

17.1 Introduction

QBASIC does not take line numbers instead, uses line numbers as labels. GOTO statements seek these labels while executing the programs. We will discuss about some important statements (CLS, REM, INPUT, LET, PRINT AND END), which are frequently used in QBASIC programs with illustrations.

Type the following program after loading QBASIC.

Example 1

```

CLS
INPUT "Enter first number";A
INPUT "Enter the second number";B
LET P=A*B
LET D =A-B
PRINT "SUM OF A AND B IS";A+B
PRINT "PRODUCT OF A AND B IS";P
PRINT "DIFFERENCE BETWEEN A AND B IS";D
END

```

Press Alt key and select Run option

Select Start. The program executes similar to GWBASIC fashion.

Message: Press any key to continue.

Instruction: Press a key to continue.

Now, save the file as follows.

- Press Alt+F (Select File option from menu.)
- Select Save from the option menu.
- A dialog box will be given.
- Type the filename (11 characters) as **MYPRG1** and save in the default drive and path.
The file will have **.BAS** extension automatically.
- The extension of the file **.BAS** is given by the system automatically.

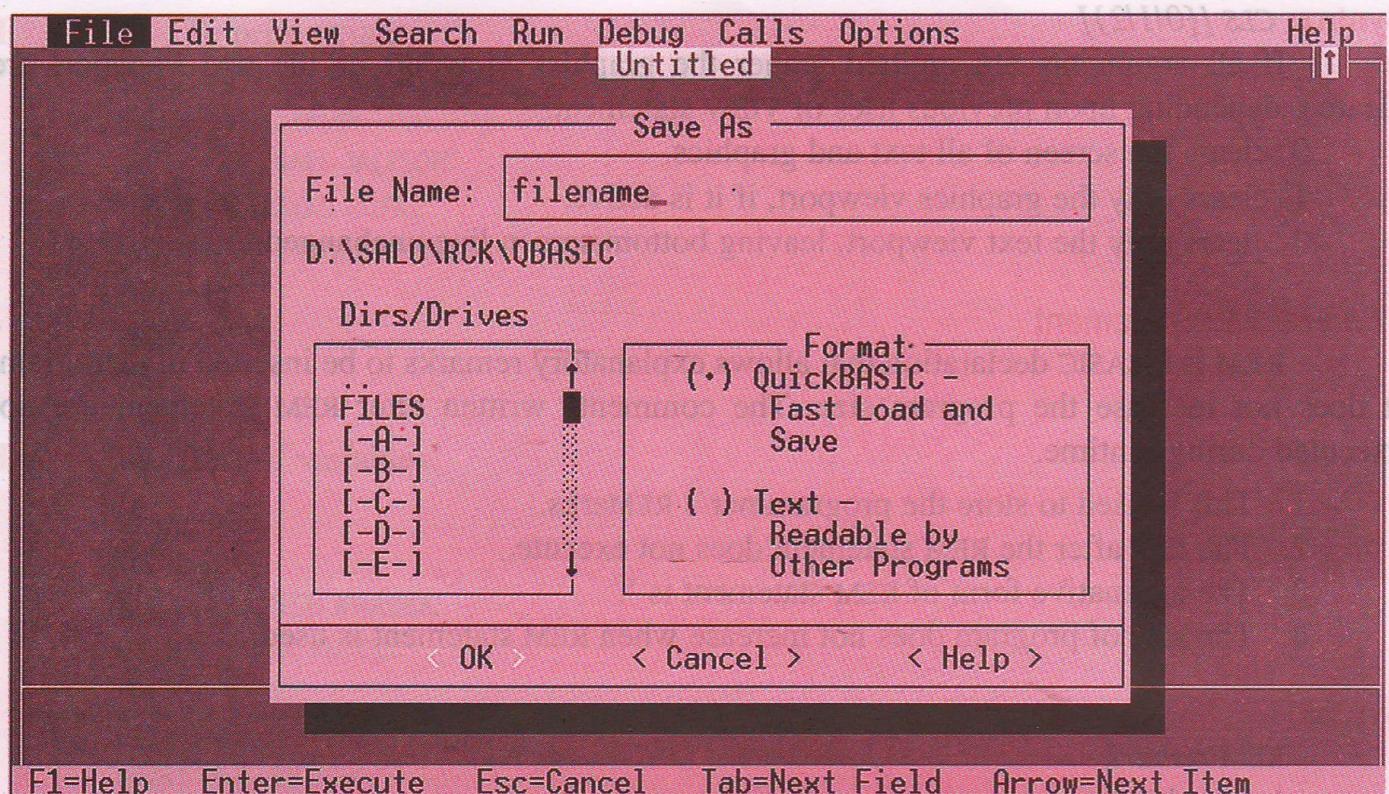


Fig. 15.1: Save as dialog box

Example 2

```

CLS
REM My first program in Qbasic
PRINT "MY NAME IS KISHORE"
PRINT "MY CLASS IS 9F"
INPUT "What is your name ";N$
INPUT "In which class do you study ";C
PRINT:PRINT:PRINT
PRINT "My name is:";N$
PRINT "I study in:";C$
PRINT "Thank you very much being a friend."
END

```

In the above program segments, we have used CLS, REM, LET, PRINT, INPUT and END statements. These statements are the most common statements. Before we discuss on other statements, let us examine the syntax and uses of these statements.

17.2 The CLS Command

This is a device I/O statement.

This clears the display screen.

Syntax: *CLS [{0/1/2}]*

If all arguments are omitted, either the graphics viewport or the text viewport are cleared, depending upon previous uses of VIEW statement.

- 0 clears the screen of all text and graphics.
- 1 clears only the graphics viewport, if it is active.
- 2 clears only the text viewport, leaving bottom screen line unchanged.

17.3 The REM Statement

REM is a BASIC declaration that allows explanatory remarks to be inserted in a program. It does not increase the program size. The comments written after REM statement are not executed during runtime.

