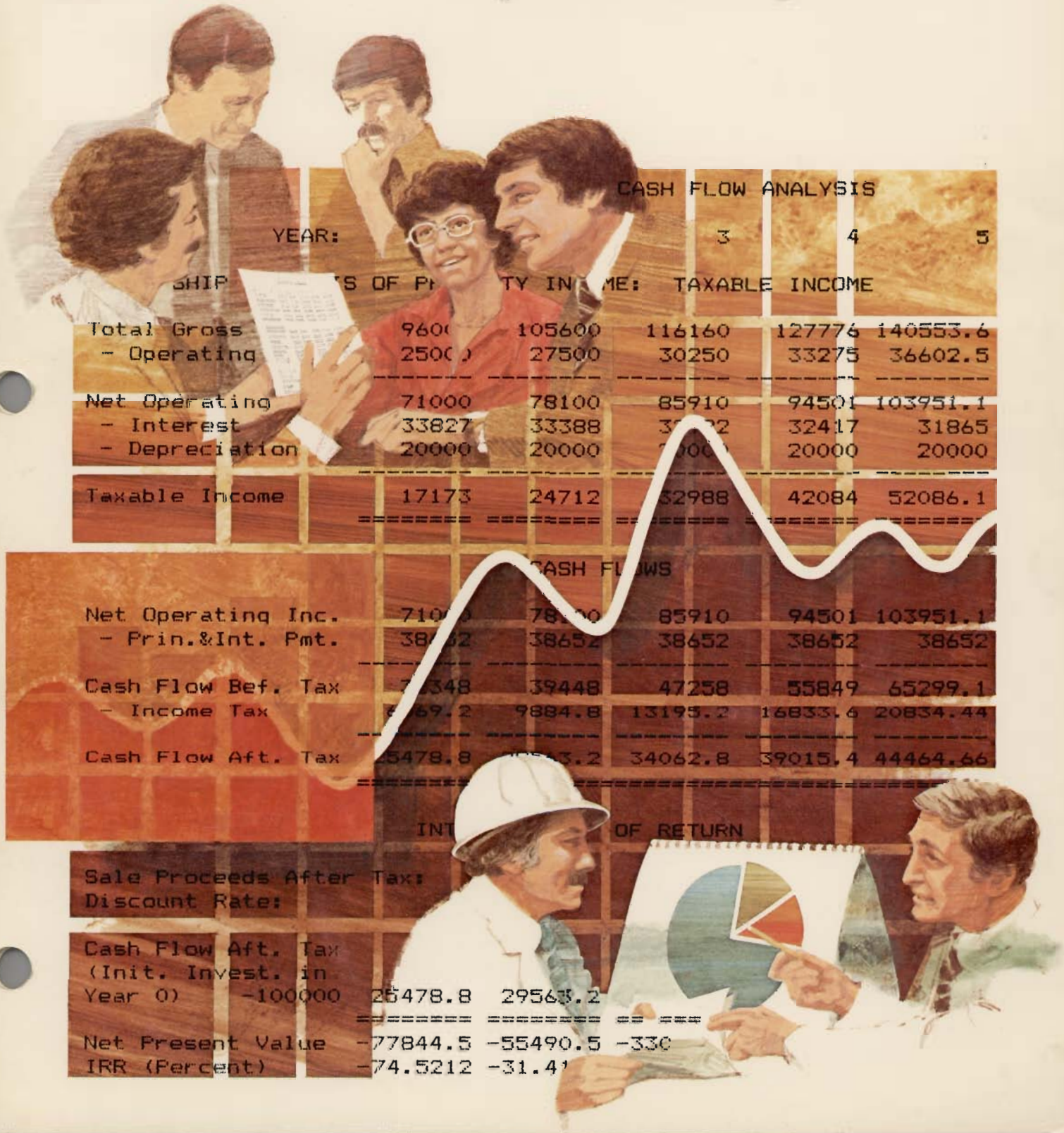
HEWLETT-PACKARD

Introduction to the 86B

Supplement for Local Language Options 001 through 021



HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

HEWLETT PACKARD

Introduction to the HP-86B

Supplement for Local Language Options 001 through 021

May 1983

Reorder Number
00086-90119

Printed in U.S.A.

Contents

How to Use This Supplement . . . 13
Available Memory . . . 14
The Display Pagesize . . . 14
Typing and Displaying Local Language Characters . . . 14
  The Alphanumeric Keys . . . 14
  The Numeric Keypad . . . 15
  System Command, Special Function, and Display Editing Keys . . . 15
Programming With Local Languages . . . 15
  Identification Functions . . . 16
  Switching Letter Cases . . . 16
  Comparison Functions . . . 17
Printing Local Languages . . . 18
  Enabling and Disabling Local Language Printing . . . 18
  Specifying the Printer Type . . . 21
Plotting Local Language Characters . . . 24
Tables 5-a. and 5-b. Character and Key Codes . . . 25
Collating Sequences . . . 28
Printer Character Mapping . . . 29
Local Language Errors . . . 34

