 34. 44. 48. BASICout
15. 25. 35. 45. 49. BASchart
16. 26. 36. 46. 47. BASCHART
17. 27. 37. 48. BASICout
18. 28. 38. 49. BASchart
19. 29. 39. 47. BASCHART

WHAT IS THE NUMBER OF THE PROGRAM? _____
```
Once the user enters a valid program number by either alternative, he has identified the specific program to be flow charted utilizing the BASCHART program.

After entry of a valid program number, a sequence of questions is asked pertaining to printer address, program line number (starting and stopping for the flow chart), and program subroutine structure. The first screen requests a valid printer address.

**Printer Addresses Available: 204, 215, 216, 211, 005**

**Select Printer Address:**

Entering the desired printer address will cause the following screen to appear:

**Printer Addresses Available: 204, 215, 216, 211, 005**

**Select Printer Address:** 215

**Do you want a hard copy printout of your entire program? (Y/N)**

If a Y is entered, the entire program will be flow charted and printed. If an N is entered, which indicates a desire to print only a portion of the flow charted program, the following display will appear:

At what line No. do you want a hard copy print to start?
If first line No. of program, type in '0000'⇒____

At what line No. do you want the hard copy print to end?
If last line No. of program, type in '9999'⇒____

(Processing is done from the start regardless of where your line No. begins)
The user must now enter the starting and stopping line numbers of the portion of the program to be flow charted.

NOTE: If printer address 005 (CRT) is selected, questions pertaining to starting and ending line numbers will not appear. The entire flow charted program will appear on the screen in a continuous rollup display.

Once the program line numbers are entered, the following display, requesting the inclusion of remarks and image statements, appears:

```
DO YOU WISH TO INCLUDE THE REMARKS AND IMAGE STATEMENTS IN THE FLOW CHART? 
(Y/N)
```

A Y response will include all program remarks and image statements in the flow chart. An N response will exclude all statements of that type and results in a flow chart containing only program logic.

Regardless of which entry is made, the following screen will appear:

```
If your program has marked subroutines, are all of them after the main program?

Y = Will cause the oversized heading, 'SUBROUTINES', to be printed upon encountering the first marked subroutine. Everything after this is assured by the program to be subroutine connected.

N = Will treat all subroutines as if they are found within the main program. No oversized 'SUBROUTINES' heading is printed.

SELECT OPTION (Y/N) __
```
Technical Note 3-82

A Y response will cause the BASCHART program to print the heading "SUBROUTINES" when it encounters the first subroutine in the program being flow charted. This signifies that the main program is complete and everything that follows the SUBROUTINE heading is, in fact, a subroutine. If the user inserts an N, no heading will be printed and the BASCHART program will treat the subroutines as if they are contained in the main program.

Finally, a list of all the user's responses is presented providing the ability to re-input the data if erroneous results were entered.

---

After the user enters the desired response, BASCHART will proceed to flow chart the requested program. If a portion of a program is to be flow charted on a printer, and that portion does not begin at zero, the screen will display the flow chart of the program beginning at the first line number. The reason processing is done from the first Line Number, regardless of where the user would like the flow chart to begin, is simply to know if printing is started in a loop or branching sequence. When the starting Line Number to be printed is encountered, the screen displays the status, while flow charting is done at the printer.
During the flow charting process, the BASCHART program will keep the user informed as to status by displaying the following screen:

```
<<<<<<<<<<< NOW READING >>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

```
0190 IF 07$="605" THEN PRINT HEX(03)
```

<table>
<thead>
<tr>
<th>now at sector # 36086</th>
<th>numbered statements in sector: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ending sector # 36219</td>
<td>Percentage of processing completed: 2%</td>
</tr>
</tbody>
</table>

```
NOW FORMING A FLOW CHART OF BASCHART ON PRINTER 215
```

```
REMARKS ARE BEING INCLUDED
```

OPERATOR NOTE: PROCESSING TIME VARIES WITH EACH PROGRAM, THE AVERAGE IS 10 MINUTES FOR EVERY 30 SECTORS.

The status screen displays the line that is currently being flow charted, the ending sector, the sector being flow charted, and the percentage of processing completed. Once finished, the program returns to query the user as to any more flow chart requirements at this time.

**BASCHART PROGRAM FLOW CHART**

This subsection contains an example of the automated flow charting output produced by the BASCHART program. Program comments are supplied by BASCHART during the processing and at the end of an entire printout. The comments can range from remarks about the programming technique to remarks about a command or commands that could not be translated properly from hexadecimal or exceeded the parameters of the BASCHART program.
Technical Note 3-82

Flow chart of program BASCHART

(- --------)
(-        START     )
(- --------)

REM % 'BASCHART' makes a flow chart of any given program
REM ! (Charles C. Johnson) TAEG - U.S. NAVY Software
REM N.T.C Orlando, Fl.
REM % dimensioned for sector reading

..................0120
  : PROCESS:
  :------------:
  : 0120 : COM Zl
  :------------:
  : 0120 : DIM Z$(256),Q3$(256),Q2$(256),Q1$(256),Q0$(256),X$(256),T$8

REM % dimensioned for the statement conversion

..................0120
  : PROCESS:
  :------------:
  : 0120 : DIM Q7$(123),Q6$124
  :------------:
  : 0120 : DIM W$(256),L$(256),A$124,A$124

REM % dimensioned for sorting + statement assignment
REM
/*0190
IF *N2=0000
THEN
  [YES]
  PRINT( AT(6,33);"Thinking..."

[NO]
V
...
*/0190

/*0190
IF *07$="005"
THEN
  [YES]
  PRINT( HEX(03)

[NO]
V
...
*/0190

/*0190
IF *07$="215"
THEN $ OPEN /215

V
...
/*0190
/ *
/ * IF *07$="216"
/ THEN $ OPEN /216
*
*/

/*0190
/ *
/ * IF *07$="204"
/ THEN $ OPEN /204
*
*/

/*0190
/ *
/ * IF *07$="211"
/ THEN $ OPEN /211
*/
Technical Note 3-82

**PROCESS**: 0200

S9$=HEX(20)

*/0200
*/
*/ IF *D7$="005" OR N2[]0000
*/ THEN 08$="005"
*/
*/
*/
*/
*/
*/
*/
*

*********** 0200
** ELSE ** =========
***********

**PROCESS**: 0200

08$=07$

*/

---

**PROCESS**: 0200

SELECT PRINT [08$] (130)
APPENDIX A

SYSTEM DIAGRAM OF BASCHART PROGRAM
AND
BASCHART SUBROUTINE LISTING
Figure A-1. Interrelationship of BASCHART Subroutines
Technical Note 3-82

BASCHART SUBROUTINE LISTING

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Line</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>'4</td>
<td>3680</td>
<td>Print direction lines branching horizontally in and out of the flow of logic</td>
</tr>
<tr>
<td>'8</td>
<td>3650</td>
<td>Print (down) direction lines with arrows, prior to printing a new sequence in the flow chart</td>
</tr>
<tr>
<td>* '10</td>
<td>1670</td>
<td>Search new command string for (ON..GOSUB, ON..GOTO, FOR..NEXT, REM, GOTO, GOSUB), AND send to appropriate subroutine</td>
</tr>
<tr>
<td>* '15</td>
<td>1480</td>
<td>Search command string for (REM, %), separate the line No. from the command, in modular phase check for (FOR..NEXT)</td>
</tr>
<tr>
<td>'19</td>
<td>3800</td>
<td>Modular print routine or comment on (IF..THEN) or (FOR..NEXT) statement</td>
</tr>
<tr>
<td>* '20</td>
<td>1830</td>
<td>Search new command string for (IF..THEN, PRINT, INPUT, STOP, END, LOAD, RETURN) AND send to appropriate subroutine</td>
</tr>
<tr>
<td>'30</td>
<td>2420</td>
<td>Print (FOR symbol) in a (FOR..NEXT) Loop, branch out for start of loop</td>
</tr>
<tr>
<td>'35</td>
<td>2520</td>
<td>Print (NEXT symbol) in a (FOR..NEXT) Loop, branch in for end of loop</td>
</tr>
<tr>
<td>'40</td>
<td>2620</td>
<td>Print routine for (ON..GOTO, ON..GOSUB)</td>
</tr>
<tr>
<td>'45</td>
<td>2850</td>
<td>Print (GOTO symbol)</td>
</tr>
<tr>
<td>'50</td>
<td>2920</td>
<td>Print (STOP, END, LOAD symbols)</td>
</tr>
<tr>
<td>'60</td>
<td>2120</td>
<td>Print routine for (IF..THEN) branching sequence and symbol</td>
</tr>
<tr>
<td>'65</td>
<td>2980</td>
<td>Print routine for ELSE, ERROR) Loop branching sequence, begin Loop</td>
</tr>
<tr>
<td>'66</td>
<td>3960</td>
<td>Print routine for (ELSE, ERROR) Loop branching sequence, terminate Loop</td>
</tr>
</tbody>
</table>

*One of three main directional subroutines that decide, based on the present command string, what print subroutine to call up and process.*
Technical Note 3-82

BASCHART SUBROUTINE LISTING (continued)

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Line</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>'75</td>
<td>3040</td>
<td>Print REMARKS from program being flow charted</td>
</tr>
<tr>
<td>'80</td>
<td>3550</td>
<td>Print routine for (data process) symbol, unidentified commands default here</td>
</tr>
<tr>
<td>'81</td>
<td>3080</td>
<td>Print routine to end (data process) symbol</td>
</tr>
<tr>
<td>'82</td>
<td>3860</td>
<td>Print routine for (PRINT symbol)</td>
</tr>
<tr>
<td>'82</td>
<td>3940</td>
<td>Print routine to end (PRINT symbol)</td>
</tr>
<tr>
<td>'85</td>
<td>3120</td>
<td>Print (RETURN symbol) and/or comment on, no (STOP) or (END) encountered prior to this subroutine</td>
</tr>
<tr>
<td>'90</td>
<td>3180</td>
<td>Print flow chart (down) direction lines or straight or in branched, nested Loops</td>
</tr>
<tr>
<td>'91</td>
<td>3230</td>
<td>Print comment on Instruction number out of order</td>
</tr>
<tr>
<td>'92</td>
<td>3280</td>
<td>Print comment on Instruction number exceeding the program parameters</td>
</tr>
<tr>
<td>'93</td>
<td>3320</td>
<td>Print routine for Loop flow direction arrow ('↓' or '↑') up or down</td>
</tr>
<tr>
<td>'95</td>
<td>3370</td>
<td>Search the command string for the Operand, or a second main command statement</td>
</tr>
<tr>
<td>'96</td>
<td>3440</td>
<td>Print comment on illegal BASIC-2 Language statement encountered</td>
</tr>
<tr>
<td>'98</td>
<td>2020</td>
<td>Loop nest counter for the (FOR..NEXT) sequence, check count is within parameters</td>
</tr>
<tr>
<td>'99</td>
<td>3500</td>
<td>Print comment on unmatched Loop complement encountered</td>
</tr>
<tr>
<td>1100</td>
<td>1110</td>
<td>Search New command string for the First and Last statement of the program and the First and Last statement of a marked subroutine, print summation comments at end of program run</td>
</tr>
</tbody>
</table>
APPENDIX B

