BREAK

BREAK

Format
BREAK(line-number-list)

Cross Reference
CONTINUE, ON BREAK,   UNBREAK

Description
The BREAK instruction sets a breakpoint at each program statement you specify. When the computer encounters a line at which you have set a breakpoint, your program stops running before that statement is executed.

BREAK is a valuable debugging aid. You can use BREAK to stop your program at a specific program line, so that you can check the values of variables at that point.

You can use BREAK line-number-list as either a program statement or a command.

The line-number-list consists of one or more line numbers, separated by commas. When a BREAK instruction is executed, breakpoints are set at the specified program lines. If you use BREAK as a program statement, line-number-list is optional. When a BREAK statement with no line-number-list is encountered, the computer stops running the program at that point.

If you use BREAK as a command, you must include a line-number-list.

Breakpoints

When your program stops at a breakpoint, the message Breakpoint in line number is displayed. While your program is stopped at a breakpoint, you can enter any valid command.

To resume program execution starting with the line at which the break occurred, enter the CONTINUE command. However, if you edit your program(add, delete or change a program statement) you cannot use CONTINUE. This prevents errors that could result from resuming execution in the middle of a revised program. You also cannot use CONTINUE if you enter a MERGE or SAVE command or a LIST command with the file-specification option. Note that pressing CLEAR also causes a breakpoint to occur before the execution of the of the next program statement.

When your program stops at a breakpoint, the computer performs the following operations:

It restores the default character definitions of all ASCII characters   from 33 thru 126.

It restores the default foreground-color and background-color to all characters.

It restores the default screen color.