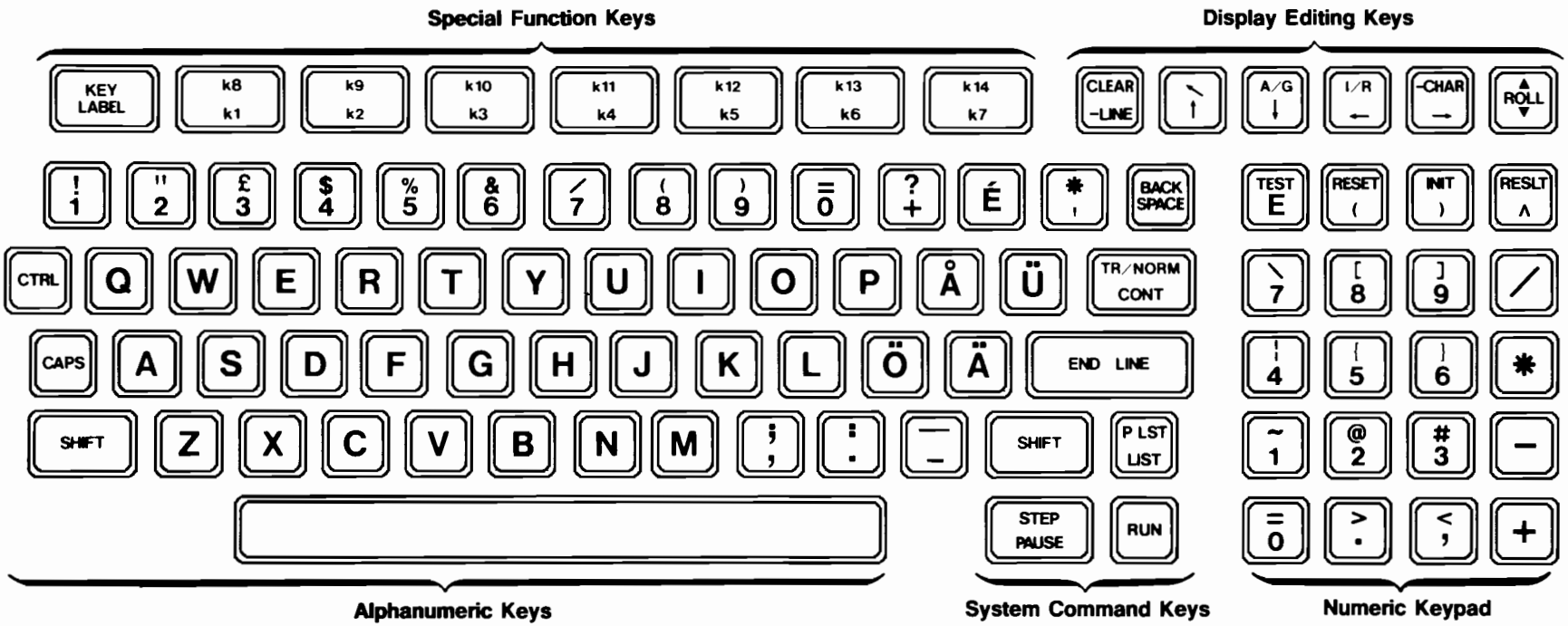


Figure 1-a. Swedish/Finnish Keyboard

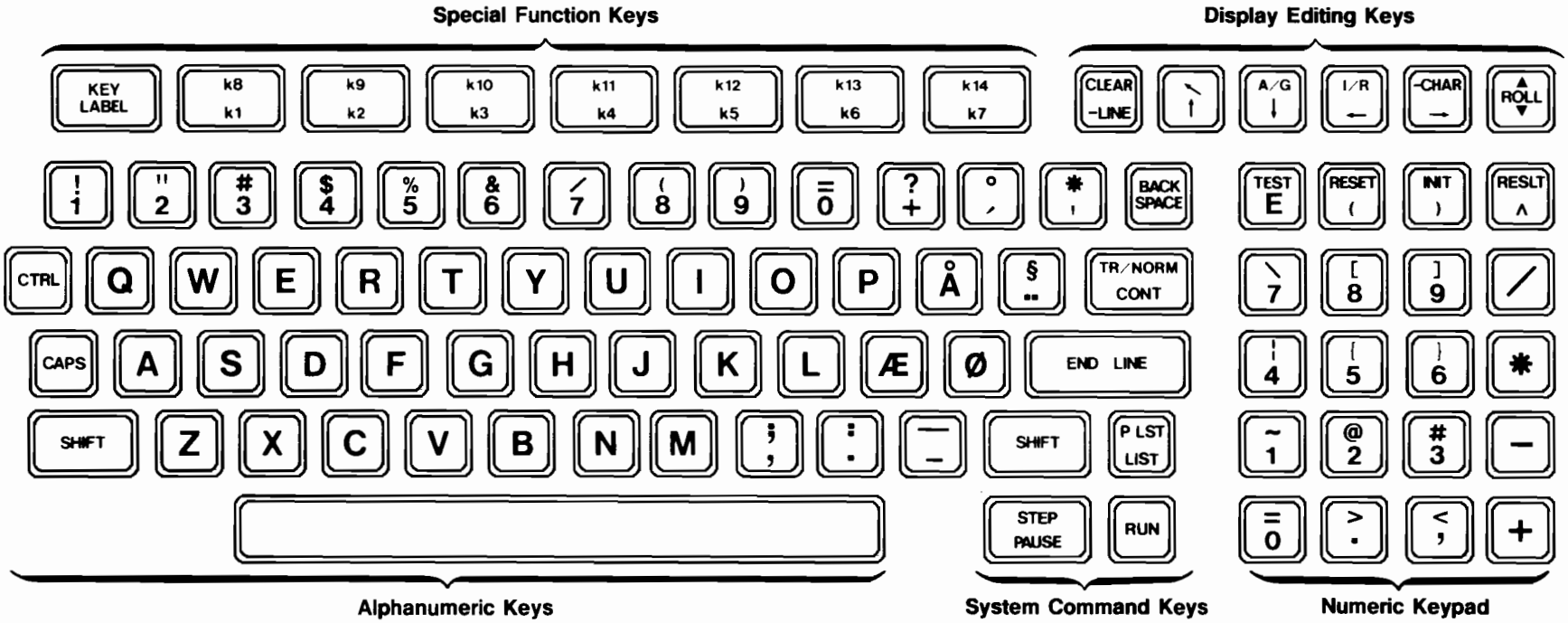


Figure 1-b. Danish/Norwegian Keyboard

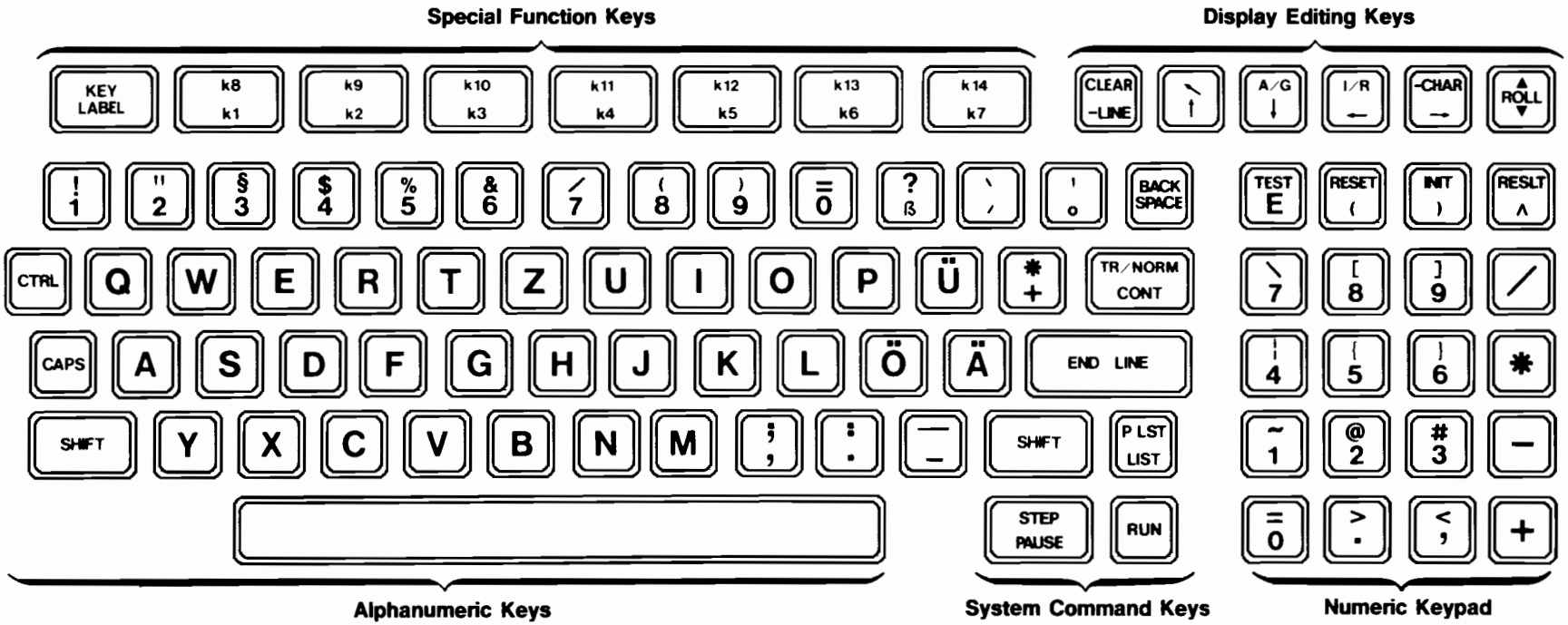


Figure 1-c. German Keyboard

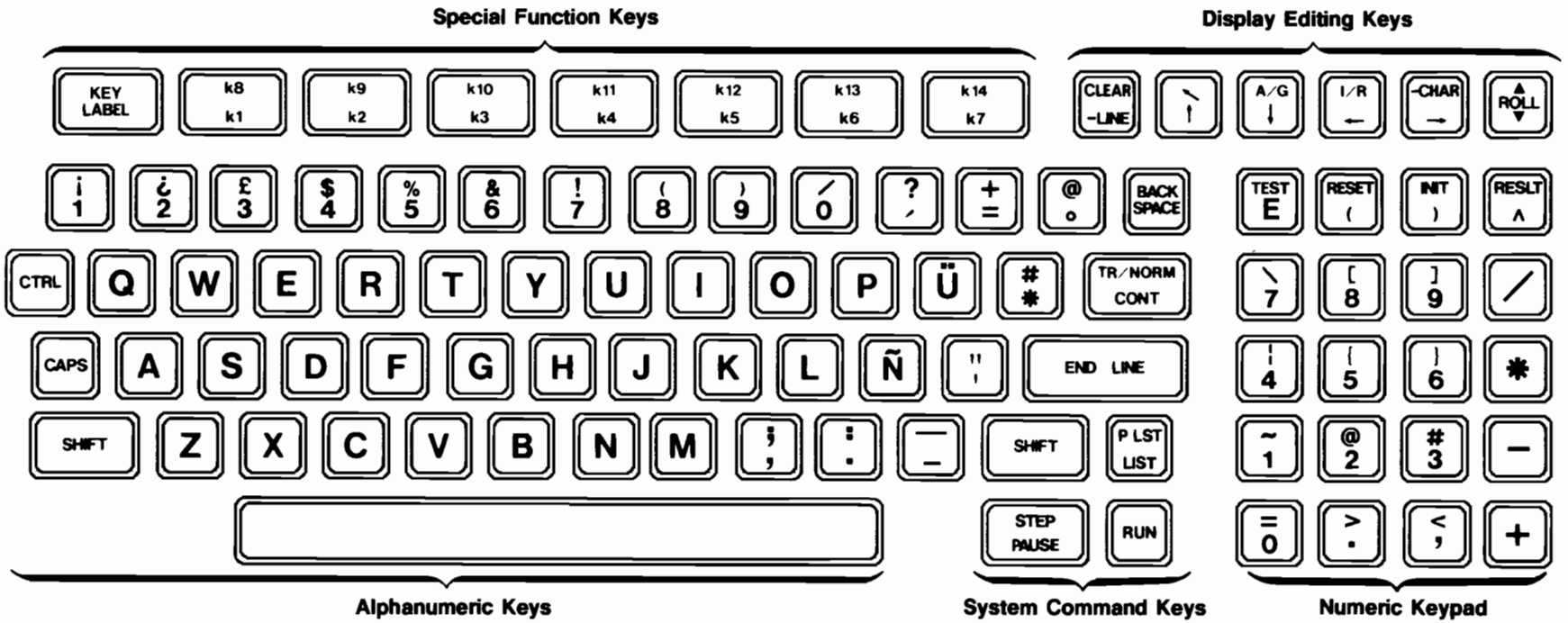


Figure 1-d. Spanish Keyboard

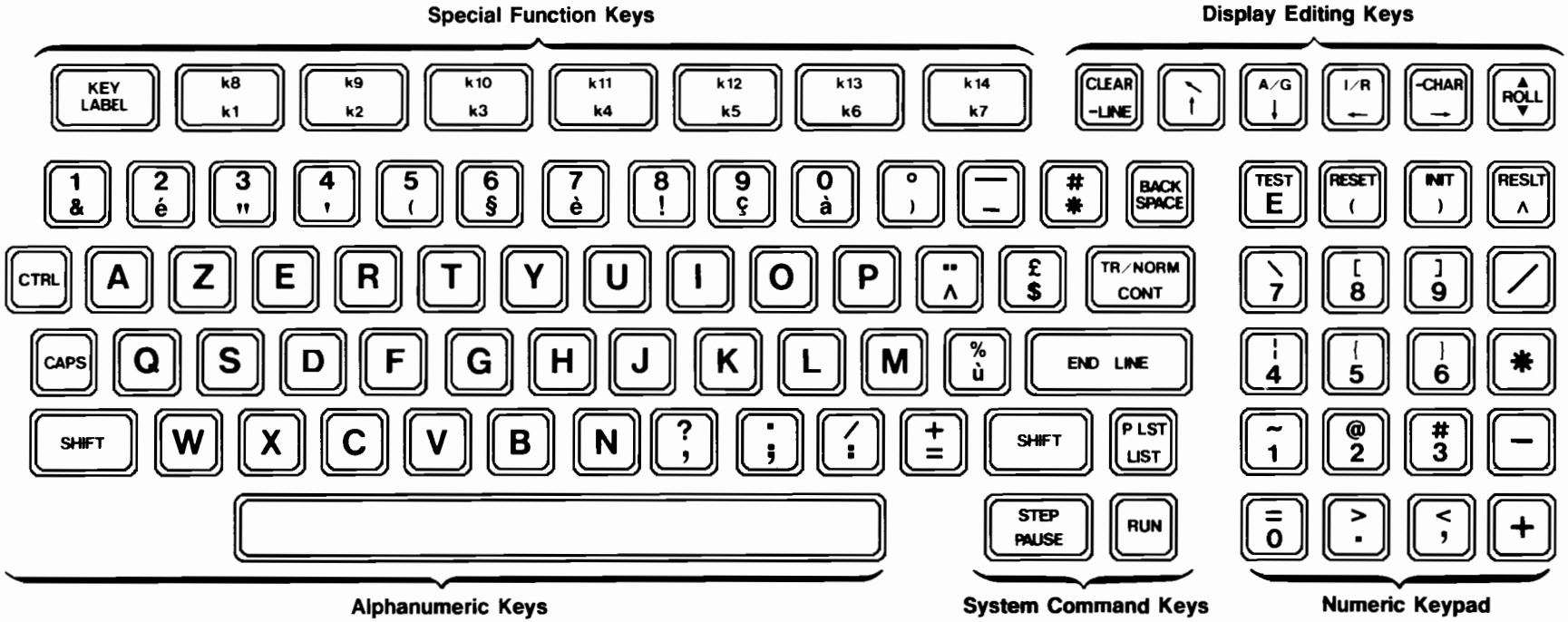


Figure 1-e. French Keyboard

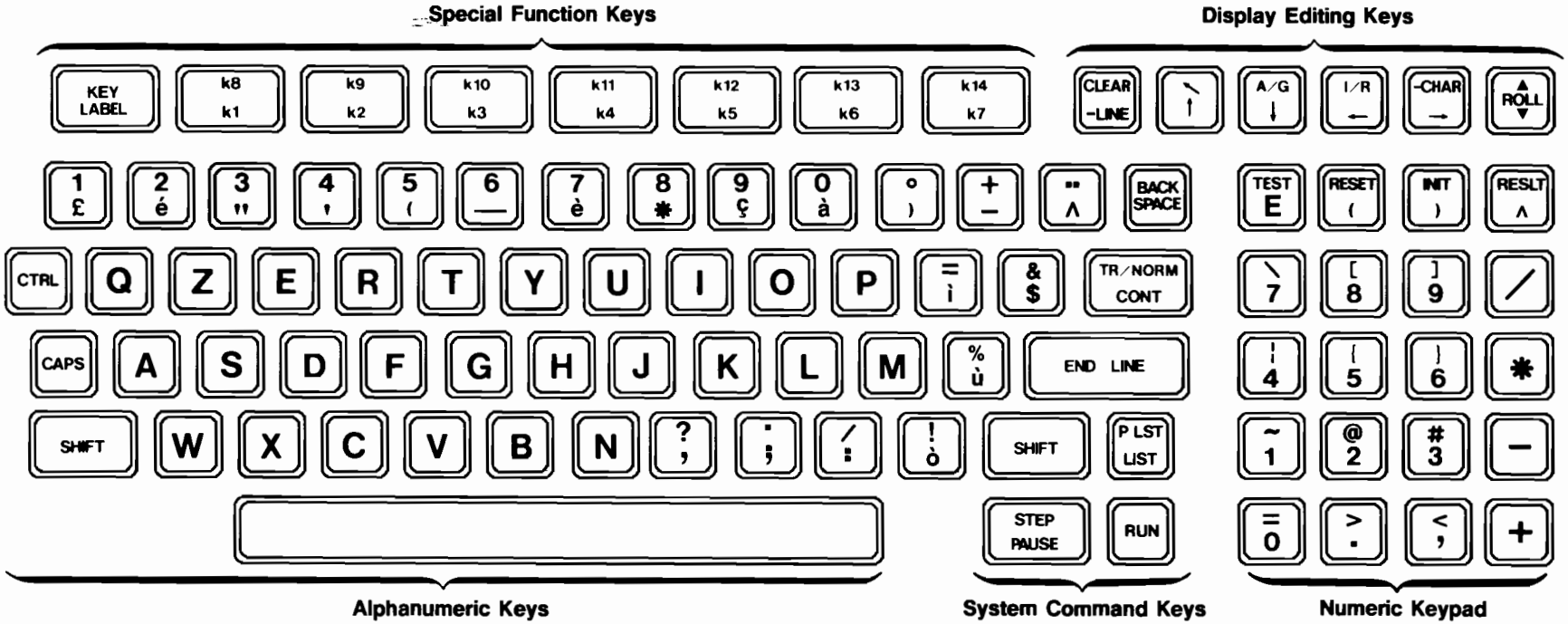


Figure 1-f. Italian Keyboard

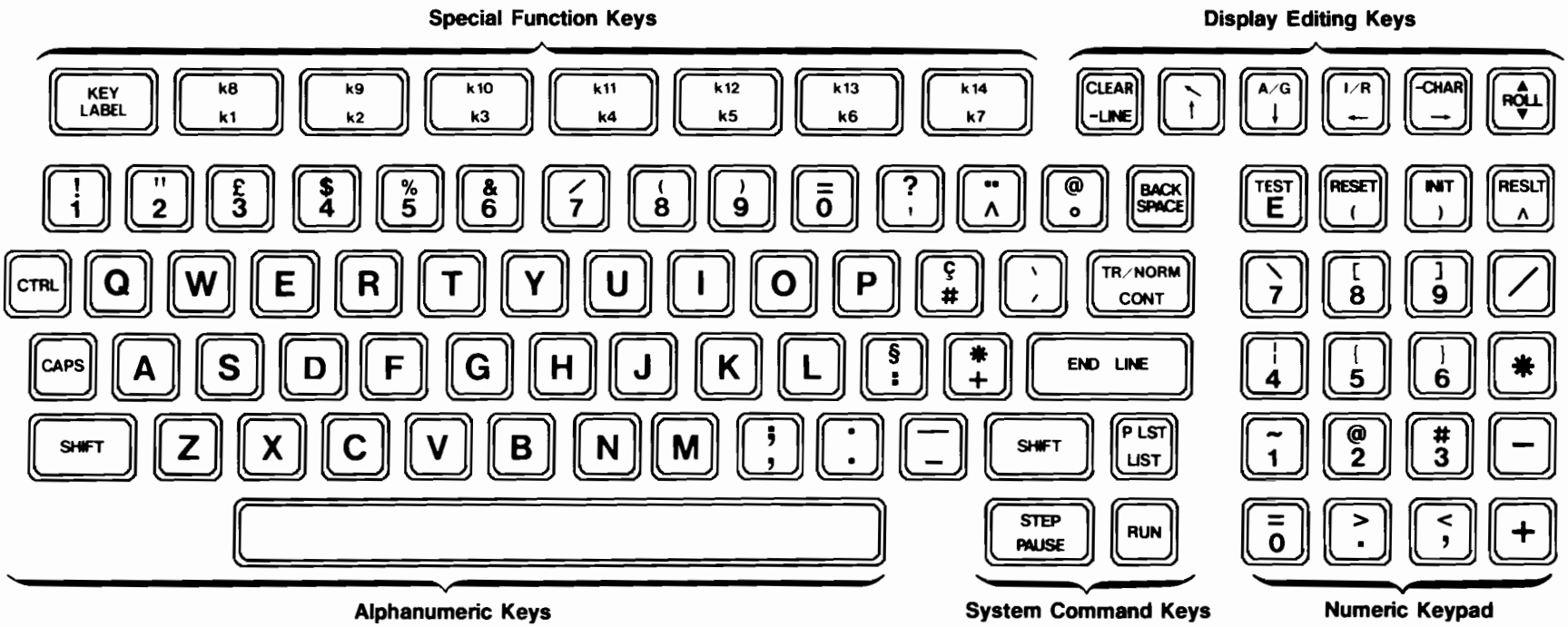


Figure 1-g. Dutch Keyboard

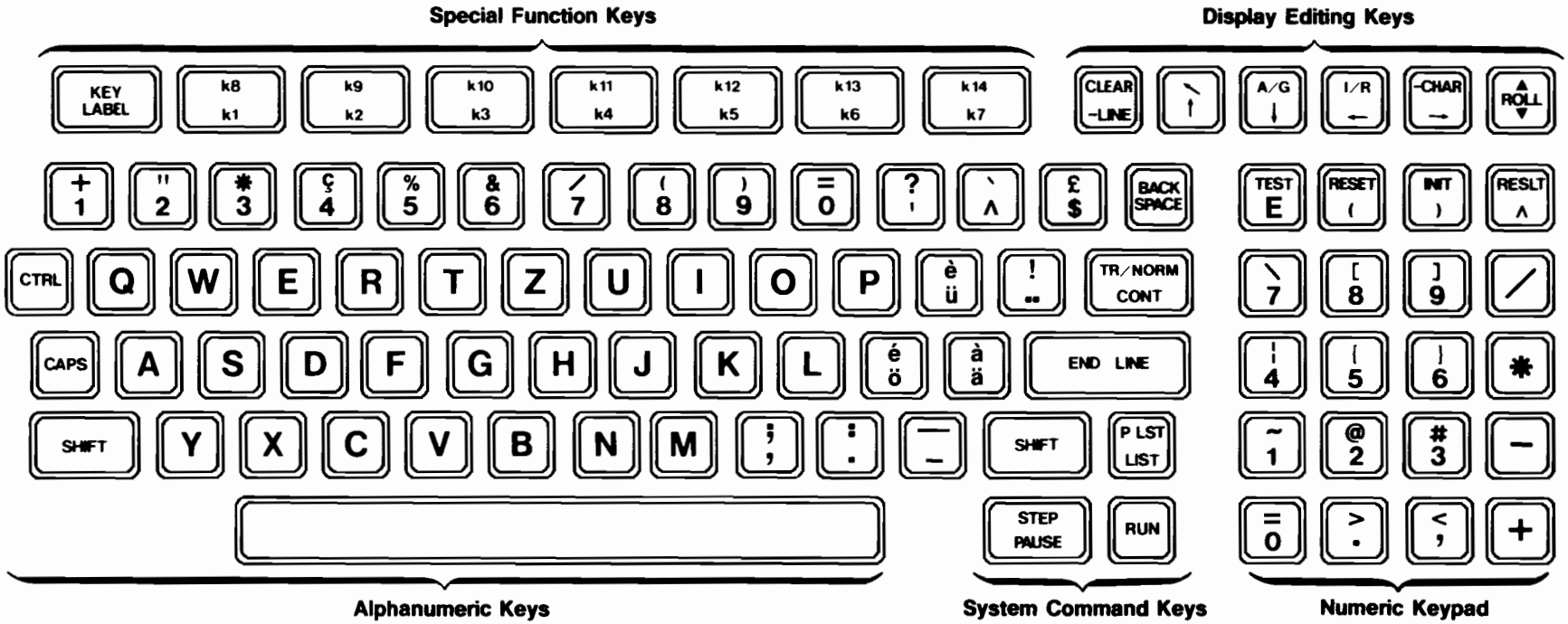


Figure 1-h. Swiss German Keyboard

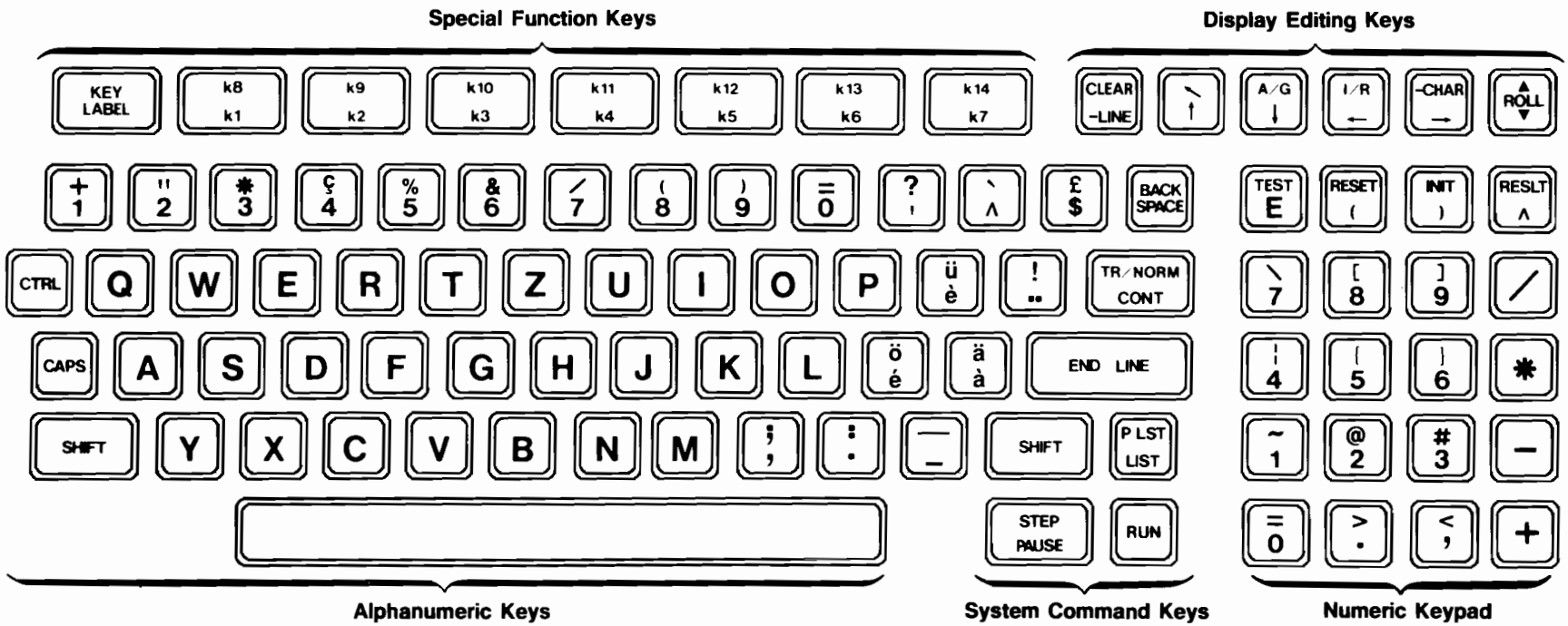


Figure 1-i. Swiss French Keyboard

How to Use This Supplement

This supplement explains the local language features of your HP-86B. Your owner's documentation *(Introduction to the HP-86B* and the *HP-86/87 Operating and BASIC Programming Manual)* describes, in general, the operation of all HP-86B computers and, in particular, the features of the English language model. This supplement describes the differences between the English and local language keyboards, including how to display and print local language characters. It also contains important information on writing BASIC programs using local language characters.

Operation of the English and local language keyboards is nearly identical. Where differences exist, they are explained here. You should refer to this material as you read the descriptions of the computer keyboard in section 4 of the *Introduction to the HP-86B* manual and in section 1 of the *HP-86/87 Operating and BASIC Programming Manual.* The keyboard diagram and table of character and key codes in this supplement replace the analogous information in section 1 and appendix D of the *HP-86/87 Operating and BASIC Programming Manual.*

You should read the material entitled Printing Local Languages before you attempt to print data containing local language characters. Also refer to that section if you have questions about the compatibility of your printer with the HP-86B local language character set.

In addition to its ability to display and print local language characters, the computer provides several statements and functions for manipulating these characters. These statements and functions are provided by the computer's built-in Language ROM and are described here.

Table 1 lists the language options available for the HP-86B. Models with option numbers are equipped with local language keyboards and the built-in Language ROM.

