

The following is a list of the keywords, their syntax and an example of the use of each. In most cases, the parameters are numeric expressions, string expressions or literals (refer to section 3 of the HP-71 Owner's Manual).

DELETE# channel number, record number

The **DELETE#** statement deletes the specified record from the text file associated with the channel number. Channel numbers are assigned with the **ASSIGN#** statement (refer to section 14 of the HP-71 Owner's Manual). Record numbers always begin at 0, so line number 1 is record number 0.

The channel number and the record number are numeric expressions, rounded to integer values.

DELETE# generates an error message if the assigned file is external, protected or not a text file.

EXAMPLE: **DELETE# 11,14** deletes record number 14 from the file associated with channel 11.

FILESZR (filename)

The **FILESZR** function returns the number of records in the specified text file if that file exists. The filename parameter is a string expression. If an error is detected, the negated error number is returned so that you can tell the difference between an error and the number of records. If filename contains an illegal port specifier, such as **FROGS:PORT(8)**, the error message Invalid Filespec is generated.

EXAMPLE:**FILESZR ('FROGS')** returns the number of records in the file **FROGS**.

INSERT# channel number, record number;new record

The **INSERT#** statement inserts the new record immediately before the specified record number in the file associated with the specified channel number. The channel number must first be assigned to the file using the **ASSIGN#** statement. Record numbers always begin at 0, so line 1 is record 0.

The new record must be a string expression. The channel number and the record number are numeric expressions, rounded to integer values.

INSERT# generates an error if the file is external, protected or not a text file.

EXAMPLE: **INSERT# 11,35;"This is the new line being inserted."** inserts the string before record 35 (line 36) of the file associated with channel 11. The old record 35 becomes record 36.

LIST filename [begin line[end line]]

The **LIST** statement lists a text file. Depending on the parameters you specify, it lists either the entire file, a single line, or a range of lines. Line numbers are specified using integer constants. The line number parameters are optional, and the whole file is listed if they are not included. Refer to **LIST** in the HP-71 Reference Manual for details.

REPLACE# channel number, record number;new record

The REPLACE# statement replaces the record indicated by record number with the new record. The channel number must first be assigned to the file by using the ASSIGN# statement. Record numbers always begin at 0, so line number 1 is record 0.

The new record is a string expression. The channel number and the record number are numeric expressions rounded to integer values.

REPLACE# returns an error if the file is external, protected or not a text file.

EXAMPLE: REPLACE# 11,35;"This line will replace the old line."replaces record 35 of the file associated with channel 11. Old record 35 no longer exists.

SEARCH (search string, column, begin record, end record, channel number)

The SEARCH function searches the file associated with the indicated channel number for the search string, beginning with the specified column and record number. The search continues through the end record specified. If the search is successful, SEARCH returns a value in the form nnn.ccclll, where nnn is the record number, ccc is the column is the column number and lll is the length of matched string. If the search is unsuccessful, SEARCH returns a zero.

The search string can be any string expression, and can contain the special pattern characters discussed on the next page. The other parameters are numeric expressions rounded to integer values.

EXAMPLE: Suppose that channel 11 has been assigned to the file FROGS and the string 'frogs are green' appears beginning in column 8 of line 36.

```
A=SEARCH("frogs are green",1,1,9999,11)
```

searches the file FROGS, beginning with column 1 of rec. 1 through rec. 9999, for the search string and returns the value 35.008015 in A.

**Note that since the first line is record 0, line 36 is actually record 35.

Cont. next page

71-00006

SPECIAL PATTERNS

A feature of SEARCH is the availability of four characters that have special meaning when used in patterns. Using these characters in a search string tells SEARCH to look, for example, only for those occurrences of the string at the beginning of the line, or at the end of the line, or allow any pattern between two specified patterns. The four characters that can be used in this special way are ., @, ^, \$.

The backslash (\) character can be used like a "switch" in the search string to start and stop this feature that makes these four characters take on special meaning. The backslash character is CHR\$(92), and for convenience, may be assigned to a key by executing:

```
DEF KEY <key name>,CHR$(92);
```

(See page 69 in the HP-71 Reference Manual for further information about key assignments). The first occurrence of the backslash turns on the feature, so that the four characters take on their special meanings. The next occurrence of the backslash turns this feature off.

The four characters, their meanings, and some examples of their use are described in the following paragraphs. In all the examples, assume that the specified file is open to channel number 3. Also, all the examples specify the search to start in record zero, column 1 (the start of the file), and to continue through record number 9999.

- 1) The period (.) is a 'wild card' character. SEARCH looks for the specified string, but any character can be in those positions in the string where you put a period.

```
Example: SEARCH("ABC"&CHR$(92)&"...W",1,0,9999,3)
```

Looks for the first occurrence of ABC followed by any three characters, followed by W. Possibilities are ABC999W, ABCzyzW, or ABC yzW

- 2) The commercial "at" symbol (@) indicates that any number of characters between the beginning of a string and the end of a string on the same line are 'wild cards' -- that is, there can be any number of characters-- you don't have to specify how many characters or what they are. Because SEARCH starts looking for the end of the string at the end of the line, the longest match is found.

```
Example: SEARCH("ABC"&CHR$(92)&"@CDE",1,0,9999,3)
```

Looks for the first occurrence of a string beginning with ABC and ending with CDE on the same line, such as ABC123CDE, ABC@CDE, or ABC12 zzzCDE

- 3) The up-arrow (^) is used to find a string only when it occurs at the beginning of a line. If the string appears anywhere else in the line, it will be ignored. The up-arrow has this special meaning only when it appears as the first character of the string. Anywhere else in the string, ^ will have its normal meaning.

```
Example: SEARCH("^ABC",1,0,9999,3)
```

Looks for the first occurrence of ABC only at the beginning of each line. If ABC appears anywhere else in the line, a match will not be found.

- 4) The dollar sign (\$) following the string causes SEARCH to look for the string only at the end of a line. The dollar sign character must appear at the end of the string. When it appears anywhere else in the string, it has its normal meaning.

71-00006

Example: SEARCH("ABC\$",1,0,9999,3)

Looks for the first occurrence of ABC at the end of a line.

If ABC appears anywhere else in the line, it will be ignored.

Sometimes, your string may contain a backslash character as part of the actual text. In this case, you don't want SEARCH to see the backslash as a switch. The solution is to use two sequential backslashes. SEARCH interprets \\ as a single backslash character, not as a switch.