1. This is used to store the programmer's REMarks.
2. The text after the REM statement does not execute.
3. The alternative form of REM statement is '.
4. The size of program does not increase when REM statement is used.

Syntax:

REM remark
‘ *remark*

Remark is any explanation you think will be useful to a reader of the program text and which will fit on the rest of the program line

17.4 The LET Statement

This is the expression evaluation statement that assigns the value of an expression to a variable. Let is optionally used in the program.

1. This is assignment statement.
2. The values or strings on the right of equal to (=) sign are assigned to the left.
3. The calculation on the right of equal to (=) sign is stored to the left.

Syntax: *[LET] variable=expression*

variable represents a variable in which data will be assigned.

Expressions are the numeric or string (data) expressions.

Example 3: Calculate simple interest.

```
LET P=5000
LET N = 5
LET R = 8.5
LET I = (P*N*R)/100
LET A=P+I
PRINT "Deposit";P
PRINT "Number of years";N
PRINT "Rate of Interest is";R
PRINT "Interest";I
PRINT "Amount";A
END
```

Example 4: To print strings.

```
CLS  
PRINT "THIS STRING CONSTANT TEST"  
LET A$="MICHAEL JACKSON"  
LET B$="NARAYAN GOPAL"  
LET C$="KATHMANDU, NEPAL"  
PRINT A$  
PRINT B$, C$  
END
```

Example 5: Test of variables.

```
LET A= 60  
LET B = 100  
LET A$="RAMESH KHADKA"  
LET B$="SUNIT KHAREL"  
LET A = 10000  
LET B = 0.1  
PRINT "SALARY =";A  
PRINT "PROVIDENT FUND=";A*B  
PRINT A$, B$  
END
```

In the above program, the variables A and B are repeated. They are replaced by new values. The last values assigned are functional in the variables. Hence, the value of A will be 10000 and that of B will be 0.1 in this program.

17.5 The PRINT Statement

This is a device I/O statement that outputs data on the screen

1. This statement is used to print the strings or numbers.
2. The strings are enclosed in double quotes.
3. The numbers are not enclosed in double quotes.
4. The alternative form of the PRINT statement is: ?.
5. The variables are also not enclosed in quotes.

Syntax: *PRINT [expressionlist][{, /;}]*

If all arguments are omitted, a blank line is printed.

If *expression list* is included, the values of the expressions are printed on the screen.