Table 1. HP-86B Options

Product	Language
HP-86B	English
HP-86B Option 001	Swedish/Finnish
HP-86B Option 002	Danish/Norwegian
HP-86B Option 004	German
HP-86B Option 006	Spanish
HP-86B Option 008	French
HP-86B Option 009	Italian
HP-86B Option 010	Dutch
HP-86B Option 020	Swiss German
HP-86B Option 021	Swiss French

Available Memory

Your HP-86B contains 125,452 bytes of built-in random access memory (RAM) available to the user. This figure is somewhat less than the English language model, since the built-in Language ROM requires a certain amount of RAM to implement the local language features.

The Display Pagesize

The *pagesize* of the display is the number of lines of text that can be displayed at a time. At power-on, the local language HP-86B is capable of displaying 24 lines of information. The spacing between lines of text can be changed so that 16 lines are displayed at a time by executing the statement `PAGESIZE 16`. The display is restored to the 24-line format by executing `PAGESIZE 24`.

Typing and Displaying Local Language Characters

Your HP-86B (options 001 through 021) computer is equipped with a local language keyboard that enables you to easily type local language characters. In addition, the computer has been designed to display these characters in both *alpha* and *graph* modes. Section 1 of the *HP-86/87 Operating and BASIC Programming Manual* describes the general features of the English keyboard. This section describes the differences between that keyboard and the local language keyboard provided with your computer.

The local language characters that can be typed and displayed are shown in the tables of character and key codes (tables 5-a or 5-b) on pages 26 and 27. Each of the characters in the table can be displayed, whether or not the character appears on the keyboard. The local language characters are those with decimal codes in the range 0 through 27 and 30 (128 through 155 and 158 for inverse video characters). All other characters are identical between the English and local language computers.

The keyboard layout of your HP-86B local language keyboard has been designed for efficiency and for compatiblity with other local language typing equipment. Please refer to the keyboard diagram (figures 1-a through 1-i) and table of character and key codes for your option computer as you read this material.

The Alphanumeric Keys

There are two types of keys on the alphanumeric keyboard—keys with one character printed on them, and keys with two characters.