BASCHART ALPHANUMERIC VARIABLES
BASCHART ALPHANUMERIC VARIABLES

A$ holds the command string, after isolating from the disk sector and translating from hexcode to normal text

A1$ holds the current command string line number

A2$ holds the first word located in the current command string

B2$ holds the command string, other than the first isolated word of the string

C$ holds the operator input 'Y' or 'N' on the question of marked subroutines following the main program body

C2$ holds the operand section of the current processed command string

D2$ holds the second command encountered in a large string command (double command string) (i.e., ELSE, ERROR, IF..THEN)

D7$ holds the printer address

D8$ holds the temporary printer address

D9$ holds the disk address

E$ holds the referenced line numbers in (ON..GOTO), (ON..GOSUB) statements

F$ holds temporary first word of the current command string

F2$ holds the variable and starting value in a (FOR..NEXT) statement, to use later in identifying the end of loop

K2$ holds the (FOR..) or (..NEXT) command in the modular phase of the flow chart being created

L3$ holds the (FOR..) or (..NEXT) command during a check routine to match each with its correct complement

N$ holds the present statement line number

Q0$ holds the locations of the starting line numbered statements, for each sector of the program being processed

Q1$ holds the location of the end of sector data marker (FD) or (FE) for each sector of the program being processed

Q6$ holds the first byte of each command string and, if needed, the translated ASCII of the hexcoded command byte

S9$ holds a blank field, used in translating statements during the flow charting sequence of the program
Technical Note 3-82

BASCHART ALPHANUMERIC VARIABLES (continued)

$TS$ holds the name of the program being flow charted

$T1S$ holds the temporary value of '$S9$', used as a flag when encountered to indicate some operation was completed

$T2S$ holds the entire (intact) command string, while elsewhere it is being broken down and processed during the flow chart sequence

$U1S$ holds the location in a command string, while being read from the sector, of the actual end of the statement, not counting the ':' (colon), which can be a statement separator

$Z1S$ holds the operator input 'Y' or 'N' on the question of including remarks and image statements in the flow chart output
BASCHART ALPHANUMERIC ARRAY VARIABLES

- \( L_S( ) \) holds the sorted sector location of each beginning string command
- \( L_S1( ) \) holds the stored, nested (FOR..NEXT) statement, to match against each equal complement
- \( L_S2( ) \) holds the stored line number of each nested, starting (FOR..NEXT) statement
- \( Q_S2( ) \) holds locations of each start of numbered statements in each sector, read by the program
- \( Q_S3( ) \) holds locations of each start of nonnumbered statements in each sector, read by the program
- \( Q_S4( ) \) holds the ASCII equivalent of a hexcoded command
- \( Q_S5( ) \) holds the hexcoded commands used in the BASIC-2 Language
- \( Q_S7( ) \) holds the sector position of each ":" mark, and later each "\%", in the sector currently being read
- \( W_S( ) \) holds the work array for the MAT SORT command
- \( X_S( ) \) holds the entire sector being read by the program
- \( Z_S( ) \) holds the temporary location of "\%" statements in the process of isolating string commands, later it holds the sorted location, in the sector being read, of each string command prior to its being translated from hexcode into ASCII
BASchart - start program for

the BASCHART flowchart: REM : [ Charles C. Johnson ] TAEG-U.S. NAVY S
oftware : REM N.T.C. Orlando, Florida

C$(450)8 - File Names from the Disk Directory.
Cl$2 - File Status in hex.

50 REM %

dimensioning

DIM 0$3,T$8,07$3,Q4$(123)10,Q5$(123)10,Z1$1,C$1,N2,N3
DIM N2$4,Z1$(123)8,04$(123)10,Z2$1,Z1$3,Z4$3,Z5$3,A$3
DIM N$8,S$2,52$2,Z$3,Z1$1

I0 DIM B$(16)16,C$(1100)8,C1$2,C2$(1100)8: REM %------------------------

$STAT="BASCHART"

80 C$()=ALL(FF)
90 IF A=1 THEN 230
   IF Z1=1 THEN 210
100 PRINT HEX(06)
   FOR G=1 TO 3
     E=11
     F=36
     H=0
110 H=H+1
   PRINT HEX(03H)
   :PRINT USING 00
   :PRINT USING 03
   :PRINT USING 10
   IF E[]0 AND F[]0 THEN 110
   NEXT G
   PRINT HEX(03)

120 % BBBBBB A SSSSSS CCCCCC H H H A RRRR T
TTTTTT
130 % B B A A S C H H A A R R R
140 % BBBBBB A A SSSSSS C HHHHHH A A RRRR
150 % B B AAAAAAA S C H H AAAAAAA R R
160 % B B A A S C H H A A R R
170 % BBBBBB A A SSSSSS CCCCCC H H A A R R

180 PRINT AT(7,0);
   :PRINT USING 120
   :PRINT USING 130
   :PRINT USING 140
   :PRINT USING 150
   :PRINT USING 160
PROGRAM LISTING

:PRINTUSING 170
190 PRINT AT(17,10);"This program is designed to create a logic flowchart of"
 :PRINT AT(18,12);"any active program on disk in the Basic-2 language."

200 Z1$=HEX(0D)
 :PRINT AT(22,7);"Is it your intention to run the program 'BASCHART'?" 
 :PRINT AT(22,10);"(Y/N)";
 :KEYIN Z1$
 :IF Z1$="N" OR Z1$="n" THEN GOSUB '4
 :IF Z1$[1]"Y" AND Z1$[1]"y" THEN 200
 :PRINT HEX(03)

210 A=1
220 REM %

======== SELECT DISK

GOSUB '205

230 DATA LOAD BA T#1,(0)8$(())
 :ERRORPRINT HEX(07)
 :GOTO 220

240 % ****************************

250 % *  Now Loading All the Active Programs on ### *

260 % ****************************

270 PRINT HEX(0306);AT(8,0); 
 :PRINTUSING 240
 :PRINTUSING 250,D9$(())
 :PRINTUSING 260

280 S2$=STR(B$(),1,2)
 :S=VAL(STR(S2$,2,1))
290 S$=ALL(FF)

: L=0
 : L=0
300 PRINT HEX(06);AT(18,25);"LOADING CATALOG";
310 REM %

======== LOAD FILE NAMES INT

0 C$(())

320 S$=ADOC BIN(1)
330 IF S$=S2$ THEN 420
340 DATA LOAD BA T#1,(S$)8$(())
350 PRINT AT(18,45,5);AT(18,45); S;

360 FOR J=1 TO 16

28 31
PROGRAM LISTING

370 IF STR(B$(J),1,2)[STR(C1$,1,2) THEN 390
380 L=L+1
:CS(L)=STR(B$(J),9,8)
390 NEXT J
400 GOTO 320

410 REM %

======== SORT FILE NAMES

420 PRINT HEX(06);AT(18,25);"Sorting the Catalog"
:IF L=0 THEN 500
:MAT SORT CS() TO W$04-$0
430 MAT MOVE CS(),L$(1) TO C2$(1)
:PRINT HEX(0306)
440 FOR L2=1 TO L
445 PRINT HEX(06);AT(43(L2,10)+3,L1) ;L2;TAB(L1+4);C2$(L2)
450 NEXT L2
500 GOSUB 810
:GOTO 440

540 REM %

======== OEFFN'5 / SEARCH R OUTINE

550 OEFFN'5
560 PRINT AT(22,0,79)
570 PRINT AT(22,20);"Input character string:";
:INPUT-N$
580 IF N$[]" THEN 600
590 PRINT AT(22,0,79);AT(22,0);
:RETURN

600 MAT SEARCH C2$(),N$ TO L$(1)
610 IF L$(1);HEX(0000) THEN 640
620 PRINT AT(22,52);"NOT ON FILE"
630 GOTO 570

640 J,L1=1
:J1=2
:Y1=0
:FOR K=0 TO 21
:PRINT AT(K,0,80)
PROGRAM LISTING

: NEXT K
: K = 1
650 IF L$(j) = HEX(0000) THEN 780
655 PRINT HEX(06); AT(0, 20); "***** Disk "; 09$; " Search of Programs *****"
660 X = VAL(L$(j), 2)
670 Y = INT((X - 1) / 8) + 1
680 IF Y = Y THEN 715
690 I = MOD(J, 10) + 1
700 IF I[L] THEN 720
710 I = I + 15
: J = 1
: I = 2
: IF I[L] THEN 70 THEN I = 0
: GOTO 720

715 IF MOD(J, 45)[J] THEN 760
: PRINT AT(18, 12 - LEN(N$)); K - 1; " Files were searched out having "; N$; " (more on next screen)"
: GOSUB 810
: I = 2
: J, I = 1
: GOTO 760

720 IF MOD(J, 45) = [J] THEN 730
: PRINT AT(18, 12 - LEN(N$)); K - 1; " Files were searched out having "; N$; " (more on next screen)"
: GOSUB 810
: I = 2
: J, I = 1
730 PRINT AT(I + 3, I + 4); BOX(1, 9); AT(I + 3, 11); Y; AT(I + 3, I + 5); C2$(Y)
: K = K + 1
740 Y = Y
750 J = J + 1
760 I = I + 1
770 IF J[L] THEN 650
780 PRINT AT(18, 20 - LEN(N$)); K - 1; " Files were searched out with string "; N$
: GOSUB 810
790 N$ = ALL(20)
: GOTO 570

800 REM %

============== LOAD ROUTINE

810 PRINT HEX(06); AT(20, 0, 79); AT(20, 2); "[ FN'4 = End / FN'5 = Search / FN'6 = Program you want to run on BASCHART ]"
: PRINT HEX(06); AT(22, 0, 79); AT(22, 17); L; " Total Programs "; HEX(06);
: INPUT " Press [RTN/FN] "; A$
820 IF A$ = HEX(0020) OR A$ = " " OR A$ = "[R]" THEN 830
PROGRAM LISTING

: A$ = "[R]"
: GOTO 810

830 PRINT HEX(03)
: RETURN

840 REM %

========== DEFFN'205 / SELECT

OISK SUBROUTINE ===========

850 DEFFN'205
: PRINT HEX(03); AT(12,11); "Disk Addresses available: 325 B25 365 010 011 012 013 B20"; AT(13,38); "014 015 030 031 032 033 034 035"; AT(12,8); BOX(2,61); AT(9,9); "*** Disk address of the program you want to Flo wchart ****"