The expressions in the list may be numeric or string expressions. (String literals must be enclosed in quotation marks.)

expression anything (string, number or variable)

- ;* can be used to separate two expressions so that it prints at the gap of one space. In fact each character leaves one space before and after.
, the expressions can be separated with comma (13 spaces; including the space before and after a character it counts 15.).

Syntax: Other Uses of the PRINT Keyword are:

- (a) PRINT USING to output formatted text to the screen
- (b) PRINT# to output data to a sequential file which will be discussed in coming sections.

17.6 The END Statement

This is a BASIC declaration that ends a BASIC program, procedure, or block

1. This statement is usually written at the end of the program.
2. It may be written in the middle of the program also.
3. The END statement terminates the program.

executed during runtime.

Syntax:

`END [{DEF | FUNCTION | IF | SELECT | SUB | TYPE}]`

Where,

END DEF ends DEF,
END FUNCTION ends FUNCTION
END SELECT ends SELECT
END TYPE ends TYPE.

Without any parameters, it simply ends a program.

Note: END DEF, END FUNCTION, END SELECT & END TYPE are discussed in respective sections.

17.7 The INPUT Statement

INPUT is a device I/O statement that reads input from the keyboard during program execution and stores it into a list of variables.

1. This allows the user to enter data from keyboard.
2. The INPUT is followed with a variable (string or numeric).
3. INPUT statement prompts with a question mark when program executes.
4. The comment can be enclosed in double quotes.

Syntax: `INPUT[;][“prompt string”{;|,}]{variable list}`

Variable list is one or more variables separated by commas.

Prompt string, if used, is displayed on the screen to tell the user of the program what to enter at the keyboard.

While entering data by INPUT statement, the ; is written immediately after the prompt string. The comma (,) is used while multiple data is in single line input.

Other Uses of the INPUT Keyword are:

- (a) LINE INPUT
 - to read an entire line of input from the keyboard and store it in one string variable
- (b) INPUT #
 - to retrieve data items (or “fields”) from a sequential file record and store them into a list of variables
- (c) LINE INPUT #
 - to retrieve an entire record from a sequential file and store it into one string variable

Example 6

```
Input "Enter name of a person";n$  
Input "Enter makrsk";m1, m2, m3  
Input "Enter number";n
```

17.8 Immediate Mode Commands

Some commands in QBASIC can be executed in IMMEDIATE mode. Press F6 to come into IMMEDIATE mode.

17.8.1 Listing files

The FILES Command: The files command displays the files on the disk. The directories and sub-directories are also displayed on the screen.

Syntax: FILES *file specification*

Example 7

FILES	Lists files from the default directory (drive & path).
FILES "A:\	Lists files from drive B:.
FILES "B:\QBASIC*.BAS	Lists files from GWBASIC directory of drive B:..

17.8.2 Changing Directory

The CHDIR Command: This command is used to change a directory. This command changes the default directory on the specified drive.

Syntax: CHDIR *path specification*

Example 8

CHDIR "A:\QBASIC	Changes directory to A:\QBASIC
CHDIR "B:\QBASIC	Changes directory to B:\QBASIC

17.8.3 Creating New Directory

The MKDIR Command: This command is used to create a new directory. A directory on a duplicate filename or directory name cannot be created. The syntax is:

Syntax: MKDIR *path specification*

Path specification is a string expression that identifies the directory to be created

Example 9

MKDIR "A:\FILES	Creates a directory FILES in drive A:.
MKDIR "B:\BASICA	Creates a directory BASICA in drive B:..

To exit from QBASIC, either you type system on immediate mode or select Exit option from File option menu.

17.8.4 Removing a Directory

The RMDIR Command: This command is used to remove a directory.

The directory must be empty.

If the directory is not empty, delete the files from the directory and execute the command. An attempt to remove a directory that contains files result in error.

Syntax: RMDIR *path specification*

Path specification is a string expression that identifies the directory to be created

Example 10

RMDIR "A:\FILES"

Removes FILES directory from drive A:.

RMDIR "B:\BASICA"

Removes BASICA directory from drive B:.