Keys with one character on them produce unshifted (the shift key not pressed) uppercase and shifted lowercase letters. Keys with two characters on them display the upper character using the (SHIFT) key.

The (CAPS) key allows you to reverse the function of the alphanumeric keys that have one character printed on them. The (CAPS) key functions as a toggle key—pressing it once causes the keyboard to produce unshifted lowercase and shifted uppercase letters. Pressing (CAPS) again returns the keyboard to its normal function. The (CAPS) key has no effect on keys with two characters on them.

The (CAPS) key always stays level with the other keys; the key does not "lock" into place.

The (CTRL) key, used in conjunction with other alphanumeric keys, allows the computer to display non-keyboard characters with decimal codes lower than 32. In addition, the (CTRL) key provides an alternative way to display the local language characters. Refer to the table of character and key codes for the necessary keystrokes.

The FLIP statement, covered in section 1 of the *HP-86/87 Operating and BASIC Programming Manual*, allows you to use the keyboard like a typewriter to produce unshifted lowercase letters and shifted uppercase letters. The FLIP statement, like the (CAPS) key, affects only alphanumeric keys with one character printed on them.

Example: The £ character can be produced on the French keyboard using either of the following keystrokes:

(SHIFT) (S)

(CTRL) (^)

Accented characters are produced in several ways:

1. If the accented character is on the keyboard, simply press that key. The computer will display the character, complete with its accent mark.
2. If the accented character is not on the keyboard but is in the table of character and key codes, it can be produced in either of two ways. Each way involves two keystrokes.
   - First, press the key for the appropriate accent mark. The accent mark is stored in CRT memory, although it is not displayed; the cursor remains at the same position. Next, press an appropriate vowel key; the computer automatically displays the accented character. Typing a second accent mark immediately after the first replaces the first accent mark. Typing any character after an accent mark whose accented equivalent is not in the table of character and key codes displays the unaccented character.
   - Press the (CTRL) key and the alphanumeric key as shown in the table of character and key codes.