860 PRINT AT(18,24); "Select Disk Address: ";
: D9$ = " 
: INPUT -D9$
: PRINT HEX(06)
: IF D9$ = " " THEN GOSUB '4
: IF POS("3BD", TR(D9$)) * POS("123456", TR(D9$, 1)) * POS("012345", STR(D9$, 3, 1)) = 0 THEN 860
: SELECT #1[D9$]
: RETURN

870 REM %

========== DEFFN' UR

N START ===========

880 DEFFN'4
: $PSTAT = " "
: COM CLEAR
: RETURN CLEAR ALL

: LOAD T "START"

: ERROR PRINT HEX(03); AT(8,19); "NOT ABLE TO LOAD START"
: END

890 REM %

========== DEFFN'6 / LOAD BASCH

ART ===========

900 DEFFN'6
910 PRINT AT(22,0,79)
920 Z=0
: PRINT AT(22,10);

: PRINT AT(22,16); "WHAT IS THE NUMBER OF THE PROGRAM: ";
Technical Note 3-82

10/25/82 BASHchart PAGE 7

PROGRAM LISTING

:LINPUSZ$
:IF Z$=" " THEN RETURN
:CONVERT Z$ TO Z
:ERRORGOTO 920
930 Z$=HEX(00)
:IF Z$="0" OR Z$="1" THEN 920
:PRINT AT(22,14);""
:PRINT AT(22,16);" IS ";CZ$(Z);" THE PROGRAM YOU WANT? (Y/N) ";
:KEYIN Z$
940 IF Z$="N" OR Z$="n" THEN 920
:IF Z$="y" AND Z$="y" THEN 930
950 DIM DB$(5),C8$2
:PRINT HEX(D03);AT(9,13);"Printer Address available: 204 215 216 21
005"
:PRINT AT(12,21);"Select Printer Address: ";
:DB$=" "
:INPUT-DB$
960 STR(0$(),1)="20421521621005"
:MAT SEARCH DB$((),=0$ TO C8$ STEP 3
:IF C8$=HEX(0000) THEN 950
970 DATA "AND", "RUN", "CLEAR", "RENUMBER", "TAPE", "OR", "DISK", "TEMP", "XOR", "KEYIN", "COPY"
980 DATA "DAC", "DSC", "SUB", "ROTATE", "BACK", "UNPACK", "BOOL", "ADD", "INIT", "ERROR", "ERR"
:DATA "VERIFY", "DBACKSPACE", "BEG", "OFF", "C1", "C0", "D", "R", "OPEN", "LINPUT", "ELSE"
990 DATA "ROUND", "SPACE", "PRINTING", "CONVERT", "MOVE", "PLOT", "PRINT", "REM", "COM", "RESTORE"
:DATA "SELECT", "LOAD", "MAT", "PLOT", "REWIND", "BACKSPACE", "SKIP", "SCRAT CH", "DA", "DC";
LEN"
1000 DATA "VAL", "NUM", "POS", ",", "ATN", "SIN", "COS", "ARC", "TAB", "DFN"
:DATA "RND", "TAN";
:DATA "TAN", "FIX", "HEX", ",", "LGT", "HEX", "AT", "HEXOF", "MAX", "MIN"
:MOD", "VER"
1010 REM
1020 DATA "AA", "BB", "CC", "DD", "EE", "FF", "GG", "HH", "II", "JJ"
1030 DATA "AA", "BB", "CC", "DD", "EE", "FF", "GG", "HH", "II", "JJ"
1040 DATA "AA", "BB", "CC", "DD", "EE", "FF", "GG", "HH", "II", "JJ"
1050 DATA "AA", "BB", "CC", "DD", "EE", "FF", "GG", "HH", "II", "JJ"
1060 DATA "AA", "BB", "CC", "DD", "EE", "FF", "GG", "HH", "II", "JJ"
10/25/82 BASchart PAGE 8

PROGRAM LISTING

:DATA "05","DC","00","0F","07","04","03","01","08","CD","CE"
1070 DATA "0D","CC","01","C3","C4","C5","C6","C7","C8","C2","C9"
:DATA "CA","CF","92","E5","EA","89","FC","F5","F6","F7","F8"
:DATA "F9","F1"
1080 REM
1090 RESTORE LINE 970
1100 FOR I=1 TO 123
1110 REM
1120 PRINT AT(15,6);"00 you want a hard copy print-out of your entire program ? (Y/N)");
1130 IF Z2$="N" OR Z2$="n" THEN 113D
1140 IF Z2$="Y" AND Z2$="y" THEN 1120
1150 GOTO 1170
1160 IF N3>N2 THEN 1170
1170 Z1$=HEX(00)
1180 REM
1190 RESTORE LINE 1020
1200 FOR I=1 TO 123
1210 REM
1220 PRINT AT(15,6);"00 you want a hard copy print-out of your entire program ? (Y/N)");
1230 IF Z2$="N" OR Z2$="n" THEN 113D
1240 IF Z2$="Y" AND Z2$="y" THEN 1120
1250 GOTO 1170
1260 IF N3>N2 THEN 1170
1270 PRINT AT(15,6);"Sorry, your line No.'s are out of order.");
1280 GOTO 1140
1290 PRINT AT(14,10);BOX(2,58);AT(14,11);"At what line No. do you want the hard copy print to end? "
1300 IF N3>N2 THEN 1170
1310 PRINT AT(5,20);"Sorry, your line No.'s are out of order.");
1320 GOTO 1140
1330 PRINT AT(5,0);BOX(2,78)
1340 PRINT AT(5,0);BOX(2,78)
:PRINT AT(5,2);"Do you wish to include the remarks and image statements in the flow-chart?"
:PRINT AT(6,36);"(Y/N)";
:KEYIN Z1$
:IF Z1$="y" THEN Z1$="Y"
:IF Z1$="n" THEN Z1$="N"
:IF Z1$="N" OR Z1$="Y" THEN 1180
:GOTO 1170

1180 C$=HEX(00)
:PRINT AT(10,0);80)010,79);AT(10,1);"If your program has marked subroutines, are all of them after the main program?"
1182 PRINT HEX(06);AT(13,5);"Y =] Will cause the oversized heading, 'SUBROUTINES', to be printed";AT(14,10);"upon encountering the first marked subroutine. Everything after";AT(15,10);"this is assumed by the program to be subroutine connected."
1184 PRINT HEX(06);AT(17,5);"N =] Will treat all subroutines as if they are found within the";AT(18,10);"main program. No oversized 'SUBROUTINES' heading is printed."
1185 PRINT AT(11,36);"(Y/N)");
:KEYIN C$
:IF C$="y" THEN C$="Y"
:IF C$="n" THEN C$="N"
:IF C$="N" OR C$="Y" THEN 1400
:GOTO 1170

1190 :: ... :: ... :: ... :: ... :: ... :: ... :: ... :: ... :: ... :: 
1200 :: ... :: ... :: 1210 :: ... :: ... :: ... :: 1220 :: 1230 :: 
1240 :: Your program name is .............................................
1250 :: 1260 :: Your program will be flow-charted on .................
1270 :: Your program will be flow-charted on the .................
SCREEN ::
1280 ::
1290 :: You want a print-out of the entire program .................
1300 :: 1310 :: Starting at line number .......
1320 % ::
::
1330 % :: Ending at line number ........ #
::
1340 % ::
::
1350 % :: You want remarks & image statements included ............
### ::
1360 % ::
::
1370 % :: Your marked subroutines follow the main program ........
### ::
1380 % ::
::
1390 % ::
::
1400 IF N2=0000 AND N3=9999 THEN Z3$="YES"
 :ELSE Z3$=" NO"
 :IF Z1$="Y" THEN Z4$="YES"
 :ELSE Z4$=" NO"
 :IF C$="Y" THEN Z5$="YES"
 :ELSE Z5$=" NO"
 :PRINT HEX(03)
1410 PRINT
 :PRINT USING 1190
 :PRINT USING 1200
 :PRINT USING 1210
 :PRINT USING 1220,C2$(Z)
 :PRINT USING 1230
 :PRINT USING 1240,D9$
 :PRINT USING 1250
 :IF D7$["005"] THEN PRINT USING 1260,07$
 :ELSE PRINT USING 1270
 :PRINT USING 1280
 :PRINT USING 1290,Z3$  
1420 PRINT USING 1300
 :PRINT USING 1310,N2
 :PRINT USING 1320
 :PRINT USING 1330,N3
 :PRINT USING 1340
 :PRINT USING 1350,Z4$
 :PRINT USING 1360
 :PRINT USING 1370,Z5$
 :PRINT USING 1380
 :PRINT USING 1390
 :PRINT AT(5,65);BOX(1,9)
1430 Z2$=" 
 :PRINT AT(23,6);"Press RETURN to continue, any other key to re-input
 data == ? ";
 :KEYIN Z2$
 :IF Z2$=HEX(00) THEN 1440
 :Z1=1

PROGRAM LISTING

:09$,D7$,N2$,N3$,Z$,Z1$,C$=HEX(20)
:A=0
:GOTO 80

1440 PRINT HEX(03);HEX(020402020E);AT(8,18);"Now Loading main BASCHART program"
:T$=C2$(Z)
:RETURN CLEAR ALL
:LOAD T"BASCHART"

9000 REM %

GENERAL SUBROUTINE

:PRINT HEX(030E);"YOU HAVE LOADED 'OEFNO' THE GENERAL SUBROUTINES";
HEX(0A)
:LIST
:END

9010 DEFFN 'O "SCRATCH";HEX(22);"BASchart";HEX(22);"SAVET ()";HEX(22);
"BASchart";HEX(2200)
9070 DEFFN '31
'BASCHART' makes a flowchart
of any given program.

110 REM : (Charles C. Johnson ) TAEG - U.S. NAVY Software: REM
N.T.C Orlando, Fl.

120 REM %
dimensioned for sector readin

g:
:COM 21
:DIM Z$(256),Q3$(256)2,Q2$(256)2,Q1$2,Q0$2,X$(256),T$8: REM % dimen-
ioned for the statement conversion
:DIM Q7$(123)2,Q6$124

130 REM %
dimensioned for sorting + sta-
tement assignment
:DIM Q2(256),W$(256)2,L$(256)2,A$124,A1$124: REM
140 % :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: ::
150 % :: ::
160 % :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: ::
170 SELECT PRINT 005(60)
:PRINT HEX(DF);HEX(03)

180 REM %
dimensioning for main Program

:DIM A2$124,B2$124,C2$124,D2$124,F$124,F2$124,K2$124,L3$124,S9$124,T1
$124,T2$124
:DIM E$10,N$5,U1$3
:DIM L1$(50)124,L2$(50)124
:DIM F0(30),V(30)
:G1=0
190 IF N2=0000 THEN PRINT AT(6,33);"Thinking..."