17.8.5 Erasing the Files

The KILL Command: KILL is a statement that uses DOS to delete a disk file

This command is used to erase the files.

If the files to be deleted are not in the current directory, you have to change the directory.

Syntax: KILL "*file specification*"

File specification is a string expression that identifies the file to delete.

The file specification may include a path specification.

Example 11

KILL "A:\PROGRAM.BAS"

Erases PROGRAM.BAS from drive A:.

KILL "A:\QBASIC\PROG100.BAS"

Erases PROG100.BAS from GWBASIC directory of drive A:.

17.8.6 Quick Tour to DOS

The SHELL Command: SHELL is a statement that interfaces with DOS to exit the BASIC program, run a .COM, .EXE, or .BAT program or a DOS command and return to the program at the line following the SHELL statement

Syntax: SHELL [*command string*]

If all arguments are omitted, DOS will give you a new COMMAND.COM.

SHELL and *DOS commands* can be entered at the DOS prompt.

Command string is a string expression that contains the name of a program to run any program options.

Example 12

SHELL

To return back QBASIC editing screen, type EXIT and press enter key.

17.8.7 Changing the Filename

The NAME Command: NAME is a statement that uses DOS to change the name of a disk file or directory.

Syntax: NAME *oldfilename* AS *newfilename*

oldfilename, a string expression, is the name of an existing file.

newfilename, a string expression, must be a filename that does not exist.

Tip: *Oldfilename* and/or *newfilename* may contain a pathname. If the path in *oldfilename* is different than the path in *newfilename*, the NAME statement moves the file and renames it. Both files must be on the same drive.

Example 13

NAME "ABC.BAS" AS "PRG101.BAS"

While supplying filename, the name of the file must be enclosed in double quotes. Both the *oldfilename* and *newfilename* must be separately enclosed in double quote and that must be complete.

17.9 Opening a File

- To open a file, select **File** option from the main menu.
- Select **Open program** option. You will get the following screen.
- Select a file from the files listed.

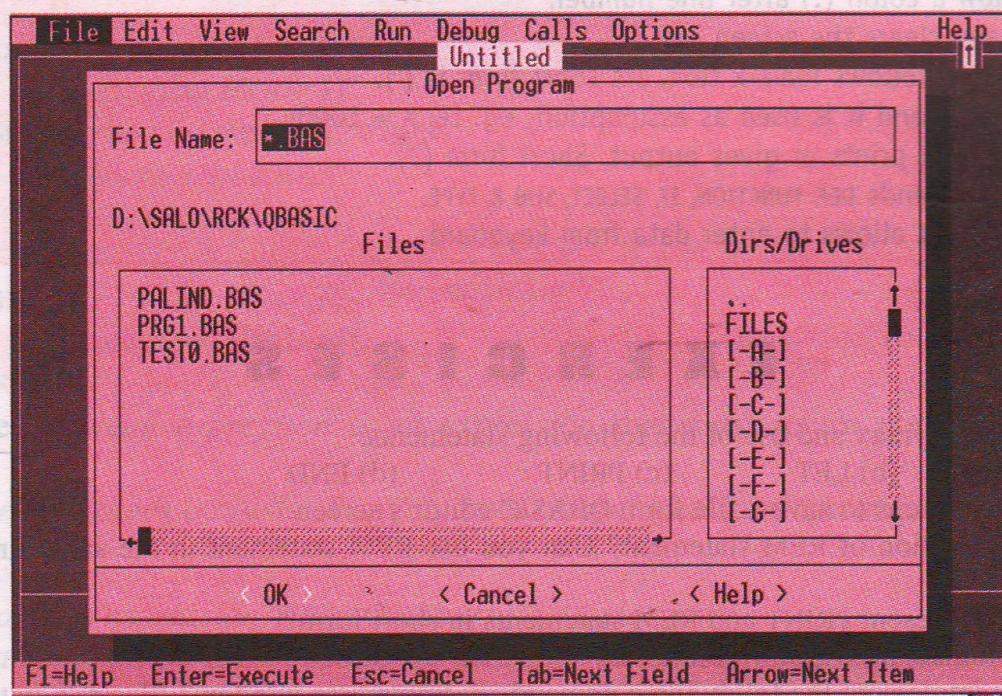


Fig. 15.2: Opening program

17.10 Loading a File

To load a file from a directory and path,

- Select **File** option from menu.
- Select **Load file**. You will get the following screen.
- Select a file by pressing **tab** key and press enter key to load the file selected.