The computer can display only those characters included in the table of character and key codes. However, other characters may be printable by your system. Refer to the discussion of Printing Local Languages on page 18.

The Numeric Keypad

The numeric keypad contains the digits 0 through 9, arithmetic operators, and a variety of other symbols. The alphanumeric numbers and numeric keypad numbers can be used interchangeably. Exponential numbers (for example, 3.4E-10) can be typed using the alphanumeric or numeric keypad (E).

Characters on the upper portion of the key are displayed using the (SHIFT) key.

System Command, Special Function, and Display Editing Keys

Operation of these keys is covered in section 1 of the *HP-86/87 Operating and BASIC Programming Manual.*

Programming With Local Language Characters

All the statements, functions, and commands described in your HP-86B owner's documentation can be used with your local language keyboard. Local language characters can be displayed in *alpha* mode (using the DISP and DISP USING statements) and in *graph* mode (using the LABEL and LABEL USING statements. The characters can be included in the character strings assigned to simple and array string variables. However, you cannot use local language characters in BASIC language variable names, statement labels, or names of user-defined functions.

Identification Functions

The following two functions are used to identify the language option with which your computer is equipped.

The numeric function LANG returns the number of the local language with which the computer is equipped. This number corresponds to the computer option number in table 1. For example, executing LANG on a German HP-86B (option 004) always returns the value 4.

```
LANG
```

The LANG$ function returns the name of the language with which the computer is equipped.

```
LANG$
```

Table 2. String Expressions Returned by LANG$

Option Number	String Expression
001	Svenska/Suomi
002	Dansk/Norge
004	Deutsch
006	Español
008	Français
009	Italiano
010	Nederlands
020	Schwyzerdütsch
021	Romand

Switching Letter Cases

Two functions provide the ability to convert character strings containing local language characters to their uppercase or lowercase equivalents.

The LUPC$ function converts a string containing local language uppercase and lowercase letters to all uppercase. If an accented character has no uppercase equivalent, the character is converted to an unaccented uppercase letter.

```
LUPC$ (string expression)
```

Note: The UPC$ function, covered in section 4 of the *HP-86/87 Operating and BASIC Programming Manual*, has no effect on characters with decimal codes in the range 0 through 31.

The LLWC$ function converts a string containing local language uppercase and lowercase letters to all lowercase. Accented uppercase letters are converted to their accented lowercase equivalents. Unaccented uppercase letters are converted to unaccented lowercase letters.

```
LLWC$ (string expression)
```

Examples:

```
LUPC$("für")
FÜR
```
Enter this expression.
The function returns the accented, uppercase Ü.

```
LUPC$("¿cómo estás?")
¿COMO ESTAS?
```
Enter this expression.
ó and á have no uppercase equivalent.

```
LLWC$("LÄNGE")
länge
```
Enter this expresion.
Accented Ä retains its accent.

```
LLWC$("SYSTEME")
systeme
```
Enter this expression with unaccented E.
The function returns the unaccented lowercase e.

```
LLWC$(LUPC$("¿cómo estás?"))
¿como estas?
```
Enter these nested functions.
The accents are dropped during the conversion to uppercase letters.

Comparison Functions

The language ROM provides a number of functions for comparing characters. The comparisons are based on the *collating sequence* for each language. The difference between using these functions and using the relational operators (<, >, =, etc.) is that this collating sequence does not correspond to the character decimal codes. The collating sequence for each local language is listed in figure 2 on page 28.

The LEX$ function returns a string containing the collating sequence.

```
LEX$
```

The following numeric functions perform string comparisons based on the collating sequence. These functions perform logical (Boolean) evaluation based on comparing the two string arguments, returning 0 (false) or 1 (true). Arguments *string1* and *string2* must be string expressions. *String1* is compared to *string2* character by character until a difference between them is found. A character is evaluated as being less than characters that follow it in the collating sequence and greater than characters that precede it.

Table 1-3. String Comparison Functions

Function Syntax	Meaning
COMPEQ (*string1, string2*)	Equal to.
COMPNE (*string1, string2*)	Not equal to.
COMPGT (*string1, string2*)	Greater than.
COMPLT (*string1, string2*)	Less than.
COMPGE (*string1, string2*)	Greater than or equal to.
COMPLE (*string1, string2*)	Less than or equal to.

Examples: The following examples are based on the German collating sequence.

```
COMPEQ("N","P")
 0
```
Does N equal P?
No.

```
COMPLT("u","ü")
 1
```
Is u less than ü?
Yes.

Printing Local Languages

The printing features of your HP-86B allow you to:

- Direct print output to compatible printers with local language character sets. Since individual printers use different coding for characters, the computer allows you to specify the type character mapping you require using the PRINTER TYPE IS statement.
- Send control codes and escape sequences to the printer during local language printing by using the LANG ON and LANG OFF statements or a specified l-escape (language escape) character.
- Specify customized character mapping using the LPRINT statement.

When the computer is printing in *local language mode*, characters output to the printer are automatically *mapped* to the local language character set of the printer. Mapping involves converting the numeric code used by the computer to identify a character to the code (one or more decimal values) used by the printer. To provide compatibility with a wide range of printers, the computer provides three types of mapping. For example, the character £ (computer decimal code 30) is converted to decimal code 187 (printer type 1), the sequence 14, 59, 15 (printer type 2), or 35 (printer type 3; only for options 004 and 008 through 021). Table 4 on page 22 lists some of the Hewlett-Packard printers compatible with these three mapping schemes. Tables 6-a through 6-e on pages 29 through 33 list the mapping schemes used for printer types 1, 2, and 3. For situations where none of these types is suitable, the LPRINT statement provides the ability to specify a customized mapping scheme.

Enabling and Disabling Local Language Printing

When the computer is printing in *local language mode* (the default condition present at power-on), the computer automatically maps certain characters according to the specified mapping scheme. The characters which are mapped are those with decimal codes in the range 0 through 9, 11, 14 through 27, and 30.

Note: Characters with decimal codes 10, 12, 13, 28, 29, and 31 are sent to the printer unchanged. These codes correspond to the most frequently used printer control codes—line feed (10), form feed (12), carriage return (13)—and to the computer's displayable line drawing characters (28, 29, and 31).

The computer allows you to turn off local language mapping; local language mapping must be turned off in order to send certain control codes and escape sequences to most printers.

The computer provides two ways to enable and disable local language mapping:

- The LANG OFF and LANG ON statements allow you to turn mapping off and on for all subsequent print output.
- The LESCAPE IS statement allows you to specify a character which, when output to the printer, signals that the next character is not to be mapped. Mapping is turned off for only that one character.

Executing LANG OFF affects printing by sending characters to the printer without first mapping the character codes to the local language character set of the printer. When local language mapping is turned off, the computer ceases to do the character mapping specified by the PRINTER TYPE IS and/or LPRINT statements. Characters with decimal codes 0 through 31 can therefore be recognized by the printer as non-printable control characters. For instance, CHR$(27) is recognized as the ASCII ESC (escape) character rather than as the printable character é.

```
LANG OFF
```

The LANG ON statement returns printing to *local language mode.*

```
LANG ON
```

The LANG OFF and LANG ON statements only affect characters output to the printer; displayed characters are unaffected.

> **Note:** When the computer is operating in *local language mode*, characters sent to peripheral devices using the I/O ROM OUTPUT statement are mapped according to the specified printer type. Therefore, a LANG OFF statement must be executed before control codes can be sent using OUTPUT.

LANG OFF and LANG ON are convenient when you are sending a series of control codes to the printer or when an escape sequence includes more than one character with decimal code less than 31. It is also necessary to turn off character mapping when you are sending bytes of graphics data to printers.

Example: The following program sends seven bytes of graphics data (variable TRIANGLE$) to the HP 82905B Printer. Statement 40 sends the escape sequence that switches the printer to graphics mode and then sends the seven bytes of data.

```
10 TRIANGLE$=CHR$(1) & CHR$(3) & CHR$(7) & CHR$(15) & CHR$(7) & CHR$(3) & CHR$(1
) !         Bytes of data to draw a triangle.
20 PRINT "Triangle -- ";
30 LANG OFF
40 PRINT CHR$(27) &"*b7G" & TRIANGLE$;
50 LANG ON
60 PRINT " -- Triangle"
70 END
```

Running the program generates the following printer output:

```
Triangle -- é -- Triangle
```

The ISLANG numeric function provides the ability to determine which mode the computer is currently using for printing.

```
ISLANG
```

The function returns the value 1 (true) when the computer is printing in *local language mode* and the value 0 (false) when mapping has been turned off.

The *l-escape* character provides an additional way to send control codes to the printer. The default l-escape character at power-on is defined as CHR$(127). (This character, ■, can be displayed by pressing (SHIFT) (+)). When the l-escape character is included in the character string to be printed, the computer turns off local language mapping for the next character; mapping is then turned back on.

> **Note:** The *l-escape* character is not equivalent to the *escape* character. The l-escape character is used in a print string to signal the *computer* to turn off mapping for the next character. The escape character is used in a print string to signal the *printer* that the next character is an instruction.

Example: The following PRINT statement sends the ASCII escape (ESC, CHR$(27)) character to the printer without mapping it to decimal code 197 (é) for PRINTER TYPE 1.

```
PRINT CHR$(127) & CHR$(27) & "18D"
```

Sets vertical line spacing to 8 lines/inch on the HP 82905B Printer.