200 S9$=HEX(20)
:IF D7$="005" OR N2[]0000 THEN D8$="005"
:ELSE D8$=D7$
:SELECT PRINT [D8$] (130)
:PRINT HEX(00)
:PRINT "Program ";S$,

210 IF D8%[]"005" THEN PRINT ""
PROGRAM LISTING

/////////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////////

220 PRINT " "
PRINT " (---------)"
PRINT " ( START )"
PRINT " (---------)"
PRINT " "
PRINT " V"
PRINT " "
PRINT " "
PRINT " "
PRINT " "
SELECT #1 [09$]
240 LIMITS T#1,T$,0,Q0,Q1,Q1
:QB=0
250 IF ABS(Q1)=1 THEN 260
:SELECT PRINT 005(80)
:PRINT HEX(03);AT(12,8);"ERROR, ";T$;" is not an active program";AT(14,8);"now Loading BASCHART start program"
:GOTO 730

260 QB=QB+1
:DATA LOAD BA T#1,(0+QB) X$( )
:MAT SEARCH X$( ),=HEX(FE) TO Q1$
:IF QB+0=Q0 THEN 270
:IF VAL(Q1$, 2)=0 THEN 260
:Q0=Q0+QB
270 REM %
READ SECTORS ON DISK, ONE AT A TIME
280 Q=Q+1
:DATA LOAD BA T#1,(0) X$( )
290 Q2$( ),Q3$( ),Z$( ),W$( ),L$( ),Q7$( )=ALL(00)
:Q1$,Q0$,Q6$=HEX(00)
:MAT Q2=ZER
:ES,1=0
300 MAT RE DIM Q2(256),L3(256)2,W$(256)2
310 REM %
SEARCH FOR BEGINNING AND ENDING LINE STATEMENTS
320 IF STR(X$( ),2,2)=HEX(0000) THEN Q1=1
:IF Q1=1 THEN 1310
330 MAT SEARCH X$( ),=HEX(FD) TO Q1$
:IF VAL(Q1$, 2)=0 THEN MAT SEARCH X$( ),=HEX(FE) TO Q1$
340 MAT SEARCH X$( ),=HEX(20FF) TO Q0$
:MAT SEARCH X$( ),=HEX(00FF) TO Q2$
350 MAT SEARCH X$( ),=HEX(3A) TO Q3$( )
360 IF VAL(Q0$, 2)=2 THEN Q0$=ALL(00): REM
( delete colons in between ' " ' marks )

370 MAT SEARCH X$09 =HEX(22) TO Q7$( )
   :IF VAL(Q7$(1),2)=0 OR VAL(Q3$(1),2)=0 THEN 420
380 I3,I=1
390 IF Q3$(I)Q7$(I) AND Q3$(I)Q7$(I+1) THEN 400
   :IF Q3$(I)Q7$(I+1) THEN I=I+1
   :IF Q3$(I)Q7$(I+1) THEN I3=I3+2
   :IF VAL(Q7$(I+1),2)=0 OR VAL(Q3$(I),2)=0 THEN 420
   :GOTO 390

400 I2=I
410 Q3$(I2)=Q3$(I2+1)
   :IF VAL(Q3$(I2+1),2)=0 THEN 390
   :I2=I2+1
   :GOTO 410

420 I2,I,I3=0
   :Q7$,Z$( )=ALL(00)
   :IF U5=1 THEN 490
430 REM %

   delete colons in between % an
d next line No.
   :MAT RE QIM Z$(256)2
   :MAT SEARCH X$(),HEX(08) TO Z$( )
   :IF VAL(Z$(1),2)=0 THEN 490
   :U5=1
   :Q7$(1)=Z$(1)
   :I2=1
440 I2=I2+1
   :IF VAL(Z$(I2),2)=0 THEN 450
   :Q7$(I2+I2-1)=Z$(I2)
   :GOTO 440

450 Z$( )=ALL(00)
   :I2=0
460 I2=I2+2
   :IF I2=2 THEN I2=1
   :IF VAL(Q7$(I2),2)=0 THEN 380
470 MAT SEARCH STR(X$(),VAL(Q7$(I2),2))=HEX(00) TO U1$
   :I=VAL(U1$,2)+VAL(Q7$(I2),2)+3
480 Q7$(I2+1)=BIN(I,2)
   :GOTO 460

490 MAT RE QIM Z$(256)4
   :U5=0: REM % number of statements with line numbers
500 I=I+1
   :IF VAL(Q2$(1),2) THEN 500
   :IF VAL(Q0$,2) THEN I=I+1
   :E5=I-1
510 REM %
CONVERTING W1$, WQ$, WQ2$(, WQ3$(

TO ZW$(

520 I=1
: I2=I3, Q=0
530 Q=VAL(WQ$(2)
: CONVERT Q TO ZW$(I), (####)
: IF VAL(Q(2))=0 THEN 540
: G1=I
: I=2
: Q=VAL(WQ$,2)+1
: CONVERT Q TO ZW$(I), (####)
: Q3(I)=VAL(WQ$(2)+1: REM % array Q2() will hold the numbered line statements
540 I2=I2+1
: IF VAL(WQ3$(I2),2)=0 THEN 570
: I=I+1
550 IF G1=0 THEN Q2(I2)=VAL(WQ3$(I2),2)+1
: IF G1=1 THEN Q2(I2+1)=VAL(WQ3$(I2),2)+1
: Q=VAL(WQ3$(I2),2)+1
: CONVERT Q TO ZW$(I), (####)
560 GOTO 540

570 I3=I3+1
: IF VAL(WQ3$(I3),2)=0 THEN 620
580 I=I+1
590 Q=VAL(WQ3$(I3),2)
: CONVERT Q TO ZW$(I), (####)
600 GOTO 570

610 REM%

RE-DIMENSIONING AND SORTING

620 MAT RE DIM Q2(E5), ZW$(I4, L9$(I2), WS$(I2)
630 MAT SORT ZW$() TO WS$((), L9$()
640 H, ML, H3, H4=0
650 REM%

LOCATING THE STRING A$ BETWEEN

IN EACH ZW$( ) ARRAY

660 H=H+1
: CONVERT ZW$(VAL(L9$(H),2)) TO H3
: IF H3=VAL(WQ$(2)) THEN 280
: ML=H
: IF ML≤E5 THEN ML=E5
: CONVERT ZW$(VAL(L9$(H+1),2)) TO H4
: I=0
670 I=I+1
: IF Q2(I)=H3 THEN A$=STR(X$((), H3+3, (H4-H3)-3)
: IF Q2(I)=H3 THEN 690
LOCATING THE LINE NUMBER THAT GOES WITH EACH STATEMENT A$.

690 IF H1()E5 THEN 710
700 IF H3()Q2(H1) THEN A1$=STR(X$(),Q2(H1)+1,2)
    :IF H3()Q2(H1) THEN 760
    :ML=ML-1
    :IF ML=0 THEN 720
    :GOTO 710

710 IF H3()Q2(H1) AND H3()Q2(H1+1) THEN A1$=STR(X$(),Q2(H1)+1,2)
    :IF H3()Q2(H1) AND H3()Q2(H1+1) THEN 760
    :ML=ML-1
    :IF ML=0 THEN 720
    :GOTO 710

720 SELECT PRINT 005(80)
    :PRINT HEX(03);AT(12,12);"ERROR, (P56) at line No. 640, 'this can be caused by a protected program."
730 IF D7$="216" THEN $CLOSE/216
    :IF D7$="215" THEN $CLOSE/215
    :IF D7$="204" THEN $CLOSE/204
    :IF D7$="211" THEN $CLOSE/211
740 FOR I=1 TO 12000
    :NEXT I
    :RETURN CLEAR ALL
    :LOAD T "BASchart"

750 REM %
 GO TO THE STRING CONVERSION T WHEN THE MAIN BODY
760 UNPACK(####) A1$ TO AI
    :Q,12,13=0
    :A1$=HEX(00)
    :Q2$(),Q3$()=ALL(00)
770 REM %
 shift the statement to the left, if needed
    :A1$=A$
    :I2=LEN(A1$)
780 Q=Q+1
    :IF Q=125 THEN 790
    :IF STR(A1$,Q,1)=HEX(20) THEN 780
    :IF STR(A1$,1,1)[HEX(20) THEN 800
PROGRAM LISTING

:A$=STR$(A1$,Q,I2-Q)
:GOTO 800

790 A$="% COMMENT, THIS LINE CANNOT BE PROCESSED"
800 A1$=HEX(00)
 :I2,Q=0
810 REM %

identify the first byte (comm
and) in the statement

820 I=0
830 I=I+1
 :HEXUNPACK STR$(A$,1) TO Q6$
 :IF Q6[1]Q5$(I) THEN 850
 :A$=A$
 :I2=LEN(A1$)
 :I3=LEN(Q4$(I))
 :IF I2+I3)I3+1) THEN I2=I2-(I3+1)
840 IF I83 THEN A$=Q4$(I)&" "&STR$(A1$,2,I2+I3)
 :ELSE A$=Q4$(I)&STR$(A1$,2,I2+I3)
 :Q6$=HEX(00)
 :GOTO 860

850 Q6$=HEX(00)
 :I2=I23 THEN 830
860 A1$=HEX(00)
 :I,I2,I3=0
870 REM %

identify any referenced line
number in the statement

880 XL=0
 :Q=POS(A$=HEX(FF))
 :IF Q[12 THEN 890
 :XL=1
 :A$=A$
 :A$=STR$(A1$,1,Q-1)
 :A1$=HEX(00)
 :Q=0
890 IF Q=0 THEN 940
 :A$=A$
 :I2=LEN(A1$)
 :IF I2-Q1 THEN 900
 :I=Q
 :GOTO 910

900 I=POS(STR$(A$,Q+1,I2-Q)=HEX(20FF))+Q
910 HEXUNPACK STR$(A$,Q+1,2) TO Q6$
 :IF I=Q THEN A$=STR$(A1$,1,Q-1)&Q6$&STR$(A1$,I,I2-(I-1))
920 IF I[]Q THEN A$=STR$(A1$,1,Q-1)&Q6$&STR$(A1$,I,I2-(I-1))