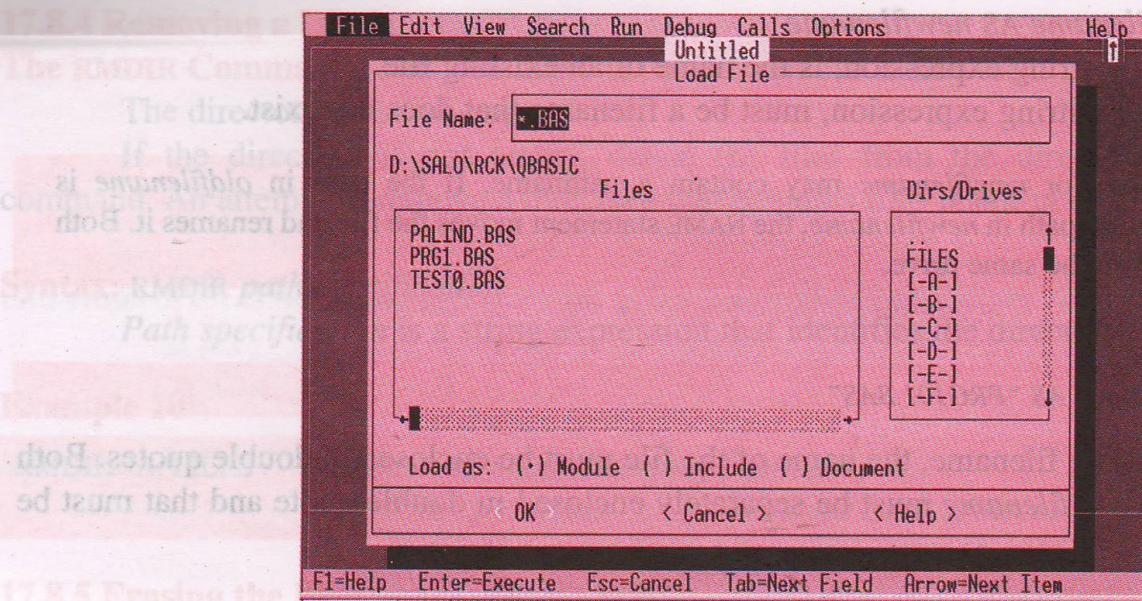


Fig. 15.3: Loading file from the disk

MAIN POINTS TO REMEMBER

- Line numbers are not used in QBASIC. If they are used, they need not be in increasing order. Line labels follow a colon (:) after line number.
- CLS command clears the screen.
- REM statement follows remark text. Its short form is (').
- LET is optional and it is used as assumption, eg. LET A = 1000.
- PRINT statement prints or gives output. Short form (?).
- END statement ends DEF, FUNCTION, IF, SELECT, SUB & TYPE.
- INPUT statement allows to enter data from keyboard.

EXERCISES

- Write down the syntax and use of the following statements.
(a) INPUT (b) LET (c) PRINT (d) END
- Write down the steps to save a file from QBASIC editor's screen.
- What is the function of REM statement? Can you use REM statement in the same line with INPUT statement?
- How do you represent PRINT and REM statements in short form?
- What is an immediate window command?
- How do you implement the following commands in Immediate window? Illustrate with example.
(a) FILES (b) CHDIR (c) MKDIR (d) RMDIR (e) KILL
- What is the function of SHELL command? To return back into Qbasic editor, what should be done after executing SHELL command?
- Can you rename a file from Qbasic environment? Illustrate with an example.
- Write down the steps to
(a) Open a file (b) To load a file
- Find the errors in the following and correct them.
(a) PRINT MY NAME IS SUNIT" (b) LET A\$=3456
(c) LET "ADDRESS\$"=KATHMANDU (d) INPUT "ENTER NAME";A
(e) FILESA:\ (f) CHDIR"C:\QBASIC*.*"