Example: The following program prints the character é (computer decimal code 27). The program then sets the HP 82905B Printer to expanded type and prints an expanded é. Finally, the program returns the printer to normal type.

```
10 PRINT CHR$(27) !                        Prints character é.
20 PRINT CHR$(13) !                        Sends end-of-line sequence to
                                           printer. This character isn't mapped.
30 PRINT CHR$(127) & CHR$(27)&"&k1S" ! Sends escape sequence for
                                           expanded type.
40 PRINT CHR$(27) !                        Prints character é.
50 PRINT CHR$(127) & CHR$(27)&"&k0S" ! Sends escape sequence for
                                           normal type.
60 END
```

The LESCAPE IS statement allows you to change the l-escape character from the default value of CHR$(127) to any character with decimal code in the range 0 through 255.

```
LESCAPE IS decimal code
           string expression
```

You can specify only one decimal code. However, the string expression can evaluate to a string of any length. If the length of the string expression is greater than 1, the computer returns a warning and ignores all characters after the first. There can be only one l-escape character at a time; executing a second LESCAPE IS statement replaces the previously declared l-escape character with the new one.

> **Note:** Many printing operations involve sending line feeds (CHR$(10)), carriage returns (CHR$(13)), and form feeds (CHR$(12)) to the printer. Keep this in mind when you are selecting an l-escape character.

Example:

```
LESCAPE IS "@"

LESCAPE IS 64
```

Both statements change the l-escape character to @.

The l-escape character is treated as a non-printable character that signals the computer to temporarily turn off mapping. However, l-escape characters that are ordinarily printable can be printed by including them twice in the print string. For example, one (l-escape) character @ can be printed by executing PRINT "@@".

The LESCAPE$ function returns a character string of length 1 containing the current l-escape character.

```
LESCAPE$
```

Example: The following program segment sends an escape sequence to the HP 82905B Printer. CHR$(27) is the ASCII escape (ESC) character.

```
10 LESCAPE IS "*" !                 Sets l-escape character to *.
20 C$=LESCAPE$ !                    Variable assignment.
30 PRINT C$ & CHR$(27) & "&l9D" !   Sets vertical spacing on HP 82905B
                                    Printer to 9 lines per inch.
.
.
.
```

Specifying the Printer Type

Since different printers use different codes for characters, it is necessary to specify the way the computer maps local language characters to the printer. The PRINTER TYPE IS statement allows you to select from three mapping schemes, or to specify your own mapping scheme.

```
PRINTER TYPE IS numeric expression
```

The numeric expresion must evaluate to a number in the range 1 through 4. Fractional numbers are rounded to the nearest integer. If the numeric expression evaluates outside the range 1 through 4, the computer returns Error 110 : UNKNOWN PRINTER TYPE and uses the previously declared (or default) printer type.

The following table lists the printer types and some of the Hewlett-Packard printers using those mappings. The default type at power-on is type 1.

Table 4. Printer Types

Type	Character Set	Printers
1	8-bit Extended Roman	HP 82905B, HP 2631, HP 9876*, HP 2671
2	7-bit Extended Roman	HP 9876*, HP 82905B*
3	ISO Substitution	HP 260x-Series
4	User-defined	Non-standard character sets

No local language mapping occurs using printer type 4 other than character substitutions you specify using the LPRINT statement.

Tables 6-a through 6-e on pages 29 through 33 show the mapping schemes used for printer types 1, 2, and 3. If your printer is not listed in table 4, compare the printer character set to the characters and decimal codes in these tables. If none of the printer types is compatible, you must use printer type 4 and the LPRINT statement to specify customized mapping for your printer.

Note: As indicated in tables 6-a through 6-e, printer types 2 and 3 frequently involve mapping one character to a series of characters, only one of which is actually printed. Although they are not printed, these extra characters are counted in the computer's computation of line length. Therefore, when local language characters are being printed, the printer may start a new line (perform a carriage return/line feed) before the declared line length (or default of 80) has been printed.

The PRINTER TYPE numeric function returns and integer corresponding to the current printer type.

```
PRINTER TYPE
```

The LPRINT statement allows you to specify character replacements for any of the HP-86B characters.

```
LPRINT string1 AS { string2 / decimal value } [, decimal value...]
```

String1 and *string2* are string expressions that evaluate to character strings of any length. The first character in *string1* is mapped to the character(s) specified after the keyword AS. All other characters in *string1* are ignored. For example, the statement LPRINT "%#" AS 97,8,96 uses only the % character. The computer returns WARNING 100 : ONLY FIRST CHAR USED to inform you that the additional characters have been dropped.

* The HP 9876 and HP 82905B printers support both 8-bit Extended Roman and 7-bit Extended Roman character coding.

If more than one replacement character is specified, the characters are printed as a continuous string. The replacement characters are specified in one of two ways:

- Each replacement character is listed using the printer's decimal code for that character. (Refer to your printer documentation for the character codes for the printer's character set.) This method is necessary when the replacement character is not in the HP-86B character set or when the printer's decimal code for the replacement character is different from the computer's character code.
- If each replacement character has the same character code in the computer and the printer, the replacement characters can be included in a string expression (*string2*)—either a quoted string or a string variable assignment.

Examples:

```
PRINTER TYPE IS 1
Character=187
LPRINT "$" AS Character
```

Maps $ to £ for HP 82905B Printer.

```
B$="ß"
LPRINT B$ AS 115, 115
LPRINT "ß" AS "ss"
```

Maps ß to ss.
Maps ß to ss. Assumes that the printer decimal code for s is 115, the same as the computer's.

Once LPRINT mapping has been specified for a character, that mapping remains in effect until a new LPRINT statement is executed for that character or until the power is turned off. LPRINT mapping is not affected by (RESET). LPRINT mapping can be turned off using the LANG OFF statement; mapping resumes when the LANG ON statement is executed.

The LPRINT statement can also be used with printer types 1, 2, and 3. LPRINT mapping takes precedence over mapping implemented by the printer type. A character mapped using LPRINT can be restored its normal PRINTER TYPE IS mapping by executing an LPRINT statement with *string2* equal to the null string.

Example: PRINTER TYPE IS 1 maps the computer character § (decimal code 0) to decimal code 189. However, executing the LPRINT statement in the following program causes the computer § character to be printed as the character assigned decimal code 186 in the printer character set. For the HP 82905B Printer, this is the character ¤.

```
10 PRINTER TYPE IS 1 !   For documentation purposes-type 1 is the
                         default printer type.
20 LPRINT "§" AS 186 !
30 PRINT "§";
40 LPRINT "§" AS "" !    Restores mapping to decimal code 189.
50 PRINT "§"
60 END
```

(RUN)

```
¤§
```

The total number of characters that can be mapped using LPRINT varies with the type of replacement. When you specify single character replacements (for example, LPRINT "W" AS "X"), you can replace a maximum of 32 characters (in other words, 32 LPRINT statements). The number of LPRINT statements possible is reduced when you are mapping characters to more than one replacement (for example, LPRINT "W" AS "XYZ").

Examples: The following LPRINT statements map one computer character to a sequence of printer characters.

```
LPRINT "$" AS "Canadian Dollars"
LPRINT "£" AS "Pounds Sterling"
```

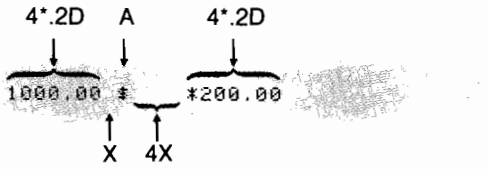
If you attempt to execute an LPRINT statement that would cause LPRINT mapping to exceed the total number of characters that can be mapped, the computer returns Error 109 : TOO MANY LPRINTS. All previously declared LPRINT mappings remain in effect.

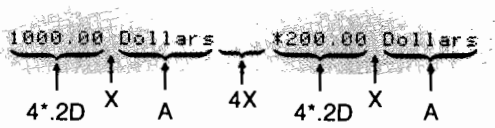
> **Note:** If you are using formatted printing (with the PRINT USING statement) and if you are using LPRINT to map a character to more than one character, you should format only one character position in the output for the print string character, regardless of the number of characters to which it is mapped. The entire string of characters will be printed, and any other output will be offset to the right by the number of additional characters.

Example: The image specifier A in statement 20 is used to format the string "Dollars".

```
10 PRINT USING "2(4*.2D,X,A,4X)"; 1000, "$", 200, "$"
20 PRINT
30 LPRINT "$" AS "Dollars"
40 PRINT USING "2(4*.2D,X,A,4X)"; 1000, "$", 200, "$"
50 END
```

RUN





Plotting Local Language Characters

If your system includes a plotter ROM, local language characters can be output to peripheral plotters using the LABEL and LABEL USING statements. Since these characters must be mapped to match the appropriate plotter character set, the plotter must be treated somewhat like a printer while local language characters are being sent to it. This is accomplished by executing a PRINTER IS statement that designates the plotter as the system printer. Then, the PRINTER TYPE IS statement and/or a series of LPRINT statements are used to specify mapping for label output. After the label is plotted, the original printer address and mapping must be restored before data can be printed on the system printer.

Example: The following program contains a subroutine that plots a German label (variable TEXT$) up to 40 characters long on an HP 7470A system plotter located at address 705. The subroutine designates the plotter as the system printer, specifies a set of LPRINT mappings, and then outputs the mapped label to the plotter. Finially, the subroutine restores the prior mapping scheme and printer address before passing control back to the main body of the program.

```
10 PRINTER IS 701 !          Designates system printer.
20 PLOTTER IS 705 !          Designates system plotter.
30 SCALE 0,100,0,100 !       Scales plotting boundaries.
40 CSIZE 8 !                 Sets character height.
50 DIM TEXT$[40],LANGLABEL$[40] ! Label may be up to 40 characters long.
60 DISP "Enter label, x-coordinate, y-coordinate"
70 INPUT TEXT$,X,Y
80 MOVE X,Y !                Moves pen to specified coordinates.
90 LANGLABEL$=TEXT$ !        LANGLABEL$ will be mapped in subroutine.
100 GOSUB PLOTLABEL
110 STOP
120 PLOTLABEL:
130   PRINTER IS 705 !       Provides for mapping label output.
140   PRINT "CS2" !        Designates character set #2 as the standard
                           character set.
150   LPRINT " " AS 65,123 ! Specifies plotter mapping.
160   LPRINT " " AS 79,123
170   LPRINT " " AS 85,123
180   LPRINT " " AS 97,125
190   LPRINT " " AS 111,125
200   LPRINT " " AS 117,125
210   LPRINT " " AS 32,124
220   LABEL LANGLABEL$ !     Mapped label is sent to plotter
230   LPRINT " " AS "" !     Restores previous printer type mapping.
240   LPRINT " " AS ""
250   LPRINT " " AS ""
260   LPRINT " " AS ""
270   LPRINT " " AS ""
280   LPRINT " " AS ""
290   LPRINT " " AS "
300   PRINT "CS0" ! Designates character set #0 as the standard.
310   PRINTER IS 701 ! Restores system printer address.
290 RETURN
300 END
```

It is not necessary to use LPRINT mapping if one of the plotter character sets matches printer types 1, 2, or 3. For example, a plotter that supports an ISO substitution character set requires a labeling subroutine that designates the plotter as the peripheral printer (for example, PRINTER IS 705), establishes the appropriate printer type (PRINTER TYPE IS 3), and designates the ISO substitution character set as the standard character set.