42
PROGRAM LISTING

:Q6$, AI$=HEX(00)
:I,I2,Q=0
:GOTO 880

930 REM %

* identify all HEX coded bytes in the statement

940 Q=0
:Q6$=HEX(00)
:I=I+1
:IF I]I23 THEN 980
:HEXPACK Q6$ FROM Q5$(I)
950 Q=POS(A$=Q6$)
:IF Q=0 THEN 940
:A1$=A$
:I2=LEN(A1$)-Q
960 IF I[83 THEN 970
:IF I2[1 THEN A3=STR(A1$,1,Q-1)&Q4$(I)
:ELSE A3=STR(A1$,1,Q-1)&STR(A1$,Q+1,I2)
:Q,I2=0
:A1$=HEX(00)
:GOTO 940
970 IF I2[1 THEN A3=STR(A1$,1,Q-1)&Q4$(I)
:ELSE A3=STR(A1$,1,Q-1)&""&STR(A1$,Q+1,I2)
:Q,I2=0
:A1$=HEX(00)
:GOTO 940
980 I=0
:Q=POS(A3=HEX(8A))+POS(A$=HEX(8B))+POS(A$=HEX(D3))+POS(A$=HEX(D2))+POS(A$=HEX(F5))+POS(A$=HEX(DC))
:IF Q[0 THEN 940
:CONVERT A1 TO A1$(####)
990 IF A1]=N2 AND A1]=N3 THEN D8$=07$
:ELSE D8$="005"
:IF A1]=N3 THEN 1310
930 Q6$=A$
:AI$=AI$&""&Q6$
:A1$,Q6$=HEX(00)
:IF D8$="005" THEN 1100
:SELECT PRINT 005(80)
1000 IF T=1 THEN 1020
:PRINT HEX(03)
:T=1
:IF A1]=N2 THEN 1020
:SELECT PRINT [D8$] (130)
:PRINT HEX(UC)
:PRINT "******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
1010 PRINT "PARTIAL FLOW-CHART OF ";T$ FROM LI NES ";N2;" TO ";N3
1020 SELECT PRINT 005(80)
   :PRINT HEX(06);AT(6,0);STR(A11,1,80)
1030 PRINT AT(3,0);" aaaaaaaaaaaaa[NOW READING ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
1040 PRINT AT(5,0);80X(0,75);AT(7,0);80X(0,75)
   :IF LEN(A$)>79 THEN 1050
   :A9=(INT((80-LEN(A$))/2))-2
   :IF A9<0 THEN 1050
   :PRINT AT(6,A9);A$
   :GOTO 1060
1050 PRINT AT(6,0);STR(A$4,78)
1060 PRINT AT(8,1);"now at sector # ";O
   :PRINT AT(9,55);"Percentage of processing completed:";INT(100-((100*(Q0-0)/Q8)+1));"%"
1070 PRINT AT(16,18);BOX(1,29)
   :PRINT AT(8,0);BOX(2,23)
   :PRINT AT(8,34);BOX(2,41)
   :PRINT AT(9,1);"Ending sector # ";Q0
   :PRINT AT(8,40);"numbered statements in sector:";E5
   :IF L1$="Y" THEN PRINT AT(16,20);"Remarks are being included"
1080 PRINT AT(19,8);"operator note: processing time varies with each pr   ogram."
   :PRINT AT(20,25);"the average is 10 minutes for every 30 sectors."
1090 PRINT AT(12,0);BOX(3,75);
   :PRINTUSING 140
   :PRINTUSING 150,T$;08$
   :PRINTUSING 160
   :PRINT AT(13,3);BOX(1,69)
1100 SELECT PRINT [08$] (130)
   :GOSUB 1110
   :GOTO 660
1110 RBM %

START OF THE MAIN PROGRAM ...

1120 P=O
   :J=O
   :GOSUB '96
   :IF P=1 THEN RETURN
   :J=POS(STR(A$;J)=HEX(20))
   :GOSUB '96
   :IF P=1 THEN RETURN
   :IF STR(A$;J,5)="DEFFN" THEN L1=O
   :IF STR(A$;J,5)="DEFFN" THEN 1160
1130 IF L2=1 THEN 1270
10/25/82   BASCHART   PAGE 10

PROGRAM LISTING

:GOSUB '66
:IF U=1 THEN GOSUB 3750
1140 GOSUB '15
   :IF E9=1 THEN 1140
   :IF U6=1 THEN GOSUB '66
1150 IF A2$="STOP" OR A2$="END" THEN L3=1
       :RETURN

1160 U=0
   :IF C$="N" THEN M=1
   :IF M=1 THEN 1230
   :IF X0=1 THEN GOSUB '81
   :IF Y0=1 THEN GOSUB '83
   :GOSUB '66
   :PRINT
   :PRINT
   :PRINT
1170 PRINT "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
PROGRAM LISTING

1250 PRINT STR(A$,J)
1260 E=0
1270 GOSUB '66
1280 GOSUB '15
1290 IF A2$="END" THEN L2=1
1300 PRINT
1310 IF X0=1 THEN GOSUB '81
1320 IF N3[9999 THEN 1390
1330 IF J1[1 THEN 1340
1340 IF L3=1 THEN 1350
1350 IF W3=0 THEN 1370
1360 PRINT "NOTICE: THERE WERE";W3;" /COMMENTS/ GIVEN DURING THIS RUN"
1370 GOTO 1370

COMMENT: THERE IS AN /END/ COMMAND IN THIS SUBROUTINE.

COMMENT, THERE WAS NO /END/ OR /STOP/ STATEMENT IN THIS PROGRAM

PRINT "NOTICE: MULTI-STATEMENT LINE NUMBERS WERE USED IN THIS PROGRAM"

PRINT "NOTICE: THERE WAS 1 /COMMENT/ GIVEN DURING THIS RUN"
1370 PRINT
1380 IF G=0 THEN 1390
   PRINT
   PRINT "COMMENT: THIS PROGRAM CONTAINS 'GOTO' STATEMENTS"
   IF G=1 THEN PRINT "THERE WAS 1 /GOTO/ DETECTED DURING THIS RUN"
   IF G=1 THEN PRINT "THERE WERE";G,"/GOTO'S/ DETECTED DURING THIS RUN"
   PRINT
   PRINT
1390 IF D7$="215" THEN $CLOSE/215
   IF D7$="216" THEN $CLOSE/216
   IF D7$="204" THEN $CLOSE/204
   IF D7$="211" THEN $CLOSE/211
   SELECT PRINT 005(80)
1400 I,J=0
   PRINT HEX(07)
   FOR I=10 TO 1200
     NEXT I
   FOR I=1 TO 3
     PRINT HEX(07)
     FOR J=1 TO 800
       NEXT J,I
     FOR I=1 TO 400
       NEXT I
     PRINT HEX(07)
     FOR I=1 TO 2400
       NEXT I
     FOR I=1 TO 2
       PRINT HEX(07)
       FOR J=1 TO 1000
         NEXT J,I
     PRINT HEX(03);AT(20,7);BOX(1,56)
1410 PRINT HEX(03);AT(20,7);BOX(1,56)
   Z1$=HEX(0D)
   PRINT AT(5,8);"Processing + Printing completed.."
   PRINT AT(20,6);"Was it your intention to run this program again? (Y/N)"
   KEYIN Z1$
   IF Z1$="Y" THEN 1420
   IF Z1$="N" THEN 1430
   GOTO 1410
1420 RETURN CLEAR ALL
   Z1=1
   LOAD T "BASchart"
1430 $PSTAT=""
   RETURN CLEAR ALL
   :COM CLEAR
:LOAD T "START"

:ERRORPRINT HEX(03);AT(15,8);"UNABLE TO LOAD THE START PROGRAM"
:STOP

:END

1440 REM %

SUBROUTINE (DEFFN'15) ........

1450 REM search statement string for / locate and return position of
1460 REM '','','THEN' / compare statement against
1470 REM (FOR,NEXT,ELSE,REM)
1480 DEFFN'15
     T2$=AS
     E9=0
     F6=1
     3=0
     GOSUB '96
     IF P=1 THEN RETURN
     E2=POS(STR(AS,J)=HEX(20))
     NS=STR(AS,J,E2-J+1)
1490 IF T2$=" " THEN RETURN
1500 3=0
     GOSUB '96
     J=POS(STR(T2$,j)=HEX(20))
     GOSUB '96
     B1=3
     J=POS(STR(T2$,J)=HEX(20))
1510 IF J\]123 OR 12 THEN 3=2
     F$=STR(T2$,B10 -1)
     IF STR(F$,1,3)=HR THEN RETURN
     IF 1=6=0 THEN A$=T2$&S9$
     T2S=" 
     GOTO 1520
1520 IF E(=4 THEN 1620
     IF F4=0 THEN 1530
     W1=W1+1
     GOSUB 3820
     F4=1
1530 J=0
     GOSUB '96
     IF P=1 THEN RETURN
     B=J
     J=POS(STR(A$,J)=HEX(20))
     E1=J
     GOSUB '96
     B1=J
     J=POS(STR(A$,J)=HEX(20))

48
:IF J<123 OR J<2 THEN J=2
1540 E3=J
:K2$=STR(A$,B1,E3-1)
1550 IF K2$["IF" THEN 1580
:F5=0
1560 MAT SEARCH STR(A$,J),1=HEX(205448454E) TO A1$
 :J3=VAL(A1$,2)
 :A1$=ALL(00)
 :IF J3=0 THEN GOSUB '19
1570 IF F5=0 THEN RETURN
 :F5=0
 :J=J+5
 :T1$=""
 :GOSUB '95
1580 IF K2$["FOR" AND K2$["NEXT" THEN 1590
 :T1$=E9$
 :GOSUB '95
1590 IF K2$="FOR" THEN E=E+1
 :IF K2$="NEXT" THEN E=E-1
 :JO=0
 :JO=POS(82$=";"
 :IF JO=0 OR K2$="NEXT" THEN 1600
 :A1$=B2$
 :B2$=STR(A1$,1,JO-1)
 :A$=N$"" &K2$ &"" &STR(A1$,JO+1)
 :E9=1
1600 A2$=K2$
 :IF E=[4 THEN 1610
 :IF E9=1 THEN PRINT N$;";A2$;" ;B2$
 :ELSEPRINT STR(A$,1,124)
 :GOTO 1630
1610 E=E+1
 :F4=0
 :A2$=K2$
 :IF K2$="FOR" OR K2$="NEXT" THEN J=E3
 :GOSUB 1680
 :GOTO 1630
1620 GOSUB '10
1630 IF A2$=" FOR" THEN A2$="FOR"
 :IF A2$=" NEXT" THEN A2$="NEXT"
 :IF A2$="FOR" THEN F2$=STR(B2$,1,POS(STR(B2$,1)="=")-1)
 :GOSUB '98
 :F6=0
1640 GOTO 1490
1650 REM %
SUBROUTINE (DEFFN '10) .......

1660 REM locate / compare (FOR,NEXT,REM,GOTO,GOSUB)
1670 DEFFN '10

: J = 0
: GOSUB '96
: B = J
: J = POS(STR(A$, J) = HEX(20))
: E1 = J
: GOSUB '96
: IF P = 1 THEN RETURN
: B1 = J
: J = POS(STR(A$, J) = HEX(20))
: IF J[123 OR J[2 THEN J = 2
: E3 = J
: A2$ = STR(A$, B1, E3 - 1)

1680 S = 0
: IF A2$[FOR THEN 1690
: T1$ = S9$
: GOSUB '95
: GOSUB '30
: RETURN

1690 IF A2$[NEXT THEN 1710
: IF E9 = 1 THEN 1700
: T1$ = S9$
: GOSUB '95
: J0 = 0
: J0 = POS(B2$ = ",")
: IF J0 = 0 THEN 1700
: A$L = B2$
: B2$ = STR(A$L, 1, J0 - 1)
: A$ = N & " " & A2$ & " " & STR(A$L, J0 + 1)
: E9 = 1
: A$L = HEX(00)
: GOSUB '35
: RETURN

1700 GOSUB '35
: RETURN

1710 IF STR(A2$, 1, 1)[REM AND STR(A2$, 1, 1)[REM THEN 1720
: B2 = B1
: B2$ = STR(A$, B2)
: GOSUB '75
: RETURN

1720 IF A2$[GOTO THEN 1730
: T1$ = ""
: GOSUB '95
: O2$ = "GOTO"
: GOSUB '45
: RETURN
1730 IF STR(A2$,1,5)"GOSUB" THEN 1740
: T1$=""
: GOSUB '95
: D2$="GOSUB"
: GOSUB '45
: RETURN

1740 IF A2$="ON" THEN 1800
: T1$=""
: J=Bl-1
: GOSUB '95
: MAT SEARCH A$,="GOTO" TO A1$
: IF VAL(A1$,2)=0 THEN MAT SEARCH A$,="GOSUB" TO A1$
: J=VAL(A1$,2)
: A1$=ALL(00)
: IF J<> THEN 1750
: J=E3
: GOTO 1800

1750 C2$=STR(A$,E0+B2,J-(E0+B2+1))
: J=POS(-STR(A$,1,POS(-STR(A$,1,J-1)[HEX(20)])=HEX(20))
: GOSUB '95
: J=0
1760 J0=J0+1
: IF STR(B2$,J0,1)=" " THEN 1760
1770 IF STR(B2$,J0,4)"GOTO" THEN 1780
: T1$="S9$
: J=B2+E0-1
: GOSUB '95
: D2$="GOTO"
: GOTO 1790

1780 IF STR(B2$,J0,5)"GOSUB" THEN 1800
: T1$="S9$
: J=B2+E0-1
: GOSUB '95
: D2$="GOSUB"
1790 GOSUB '40
: RETURN

1800 IF S=0 THEN GOSUB '20
: RETURN

1810 REM %

SUBROUTINE (DEFFN'20) .......

******************************************************************************
1820 REM compare (IF,PRINT,INPUT,GOSUB,GOTO,STOP,END,RETURN)
1830 DEFFN '20
1840 IF A2$="IF" THEN 1950
: J=Bl+E3-1

51
:GOSUB '96
:IF P=1 THEN RETURN
:B2=J
:GOSUB 1560
:IF J3=0 THEN RETURN
:J=J+82
:C2$=STR(A$,B2,J3-1)
:T1$=""
:GOSUB '95
1850 D2$=B2$
:J=O
1860 J=J+1
:IF STR(02$,J,1)=" " THEN 1860
1870 132$=S1R (02$ J POS (SIR(02$ J) =HEM()) ) -1)
:02$=" 
:IF 132$=1PRINT " OR B2$="INPUT" OR B2$="PRINTUSING" OR B2$="KEYIN" OR B2$="LINPUT" OR B2$="GOTO" OR B2$="GOSUB" THEN D2$=B2$
:J=B2+8D-1
:GOTO 1930
1900 IF B2$]= ":" THEN 1920
:CONVERT B2$ TO N
:ERRORN=O
:GOTO 1920
191D N=O
:D2$="GOTD"
:J=B2+1
:GOTD 1930
192D N=O
:D2$=" 
:J=B2-1
1930 T1$=89$
:GOSUB '95
:GOSUB '6D
194D RETURN
1950 IF A2$="ELSE" OR A2$="ERROR" THEN GOSUB '65
:IF A2$="LOAD" OR A2$="STDP" OR A2$="END" THEN GOSUB '5D
:IF A2$="RETURN" AND STR(B2$,1,5)[ ]"CLEAR" THEN GOSUB 'B5
:IF S[ ]0 THEN RETURN
1960 T1$=89$
:IF STR(A2$,1,5)[ ]"PRINT" AND STR(A2$,1,1D)[ ]"PRINTUSING" THEN 1980
:J4=J+81-1
1970 J4=J4+1
:IF STR(A4,J4,1)=" " AND J4[81 THEN 1970
:IF J4=81 THEN B2$="1 BLANK LINE"
:IF J4[81 THEN GOSUB '95
:GOSUB '82
:RETURN

1980 IF STR(A2$1,5)="INPUT" AND STR(A2$1,5)="KEYIN" AND STR(A2$1,6)="INPUT" THEN J=81-1
:GOSUB '95

1990 GOSUB '80
:RETURN

2000 REM %

SUBROUTINE (DEFFN'98) .......

------------------------------
2010 REM compare (FOR,ELSE,NEXT)
2020 DEFFN'98
2030 IF A2$="FOR" THEN 2040

2040 IF A2$="NEXT" THEN 2060

2050 C=C+1
:IF C[51 THEN L1$(C)="FOR "&F2$
:IF C[51 THEN L2$(C)=STR(A$,B,E1-B)
:GOTO 2060

2060 FOR J5=B TO El-B

-------------------------------
2070 CONVERT STR(A$,B,E1-B) TO T1
:IF T1[=L AND F6=1 THEN GOSUB '91
:L=T1
:RETURN

2080 REM %

SUBROUTINE (DEFFN'60) .......

------------------------------
2090 REM print diamond symbol / compare (GOTO) / print-
2100 REM YES or NO option / compare (PRINT,INPUT,GOSUB) / print-
2110 REM PRINT or INPUT
2120 DEFFN'60

2130 CONVERT B2$ TO N1

2140 CONVERT N$ TO N
ERRORN1, N=0
:GOTO 2150

2150 K=E*20
:IF X0=1 THEN QOSUB '81
:IF Y0=1 THEN QOSUB '83
:S=1
:FOR S3=1 TO 2
:QOSUB '90
:PRINT TAB(K);"!"
:NEXT S3
:IF E=0 THEN K=1

2160 QOSUB '90
:PRINT TAB(K);"/";N$
:QOSUB '90
:PRINT TAB(K);"/"
:QOSUB '90
:PRINT TAB(K);"/"
:IF D2$["GOTO" THEN PRINT ""
:IF D2$["GOTO" THEN 2170
:IF N\N THEN PRINT TAB(K+36);"/"
:ELSEPRINT ""

2170 QOSUB '90
:PRINT TAB(K);"/";C2$;
:IF D2$["GOTO" THEN PRINT ""
:IF D2$["GOTO" THEN 2180
:IF N\N THEN PRINT TAB(K+36);"!"
:ELSEPRINT ""

2180 QOSUB '90
:PRINT TAB(K);"/" THEN ";

2190 IF D2$["GOTO" THEN 2240
:G=G+1
:PRINT ";
:IF N\N THEN PRINT TAB(K+36);"!"
:ELSEPRINT ""
:QOSUB '93
:PRINT TAB(K);"*":D2$;"[YES]"
:IF N\N THEN PRINT TAB(K+36);"!"
:ELSEPRINT ""
:QOSUB '90

2200 PRINT TAB(K);"*":D2$;TAB(K+11);
:PRINT "\--------------\--------"
:QOSUB '90
:PRINT TAB(K);"*/";
:IF N\N THEN PRINT TAB(K+38);"!"
:ELSEPRINT ""
:QOSUB '90

2210 PRINT TAB(K);"*/";
:IF N\N THEN PRINT TAB(K+38);"!"
:ELSEPRINT ""
:QOSUB '90

2220 PRINT TAB(K);"*/";
PROGRAM LISTING

2230 PRINT TAB(K);" [NO]*
:GOTO 2380

2240 IF D2$[]"KEYIN" AND D2$[]"INPUT" AND D2$[]"INPUT" THEN 2250
:PRINT ***
:GOSUB '93
:PRINT TAB(K);" * ";D2$;" / ";B2$
:GOSUB '90
:PRINT TAB(K);" * / "
:GOTO 2370

2250 IF D2$[]"PRINT" AND D2$[]"PRINTUSING" THEN 2290
2260 PRINT ***
:GOSUB '93
:PRINT TAB(K);" * / [YES] /============/" 
:GOSUB '90
:PRINT TAB(K);" * / /" 
2270 GOSUB '90
:PRINT TAB(K);" [NO]*
:GOTO 2380

2290 IF STR(D2$,1,5)["GOSUB" THEN 2330
2300 PRINT ***
:GOSUB '93
:PRINT TAB(K);" * / [YES] ................." 
:GOSUB '90
:PRINT TAB(K);" * / =========": ;"
2310 GOSUB '90
:PRINT TAB(K);" * / :";D2$;" ";B2$;TAB(K+36);": :
:GOSUB '90
:PRINT TAB(K);" * / : 
:GOSUB '90
:PRINT TAB(K);" * / : 
:GOSUB '90
:PRINT TAB(K);" * / : 
2320 GOSUB '90
Program Listing:

```
:PRINT TAB(K);"!"
:GOSUB '90
:PRINT TAB(K);"[NO]"
:GOTO 2380

2330 IF STR(B2$,1,6)="RETURN" AND STR(B2$,1,7)="RETURN" THEN 2360
:PRINT "*
:GOSUB '93
:PRINT TAB(K);"* / [YES] :" : : : : :
:GOSUB '90
:PRINT TAB(K);"* / ==]=]:: ::":
2340 GOSUB '90
:PRINT TAB(K);"* / :: RETURN :::
:GOSUB '90
:PRINT TAB(K);"* / :: :
:GOSUB '90
2350 GOSUB '90
:PRINT TAB(K);"!
:GOSUB '90
:PRINT TAB(K);"[NO]"
:GOTO 2380

2360 PRINT B2$
:GOSUB '93
:PRINT TAB(K);"* /"
:GOSUB '90
:PRINT TAB(K);"* /"
2370 GOSUB '90
:PRINT TAB(K);"* /"
:GOSUB '90
:PRINT TAB(K);"* /"
:GOSUB '90
:PRINT TAB(K);"* /"
2380 IF E=0 THEN K=0
:GOSUB '8
:RETURN

2390 PRINT
:PRINT "COMMENT, THIS /IF...THEN/ STATEMENT MAY BE TOO LONG TO PROC
ESS TO ITS EXTENT"
:PRINT
:W3=W3+1
:RETURN

2400 REM %

SUBROUTINE (OEFFN'30) .......

**************************************************
2410 REM print diamond (decision) / compare (FOR)
2420 DEFFN'30
```

Page 21
PROGRAM LISTING

2430 K=E*20
:IF X0=1 THEN GOSUB '81
:IF YO=1 THEN GOSUB '83
:S=1
2440 GOSUB '90
:PRINT TAB(K);"!
:GOSUB '90
:PRINT TAB(K);"
:GOSUB '90
:PRINT TAB(K);"/*";N$ 
:GOSUB '90
:PRINT TAB(K);"/"
2450 GOSUB '90
:PRINT TAB(K);"/
:GOSUB '90
:PRINT TAB(K);"/
:IF A2$="FOR" THEN A2$=" FOR"
:GOSUB '90
2460 PRINT TAB(K);"/";A2$;TAB(K+12);"*";B2$
:GOSUB '93
:PRINT TAB(K);"*"/--------]---------"
2470 GOSUB '90
:PRINT TAB(K);"*/";TAB(K+28);"!"
:GOSUB '90
:PRINT TAB(K);"*/";TAB(K+27);"
2480 GOSUB '90
:PRINT TAB(K);"*/";TAB(K+28);"!"
:GOSUB '90
:PRINT TAB(K);"*/";TAB(K+28);"!"
2490 E=E+1
:FO(E)=1
:RETURN

2500 REM %

SUBROUTINE (DEFFN'35) ........