Tables 5-a and 5-b. Character and Key Codes

These tables replaces the table in appendix D of the *HP-86/87 Operating and BASIC Programming Manual.* A numeric code is attached to each character and/or key. Non-keyboard characters are displayed by using the (CTRL) key (indicated by superscript **c**), accents followed by vowels, or the (SHIFT) key (indicated by superscript **s**). Each character with a decimal code in the range 0 through 127 has a complementary inverse video character with a decimal code in the range 128 through 255. Inverse video characters are displayed using the CHR$ function. For instance, CHR$(74+128) is inverse video J. Inverse video characters are also displayed by certain keys when a program is paused at an INPUT statement. Characters with decimal codes 0 through 31 are recognized by certain peripheral devices (including printers, when local language mapping is turned off) as instructions. For convenience, ASCII convention assigns a mnemonic to each of these control characters.

Table 5-a. Character and Key Codes for:
Swedish/Finnish (option 001)
Danish/Norwegian (option 002)
Spanish (option 006)

	EQUIVALENT FORMS			EQUIVALENT FORMS			EQUIVALENT FORMS			EQUIVALENT FORMS		
Mnemonic	Char.	Binary	Dec	Char.	Binary	Dec	Char.	Binary	Dec	Char.	Binary	Dec
NUL	@^{C}	00000000	0	SPACE	00100000	32	@	01000000	64	`	01100000	96
SOH	A^{C}	00000001	1	!	00100001	33	A	01000001	65	a	01100001	97
STX	B^{C}	00000010	2	"	00100010	34	B	01000010	66	b	01100010	98
ETX	C^{C}	00000011	3	#	00100011	35	C	01000011	67	c	01100011	99
EOT	D^{C}	00000100	4	$	00100100	36	D	01000100	68	d	01100100	100
ENQ	E^{C}	00000101	5	%	00100101	37	E	01000101	69	e	01100101	101
ACK	F^{C}	00000110	6	&	00100110	38	F	01000110	70	f	01100110	102
BEL	G^{C}	00000111	7	'	00100111	39	G	01000111	71	g	01100111	103
BS	H^{C}	00001000	8	(	00101000	40	H	01001000	72	h	01101000	104
HT	I^{C}	00001001	9	)	00101001	41	I	01001001	73	i	01101001	105
LF	J^{C}	00001010	10	*	00101010	42	J	01001010	74	j	01101010	106
VT	K^{C}	00001011	11	+	00101011	43	K	01001011	75	k	01101011	107
FF	L^{C}	00001100	12	,	00101100	44	L	01001100	76	l	01101100	108
CR	M^{C}	00001101	13	-	00101101	45	M	01001101	77	m	01101101	109
SO	N^{C}	00001110	14	.	00101110	46	N	01001110	78	n	01101110	110
SI	O^{C}	00001111	15	/	00101111	47	O	01001111	79	o	01101111	111
DLE	P^{C}	00010000	16	0	00110000	48	P	01010000	80	p	01110000	112
DC1	Q^{C}	00010001	17	1	00110001	49	Q	01010001	81	q	01110001	113
DC2	R^{C}	00010010	18	2	00110010	50	R	01010010	82	r	01110010	114
DC3	S^{C}	00010011	19	3	00110011	51	S	01010011	83	s	01110011	115
DC4	T^{C}	00010100	20	4	00110100	52	T	01010100	84	t	01110100	116
NAK	U^{C}	00010101	21	5	00110101	53	U	01010101	85	u	01110101	117
SYN	V^{C}	00010110	22	6	00110110	54	V	01010110	86	v	01110110	118
ETB	W^{C}	00010111	23	7	00110111	55	W	01010111	87	w	01110111	119
CAN	X^{C}	00011000	24	8	00111000	56	X	01011000	88	x	01111000	120
EM	Y^{C}	00011001	25	9	00111001	57	Y	01011001	89	y	01111001	121
SUB	Z^{C}	00011010	26	:	00111010	58	Z	01011010	90	z	01111010	122
ESC	[^{C}	00011011	27	;	00111011	59	[	01011011	91	{	01111011	123
FS	\$^{C}$	00011100	28	<	00111100	60	\	01011100	92	|	01111100	124
GS	]^{C}	00011101	29	=	00111101	61	]	01011101	93	}	01111101	125
RS	^^{C}	00011110	30	>	00111110	62	^	01011110	94	~ *S	01111110	126
US	_^{C}	00011111	31	?	00111111	63	_	01011111	95	+^{S}	01111111	127

Table 5-b. Character and Key Codes for:

German (option 004)
French (option 008)
Italian (option 009)
Dutch (option 010)
Swiss German (option 020)
Swiss French (option 021)

		EQUIVALENT FORMS			EQUIVALENT FORMS			EQUIVALENT FORMS			EQUIVALENT FORMS	
Mnemonic	Char.	Binary	Dec	Char.	Binary	Dec	Char.	Binary	Dec	Char.	Binary	Dec
NUL	@^{C}	00000000	0	SPACE	00100000	32	@	01000000	64	`	01100000	96
SOH	A^{C}	00000001	1	!	00100001	33	A	01000001	65	a	01100001	97
STX	B^{C}	00000010	2	"	00100010	34	B	01000010	66	b	01100010	98
ETX	C^{C}	00000011	3	#	00100011	35	C	01000011	67	c	01100011	99
EOT	D^{C}	00000100	4	$	00100100	36	D	01000100	68	d	01100100	100
ENQ	E^{C}	00000101	5	%	00100101	37	E	01000101	69	e	01100101	101
ACK	F^{C}	00000110	6	&	00100110	38	F	01000110	70	f	01100110	102
BEL	G^{C}	00000111	7	'	00100111	39	G	01000111	71	g	01100111	103
BS	H^{C}	00001000	8	(	00101000	40	H	01001000	72	h	01101000	104
HT	I^{C}	00001001	9	)	00101001	41	I	01001001	73	i	01101001	105
LF	J^{C}	00001010	10	*	00101010	42	J	01001010	74	j	01101010	106
VT	K^{C}	00001011	11	+	00101011	43	K	01001011	75	k	01101011	107
FF	L^{C}	00001100	12	,	00101100	44	L	01001100	76	l	01101100	108
CR	M^{C}	00001101	13	-	00101101	45	M	01001101	77	m	01101101	109
SO	N^{C}	00001110	14	.	00101110	46	N	01001110	78	n	01101110	110
SI	O^{C}	00001111	15	/	00101111	47	O	01001111	79	o	01101111	111
DLE	P^{C}	00010000	16	0	00110000	48	P	01010000	80	p	01110000	112
DC1	Q^{C}	00010001	17	1	00110001	49	Q	01010001	81	q	01110001	113
DC2	R^{C}	00010010	18	2	00110010	50	R	01010010	82	r	01110010	114
DC3	S^{C}	00010011	19	3	00110011	51	S	01010011	83	s	01110011	115
DC4	T^{C}	00010100	20	4	00110100	52	T	01010100	84	t	01110100	116
NAK	U^{C}	00010101	21	5	00110101	53	U	01010101	85	u	01110101	117
SYN	V^{C}	00010110	22	6	00110110	54	V	01010110	86	v	01110110	118
ETB	W^{C}	00010111	23	7	00110111	55	W	01010111	87	w	01110111	119
CAN	X^{C}	00011000	24	8	00111000	56	X	01011000	88	x	01111000	120
EM	Y^{C}	00011001	25	9	00111001	57	Y	01011001	89	y	01111001	121
SUB	Z^{C}	00011010	26	:	00111010	58	Z	01011010	90	z	01111010	122
ESC	[^{C}	00011011	27	;	00111011	59	[	01011011	91	{	01111011	123
FS	\$^{C}$	00011100	28	<	00111100	60	\	01011100	92	|	01111100	124
GS	]^{C}	00011101	29	=	00111101	61	]	01011101	93	}	01111101	125
RS	^^{C}	00011110	30	>	00111110	62	^	01011110	94	~ *^{S}	01111110	126
US	_^{C}	00011111	31	?	00111111	63	_	01011111	95	+^{S}	01111111	127

Collating Sequences

Figure 2. Local Language Collating Sequences

The collating sequence is used by the character comparison functions (COMPEQ, COMPNE, COMPGT, COMPLT, COMPGE, COMPLE). A character evaluates as less than characters that follow it and greater than characters that precede it. The first character in the sequence is the SPACE character, decimal code 32.