2510 REM print diamond (decision) / compare (NEXT)
2520 DEFFN'35
2530 IF X0=1 THEN GOSUB '81
:IF YO=1 THEN GOSUB '83
:IF E=0 THEN RETURN
:FO(E)=0 
:E=E-1 
:K=E*20 
:S=1
2540 GOSUB '90
:PRINT TAB(K);"/*";TAB(K+28);"!"
:GOSUB '90
:PRINT TAB(K);"/*";TAB(K+28);"!"
:GOSUB '90

57

60
2550 PRINT TAB(K);"/ **;TAB(K+28);":"
:GOSUB '90
:PRINT TAB(K);"/ **;TAB(K+28);":"
2560 GOSUB '93
:PRINT TAB(K);"*---------[---------"
:GOSUB '90
:IF A2$="NEXT" THEN A2$=" NEXT"
2570 PRINT TAB(K);"* A2$;TAB(K+12);="/B2$
PEAT '90
:PRINT TAB(K);"* /"
:GOSUB '90
2580 PRINT TAB(K);"* /
:GOSUB '90
:PRINT TAB(K);"* /
:GOSUB '90
:PRINT TAB(K);"*/m
:GOSUB '8
:RETURN

2590 REM %

SUBROUTINE (DEFFN'40) ........

2600 REM locate position in string statement of ',,' / print -lines
2610 REM and arrows / print diamond with ON / compare (GOTO)
2620 DEFFN'40
  :IF X1=1 THEN GOSUB 2820
2630 W,JO=0
2640 V2=JO
  :W=W+1
  :J=0
2650 J=POS(STR(82$,V2+1)="","")
  :JO=JO+J
  :IF JO-V2-1=0 AND JO THEN V(W)=0
  :IF JO-V2-1=0 AND JO THEN 2640
  IF JO THEN CONVERT STR(B2$,V2+1,JO-V2-1) TO V(W)
2660 IF JO THEN 2640
  :CONVERT STR(B2$,V2+1,LEN(B2$)-V2) TO V(W)
2670 S=1
2680 IF X0=1 THEN GOSUB '81
  :IF Y0=1 THEN GOSUB '83
  :K=E*20
  :W2=W-1
  :GOSUB '90
  :PRINT TAB(K);"-];"
  :FOR Z=1 TO W2
  :PRINT "----";
  :NEXT Z
2690 PRINT "";
  :K3=W2*4+K
  :GOSUB '90
:PRINT TAB(K3);"!" &
:gosub '90
:PRINT TAB(K3);"/\";N$;
:gosub '90
:PRINT TAB(K3);" / "
2700gosub '90
:PRINT TAB(K3);" / "
:gosub '90
:PRINT TAB(K3);" / "
:gosub '90
:PRINT TAB(K3);" / "
2710gosub '90
:PRINT TAB(K3);" / "
:gosub '90
:PRINT TAB(K3);" / "
:gosub '90
:PRINT TAB(K3);" / "
:gosub '90
:PRINT TAB(K3);" / "
2720gosub '90
:PRINT TAB(K3);" / "
:gosub '90
:PRINT TAB(K3);" / "
:gosub '90
:PRINT TAB(K3);" / "
:gosub '90
:PRINT TAB(K3);" / "
2730gosub '93
:PRINT TAB(K);"-";
:for z=1 to w2
:print "---------";
:next z
:print ""
:gosub '90
:print tab(k);
:for z=1 to w
:print ";z;
:next z
2740gosub '90
:print tab(k);"
:for z=1 to w
:print ";
:next z
:print ""
2750gosub '90
:print tab(k);"
:for z=1 to w
:print "-----";
:next z
:print ""
:gosub '90
:for z=1 to w
:e$=""
2760convert v(z) to e$, (#####)
:print tab(z*8+k);str(e$,1,5);
:next z
PROGRAM LISTING

2770 PRINT " "
:GOSUB '90
:PRINT TAB(K);" ";
:FOR Z=1 TO W
:PRINT " ------";
:NEXT Z
:PRINT " "
:FOR F=1 TO 2
:GOSUB '90
:PRINT TAB(K);" ";
:FOR Z=1 TO W
:PRINT " ";
:NEXT Z
2780 PRINT " "
:NEXT F
:GOSUB '90
:PRINT TAB(K);" ";
:FOR Z=1 TO W2
:PRINT "--------";
:NEXT Z
:PRINT " "
:GOSUB '90
:PRINT TAB(K3);" "
:GOSUB '90
:PRINT TAB(K);" --[";
2790 FOR Z=1 TO W2
:PRINT "----";
:NEXT Z
:PRINT " ";
2800 IF 02$["GOTO" THEN 2810
:G=G+1
:82$=N$
:GOSUB '4
:RETURN

2810 GOSUB '8
:RETURN

2820 PRINT
:PRINT "COMMENT, THIS /ON..GOSUB(GOTO)/ STATEMENT MAYBE TOO LONG TO
PROCESS TO ITS EXTENT"
:PRINT
:AL$=82$
:J=POS(-82$="",)
:82$=STR(AL$,1,J-1)
:AL$=HEX(00)
:W3=W3+1
:RETURN

2830 REM %
SUBROUTINE (OEFFN'45) 

2840 REM print - block around GOTO / compare (GOTO)
2850 DEFFN'45
2860 S=1
:IF X0=1 THEN GOSUB '81
:IF Y0=1 THEN GOSUB '83
:K=E*20
:GOSUB '90
:PRINT TAB(K);"I el"
:GOSUB '90
:PRINT TAB(K);"IN"
:IF E=0 THEN K=1
2870 GOSUB '90
:PRINT TAB(K);".......................";N$
 :GOSUB '90
:PRINT TAB(K);": "
 :GOSUB '93
:IF 02$[]"GOTO" THEN 2880
:02$=" GOTO"
:G=G+1
2880 PRINT TAB(K);": ";02$;" ";B2$;TAB(K+14);": "
 :GOSUB '90
:PRINT TAB(K);": "
 :GOSUB '90
:PRINT TAB(K);": .....................": "
 :IF E=0 THEN K=0
2890 IF 02$[]"GOTO" THEN GOSUB '4
:ELSE GOSUB '8
:RETURN

2900 REM %

SUBROUTINE (OEFFN'50) 

2910 REM print - block around STOP or END or LOAD
2920 DEFFN'50
2930 U=1
:IF A2$[]"LOAD" THEN B2$=" "
:IF A2$[]"LOAD" THEN 2940
:J=E3
 :GOSUB '95
2940 S=1
:IF X0=1 THEN GOSUB '81
:IF Y0=1 THEN GOSUB '83
:K=E*20
:GOSUB '90
:PRINT TAB(K);" ! "
 :GOSUB '90
:PRINT TAB(K);" ! "
PROGRAM LISTING

:GOSUB '90
:IF E=0 THEN K=1
:PRINT TAB(K);" ";N$
:GOSUB '90
2950 PRINT TAB(K);" (--------)
:GOSUB '90
:PRINT TAB(K);" (--------)
:GOSUB '90
:PRINT TAB(K);" (--------)
:GOSUB '90
:PRINT
:GOSUB '90
:PRINT
:RI=1
:RETURN

2960 REM %

SUBROUTINE (DEFFN'65) ........

2970 REM print - block around ELSE OR ERROR/ print lines and arrows
2980 DEFFN'65
2990 IF X0=1 THEN GOSUB '81
3000 PRINT TAB(K);" ********* !ft
3010 J=E3
3020 REM %

SUBROUTINE (DEFFN'75) ........

3030 REM print the REMARKS statement
3040 DEFFN'75
3050 S=1
3060 IF X0=1 THEN GOSUB '81
PROGRAM LISTING

:IF NO=1 THEN GOSUB '83
  K=E*20
  GOSUB '90
  PRINT TAB(K);82$
  :RETURN

3060 REM %

SUBROUTINE (OEFFN'81) ...........

3070 REM print line then reference print routine
3080 DEFFN'81
3090 K=E*20
    :X0=0
    :IF E=0 THEN K=1
    :GOSUB '90
    :PRINT TAB(K);"...............:" 
    :IF E=0 THEN K=0
    :GOSUB '8
    :RETURN

3100 REM %

SUBROUTINE (OEFFN'85) ...........

3110 REM print block around RETURN/warning of no stop instr.before subroutine
3120 OEFFN'85
3130 U=1
    :K=E*20
    :IF X0=1 THEN GOSUB '81
    :IF Y0=1 THEN GOSUB '83
    :FOR S3=1 TO 2
    :PRINT TAB(K);": "
    :NEXT S3
    :IF E=0 THEN K=1
    :PRINT TAB(K);": ....................... ";N$
    :PRINT TAB(K);": "
    :PRINT TAB(K);": "
    :PRINT TAB(K);": "
    :PRINT TAB(K);": "
    :RETURN :" 
    :PRINT
    :S=1
3150 IF R1[0] THEN RETURN
    :PRINT
    :PRINT "COMMENT, THERE WAS NO /STOP/ INSTRUCTION BEFORE THIS SUBROUTINE"
    :W3=W3+1
    :PRINT
    :R1=1
PROGRAM LISTING

:RETURN
3160 REM %

SUBROUTINE (OEFFN'90) .......

 .................................................................
3170 REM print down line
3180 DEFFN'90
3190 IF E=0 THEN RETURN
3200 FOR K2=1 TO E
:PRINT TAB(K2*20-12);"!";
 :NEXT K2
:RETURN
3210 REM %

SUBROUTINE (OEFFN'91) .......

 .................................................................
3220 REM print warning of instruction No.
3230 DEFFN'91
3240 J1=1
:IF T1=L THEN RETURN
 :PRINT
:PRINT "COMMENT, EITHER INSTRUCTION NUMBER ";TWOR";
 :NEXT K2
 :RETURN
3260 REM %

SUBROUTINE (DEFFN'92) .......