Swedish/Finnish Collating Sequence

```
 !i"#$%&'()*+,-./0123456789:;<=>?¿@AÆBCDEÉFGHIJKLMNÑOPQRSTUÜVWXYZ
ÄÄÖØ[\]^_`aæåbcdeéfghiíjklmnñoópqrsßtuüúvwxyzääöø°§£{¦}~▒
```

Danish/Norwegian Collating Sequence

```
 !i"#$%&'()*+,-./0123456789:;<=>?¿@AÄBCDEÉFGHIJKLMNÑOÖPQRSTUÜVWXYZ
ÆØÅ[\]^_`aäåbcdeéfghiíjklmnñoóöpqrsßtuüúvwxyzæøå°§£{¦}~▒
```

German Collating Sequence

```
 !"#$%&'()*+,-./0123456789:;<=>?@AÄBCÇDEFGHIJKLMNOÖPQRSTUÜVWXYZ
[\]^_`aäàâbcdeéèêëfghiìîïjklmnoöòôpqrsßtuüùûvwxyz°§£{¦}~▒
```

Spanish Collating Sequence

```
 !i"#$%&'()*+,-./0123456789:;<=>?¿@AÄÁÆBCDEÉFGHIJKLMNÑOÖØPQRSTUÜVW
XYZ[\]^_`aáäæbcdeéfghiíjklmnñoóöøpqrsßtuúüvwxyz°§£{¦}~▒
```

French Collating Sequence

```
 !"#$%&'()*+,-./0123456789:;<=>?@AÄBCÇDEFGHIJKLMNOÖPQRSTUÜVWXYZ
[\]^_`aäàâbcdeéèêëfghiîïìjklmnoöôòpqrsßtuüûùvwxyz§°£{¦}~▒
```

Italian Collating Sequence

```
 !"#$%&'()*+,-./0123456789:;<=>?@AÄBCÇDEFGHIJKLMNOÖPQRSTUÜVWXYZ
[\]^_`aäàâbcdeéèêëfghiîïìjklmnoöôòpqrsßtuüûùvwxyz§°£{¦}~▒
```

Dutch Collating Sequence

```
 !"#$%&'()*+,-./0123456789:;<=>?@AÄBCÇDEFGHIJKLMNOÖPQRSTUÜVWXYZ
[\]^_`aäàâbcdeéèêëfghiîïjklmnoöôòpqrsßtuüûùvwxyz°§£{¦}~▒
```

Swiss German Collating Sequence

```
 !"#$%&'()*+,-./0123456789:;<=>?@AÄBCÇDEFGHIJKLMNOÖPQRSTUÜVWXYZ
[\]^_`aäàâbcdeéèêëfghiîïjklmnoöôòpqrsßtuüûùvwxyz°§£{¦}~▒
```

Swiss French Collating Sequence

```
 !"#$%&'()*+,-./0123456789:;<=>?@AÄBCÇDEFGHIJKLMNOÖPQRSTUÜVWXYZ
[\]^_`aäàâbcdeéèêëfghiîïìjklmnoöôòpqrsßtuüûùvwxyz§°£{¦}~▒
```

Printer Character Mapping

Table 6-a. Character Mapping for Swedish/Finnish (option 001)

Character	Computer Decimal Code	Printer Decimal Code Type 1	Type 2*	Type 3
§	0	189	61	—
¿	1	185	57	—
°	2	179	51	—
Ñ	3	182	54	—
ñ	4	183	55	—
ß	5	222	94	—
¡	6	184	56	—
ä	7	196	68	—
¶	8	213	85	—
ö	9	198	70	—
´	10	168	40	—
ü	11	199	71	—
¨	12	—	—	—
	13	—	—	—
É	14	220	92	64
Ø	15	210	82	—
ø	16	214	86	—
Æ	17	211	83	—
æ	18	215	87	—
Å	19	208	80	93
å	20	212	84	125
Ä	21	216	88	91
ä	22	204	76	123
Ö	23	218	90	92
ö	24	206	78	124
Ü	25	219	91	94
ü	26	207	79	126
é	27	197	69	96
!	28	—	—	—
—	29	—	—	—
£	30	187	59	—
†	31	—	—	—

* Printer type 2 sends three characters for each character in the print string—1) shift out (CHR$(14)), 2) the mapped character, and 3) shift in (CHR$(15)). The shift out and shift in characters are not printed.

Table 6-b. Character Mapping for Danish/Norwegian (option 002)

Character	Computer Decimal Code	Printer Decimal Code Type 1	Type 2*	Type 3
§	0	189	61	—
¿	1	185	57	—
°	2	179	51	—
Ñ	3	182	54	—
ñ	4	183	55	—
ß	5	222	94	—
¡	6	184	56	—
á	7	196	68	—
í	8	213	85	—
ó	9	198	70	—
´	10	168	40	—
ú	11	199	71	—
¨	12	—	—	—
	13	—	—	—
É	14	220	92	—
Ø	15	210	82	92
ø	16	214	86	124
Æ	17	211	83	91
æ	18	215	87	123
Å	19	208	80	93
å	20	212	84	125
Ä	21	216	88	—
ä	22	204	76	—
Ö	23	218	90	—
ö	24	206	78	—
Ü	25	219	91	—
ü	26	207	79	—
é	27	197	69	—
|	28	—	—	—
—	29	—	—	—
£	30	187	59	—
†	31	—	—	—

* Printer type 2 sends three characters for each character in the print string—1) shift out (CHR$(14)), 2) the mapped character, and 3) shift in (CHR$(15)). The shift out and shift in characters are not printed.

Table 6-c. Character Mapping for:

German (option 004)

Dutch (option 010)

Swiss German (option 021)

Character	Computer Decimal Code	Printer Decimal Code Type 1	Type 2*	Type 3
§	0	189	61	64
¦	1	181	53	—
°	2	179	51	—
à	3	200	72	—
è	4	201	73	—
ß	5	222	94	126
ì	6	217	89	—
ò	7	202	74	—
ù	8	203	75	—
^	9	170	42	—
´	10	—	—	—
`	11	169	41	—
¨	12	—	—	—
	13	—	—	—
á	14	192	64	—
é	15	193	65	—
í	16	209	81	—
ó	17	194	66	—
ú	18	195	67	—
ë	19	205	77	—
ï	20	221	93	—
Ä	21	216	88	91
ä	22	204	76	123
Ö	23	218	90	92
ö	24	206	78	124
Ü	25	219	91	93
ü	26	207	79	125
é	27	197	69	—
|	28	—	—	—
—	29	—	—	—
£	30	187	59	35
†	31	—	—	—

* Printer type 2 sends three characters for each character in the print string—1) shift out (CHR$(14)), 2) the mapped character, and 3) shift in (CHR$(15)). The shift out and shift in characters are not printed.

Table 6-d. Character Mapping for:

French (option 008)

Italian (option 009)

Swiss French (option 021)

Character	Computer Decimal Code	Printer Decimal Code Type 1	Type 2*	Type 3
§	0	189	61	93
£	1	181	53	92
°	2	179	51	91
à	3	200	72	96,8,97
è	4	201	73	96,8,101
é	5	222	94	—
ì	6	217	89	96,8,105
ò	7	202	74	96,8,111
ù	8	203	75	96,8,117
^	9	170	42	—
´	10	—	—	—
`	11	169	41	—
¨	12	—	—	—
	13	—	—	—
â	14	192	64	94,8,97
ê	15	193	65	94,8,101
î	16	209	81	94,8,105
ô	17	194	66	94,8,111
û	18	195	67	94,8,117
ë	19	205	77	126,8,101
ï	20	221	93	126,8,105
Ä	21	216	88	126,8,65
ä	22	204	76	126,8,97
Ö	23	218	90	126,8,79
ö	24	206	78	126,8,111
Ü	25	219	91	126,8,85
ü	26	207	79	126,8,117
é	27	197	69	123
|	28	—	—	—
—	29	—	—	—
£	30	187	59	35
†	31	—	—	—

* Printer type 2 sends three characters for each character in the print string—1) shift out (CHR$(14)), 2) the mapped character, and 3) shift in (CHR$(15)). The shift out and shift in characters are not printed.

Table 6-e. Character Mapping for Spanish (option 006)

Character	Computer Decimal Code	Printer Decimal Code Type 1	Type 2*	Type 3
§	0	189	61	—
¿	1	185	57	93
°	2	179	51	94
Ñ	3	182	54	92
ñ	4	183	55	124
ß	5	222	94	—
¡	6	184	56	91
á	7	196	68	39,8,97
í	8	213	85	39,8,105
ó	9	198	70	39,8,111
´	10	168	40	—
ú	11	199	71	39,8,117
¨	12	—	—	—
	13	—	—	—
É	14	220	92	39,8,69
Ø	15	210	82	—
ø	16	214	86	—
Æ	17	211	83	—
æ	18	215	87	—
Å	19	208	80	—
å	20	212	84	—
Ä	21	216	88	—
ä	22	204	76	—
Ö	23	218	90	—
ö	24	206	78	—
Ü	25	219	91	—
ü	26	207	79	—
é	27	197	69	39,8,101
|	28	—	—	—
—	29	—	—	—
£	30	187	59	—
†	31	—	—	—

* Printer type 2 sends three characters for each character in the print string—1) shift out (CHR$(14)), 2) the mapped character, and 3) shift in (CHR$(15)). The shift out and shift in characters are not printed.

Local Language Errors

The following errors and warnings are provided by the language ROM.

Number	Message	Meaning
100 (Warning)	ONLY FIRST CHARACTER USED	LPRINT parameter (*string1*) or LESCAPE IS parameter evaluates to a string with more than one character; characters after the first are ignored.
101 (Warning)	LANG UNDEFINED	Computer hardware is not properly configured for local languages—the computer requires servicing.
109	TOO MANY LPRINTS	Attempt to execute an LPRINT statement that exceeds space available for LPRINT mapping.
110	UNKNOWN PRINTER TYPE	Attempt to execute PRINTER TYPE IS with parameter outside the range 1 through 4.
111	LANG ROM	The computer has failed the self-test and requires servicing.
112	INVALID ARGUMENT	Attempt to execute LESCAPE IS with argument outside the range 0 through 255.