 .................................................................
3270 REM print line sequence error, abort
3280 DEFFN'92
3290 PRINT
 :PRINT "COMMENT, INSTRUCTION ";STR(A$,1)
 :PRINT " DOES NOT HAVE A CORRECT INSTRUCTION NUMBER."
 :PRINT
 :W3=W3+1
 :RETURN
3300 REM %

SUBROUTINE (OEFFN'93) .......

 .................................................................
3310 REM print up or down arrows
3320 DEFFN'93
3330 IF E = 0 THEN RETURN
3340 FOR K2 = 1 TO E
   : IF FO(K2) = 1 THEN PRINT TAB(K2*20-13); "/: ";
   : ELSE PRINT TAB(K2*20-13); "!/ ";
   : NEXT K2
   : RETURN

3350 REM %

SUBROUTINE (DEFFN'95) .......

3360 REM locate and save line following the word after the value J: REM
if can't then save the word after J
3370 DEFFN'95
3380 GOSUB '96
   : IF P = 1 THEN RETURN
   : B2 = J
   : E4 = POS(-A$ = HEX(0D))
   : IF E4 = 0 THEN E4 = POS(-A$[ ]HEX(20)) + 1
   : EO = POS(STR(A$, B2) = HEX(20))
   : IF EO = 115 THEN XL = 1
   : IF EO + B2 = E4 THEN 3400
   : B2$ = STR(A$, EO + B2, E4 - (EO + B2))
3390 RETURN

3400 IF B2] = E4 THEN 3410
   : B2$ = STR(A$, B2)
   : RETURN

3410 B2$ = " 
   : RETURN

3420 REM %

SUBROUTINE (DEFFN'96) .......

3430 REM print warning of statement error, abort
3440 DEFFN'96
3450 J = J + 1
   : IF J > 124 THEN 3460
   : IF STR(A$, J, 1) = " " THEN 3450
   : IF J = 124 THEN RETURN
3460 PRINT
3470 PRINT "COMMENT, NOT A COMPLETE BASIC INSTRUCTION"
   : P = 1
   : W3 = W3 + 1
   : PRINT A$
   : RETURN

3480 REM %
SUBROUTINE (DEFFN'99) .......

3490 REM print line sequence reference error, abort
3500 DEFFN'99
3510 \$3=W3+1
:PRINT
:PRINT "COMMENT, THIS PROGRAM HAS A LOGIC ERROR"
:PRINT "/";A2$;"/ OF LINE NUMBER ";N$
:IF C=0 THEN RETURN
:PRINT " DOES NOT MATCH UP WITH ";L1$(C);"/ OF LINE NUMBER ";L2$(C)
3520 RETURN
3530 REM %

SUBROUTINE (DEFFN'80) .......

3540 REM print block around PROCESS / come and print (PRINT,INPUT)
3550 DEFFN'80
 \$K=E*20
 :IF \$0=1 THEN GOSUB '83
3560 IF X0[0] THEN 3580
 :FOR \$3=1 TO 2
 :GOSUB '90
 :PRINT TAB(K);!
 :NEXT \$3
 :IF \$E=0 THEN \$K=1
 :GOSUB '90
3570 PRINT TAB(K);".................";N$
 :GOSUB '90
 :PRINT TAB(K);": PROCESS :
 :GOSUB '90
 :PRINT TAB(K);".................:";
 \$X0=1
 \$Z0=1
3580 IF \$E=0 THEN \$K=1
 :GOSUB '90
 :PRINT TAB(K);":-------------:
 :IF \$Z0=1 THEN GOSUB '93
 :IF \$Z0=8 THEN \$Z=0
 :IF \$Z[1] THEN GOSUB '90
3590 \$Z=\$Z+1
 :$TRAN(B2$,"""
 :$TRAN(A2$,"""
3600 IF A2$="INPUT" OR A2$="KEYIN" OR A2$="INPUT" THEN PRINT TAB(K);":
 I/O ";A2$;TAB(K+14);": ";B2$
3610 IF STR(A2$,1,2)=STR(B2$,1,2) THEN B2$=""
3620 IF A2$["INPUT" AND A2$["KEYIN" AND A2$["INPUT" THEN PRINT TAB(K);
 ";";STR(A$,B,E1-B);TAB(K+14);": ";A2$;" ";B2$
 :RETURN

66 69
SUBROUTINE (DEFFN'8) .........

REM print down arrow-line
DEFFN'8
GOSUB '90
PRINT TAB(K);"!
GOSUB '90
PRINT TAB(K);" V"
GOSUB '90
PRINT TAB(K);"!
GOSUB '90
PRINT TAB(K);"!
RETURN

SUMROUTINE (DEFFN'4) .........

REM print line and arrows, horizontal
DEFFN'4
N,N1=0
CONVERT B2$ TO N1
ERROR(N,N1=0
GOTO 3700
CONVERT N$ TO N
ERROR(N,N1=0
GOTO 3700
GOSUB '90
PRINT TAB(K);"!
IF N1[N THEN PRINT TAB(K+34);"!"
ELSEPRINT ""
GOSUB '90
PRINT TAB(K);"V"
IF N1[N THEN PRINT TAB(K+34);"!"
ELSEPRINT ""
GOSUB '90
PRINT TAB(K);"!
IF N1[N THEN PRINT TAB(K+34);"!"
ELSEPRINT ""
GOSUB '90
PRINT TAB(K);"!
IF N1[N THEN PRINT TAB(K+34);"!"
ELSEPRINT ""
GOSUB '90
PRINT TAB(K);"!
IF N1[N THEN PRINT TAB(K+34);"!"
ELSEPRINT ""
GOSUB '90
PRINT TAB(K);"------------------------]-----"
PROGRAM LISTING

:ELSEPRINT " "
:GOSUB '90
:IF NL"N THEN PRINT TAB(K+34);
:ELSEPRINT " "
:GOSUB '90

3740 U=1
:IF U6=0 THEN RETURN
:U6,U=0
:E=E-1
:IF E<0 THEN PRINT "!"
:RETURN

3750 IF U{1 THEN 3760
:K=E*20
:IF E=0 THEN K=0
:U=0

3760 IF E[1 THEN PRINT TAB(K);" "
:ELSEPRINT TAB(K);" "

3770 GOSUB '90
:PRINT TAB(K);" "
:GOSUB '90
:PRINT TAB(K);" "
:GOSUB '90
:PRINT TAB(K);" "
:GOSUB '90
:PRINT TAB(K);" "
:RETURN

3780 REM %

SUBROUTINE (OEFN'19) .......

.........................

3790 REM print illegal use of IF,THEN, / print module [==] unstructured
3800 OEFN'19
:w3=w3+1
:PRINT
:PRINT "CONVENT, CANNOT READ /IF...THEN/ IN STATEMENT (MAY BE TO L
ONG TO PROCESS)"
:PRINT "OR CANNOT MATCH /FOR/ OR /NEXT/ WITH IT'S LOOP CO
MPLEMENT"
:PRINT STR(A$11)
3810 RETURN

3820 K=E*20
:GOSUB '90
:PRINT TAB(K);" "
:GOSUB '90
:PRINT TAB(K);" "
:GOSUB '90
:PRINT TAB(K);" "
:GOSUB '90
:PRINT TAB(K);" "
:GOSUB '90
PROGRAM LISTING

:PRINT TAB(K);" / **
3830 GOSUB '90
 :PRINT TAB(K);" / MODULE **
 :GOSUB '93
 :PRINT TAB(K);" / "W1;TAB(K+13);"**
 :GOSUB '90
 :PRINT TAB(K);" / **
 :GOSUB '8
3840 RETURN

3850 REM %

SUBROUTINE (DEFFN '82) ......

3860 DEFFN '82
 :K=E*20
 :IF X0=1 THEN GOSUB '81
 :IF Y0[0] THEN 3890
 :FOR S3=1 TO 2
 :GOSUB '90
 :PRINT TAB(K);" !" 
 :NEXT S3
 :IF E=0 THEN K=1
 :GOSUB '90
3870 PRINT TAB(K);" /===============/ ";N$
 :GOSUB '90
 :PRINT TAB(K);" / --------- /"
 :GOSUB '90
 :PRINT TAB(K);"( ( PRINTING ( "
 :GOSUB '90
 :PRINT TAB(K);" * --------- **
 :GOSUB '90
 :PRINT TAB(K);" *-------------**
3880 Y0=1
 :X=1
 :Z0=0
3890 IF E=0 THEN K=1
 :IF Z0=1 THEN GOSUB '93
 :IF Z0=0 THEN Z0=0
 :IF Z0[0] THEN GOSUB '90
 :Z0=Z0+1
 :$TRAN(B2$,".." ).." ).."
 :IF B2$="PRINT.." THEN B2$="1 BLANK LINE"
 :IF MOD(X,2)=0. THEN 3910
3900 PRINT TAB(K);" )------------------")" 
 :GOSUB '90
 :PRINT TAB(K);" / /"
 :GOSUB '90
 :IF A2$="PRINT" THEN PRINT TAB(K);" / PRINT / ";B2$
 :ELSEPRINT TAB(K);" / PRINTUSING / ";B2$
 :GOTO 3920

69
72
Technical Note 3-82

10/25/82  BASCHART  PAGE 35

PROGRAM LISTING

3910 PRINT TAB(K);"(-------------(" 
    :GOSUB '90 
    :PRINT TAB(K);" * " 
    :GOSUB '90 
    :IF A2$="PRINT" THEN PRINT TAB(K);" * PRINT ";B2$ 
    :ELSEPRINT TAB(K);" * PRINTUSING ";B2$ 
3920 X=X+1 
    :RETURN 

3930 REM %

SUBROUTINE (DEFFN '83) ......

............................

3940 DEFFN '83 
    :K=E*20 
    :YO=0 
    :IF E=0 THEN K=1 
    :GOSUB '90 
    :IF MOD(X,2)=0 THEN PRINT TAB(K);"/=" 
    :ELSEPRINT TAB(K);" " 
    :IF E=0 THEN K=0 
    :GOSUB '8 
    :RETURN 

3950 REM %

SUBROUTINE (DEFFN '66) ......

............................

3960 DEFFN '66 
    :IF U6=0 THEN RETURN 
    :IF U6=2 AND STR(A$,1,J-1)=N$ THEN RETURN 
    :U6=0 
    :IF X0=1 THEN GOSUB '81 
    :IF Y0=1 THEN GOSUB '83 
    :E=E-1 
    :K=E*20 
    :IF E=0 THEN K=1 
    :PRINT TAB(K);" " 
    :RETURN 

9500 REM %

GENERAL SUBROUTINE XXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 
    :PRINT HEX(D30E);"YOU HAVE LOAD=0 'DEFFNO' THE GENERAL SUBROUTINES"; 
    HEX(DA) 
    :LIST 
    :END
9530 OEFFN 'O"SCRATCHT";HEX(22);"BASCHART";HEX(22);"SAVE T$ ()";HEX(22)
    ;"BASCHART";HEX(2200)
9570 OEFFN '31
    :GOTO 1430