HEWLETT-PACKARD EUROPE 5 1 1 9 3 **Program Description I**

Program Title 5x7 matrix letter and symbol printer

Contributor's Name Bjørn Engsig

Address Havrevænget 15

City Allerød Country Denmark Postal Code 3450

Program Description, Equations, Variables Description of the code, used for the 5x7 matrix storage

When printing the symbol, 5 columns of 7 positions each are printed. The 7 positions are first encoded to a binary 7-digit number, where 1 means "print" and 0 "don't print". ("Print is actually "print 8" and "don't print" means "print 1"). This binary number is converted to it's decimal equivalent, and five of these are, asuming they consist of no more than 2 figures, put after each other giving a 10-digit number. If the first digit in that number is 0 it's changed to 1, the entire number is then divided by 10^9. The resulting number is called a. When converting from binary to decimal, some numbers may be greater than 99. If this is the case only the last two figures are used. The first figures (0 or 1) from each of the five numbers are put after each other giving a binary number, which is converted to decimal, giving a number, that's called b. If 0 has been changed to 1 in the number called a, the final code for the 5x7 matrix symbol is $a \cdot 10^{b+64}$. If the change from 0 to 1 has not occured, the code is $a \cdot 10^b$. In this way any combination of "print" and "don't print" is possible and can be stored in a single register.

Operating Limits and Warnings Warning: Don't use the binary to decimal or decimal to binary converter for non-integers or for numbers greater than 127_{DEC} or 1111111_{BIN}.

3179 Program Description 11

Sketch(es)

Sample Problem(s): Print the text "HP-67/97"

Solution: (For print outputs, look at page 3)

Since " doesn't exist on the given datacards it must first be encoded and stored on a card. The 5x7 matrix symbol for " is given to the right. This must be entered and stored in register 1. Press:

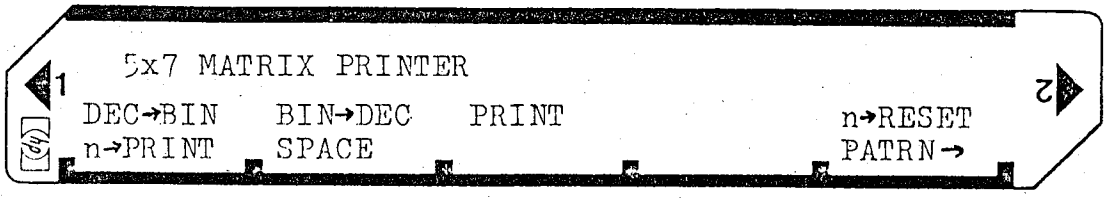
0 ■ 0 ■ 0	top
0 ■ 0 ■ 0	
0 0 0 0 0	
0 0 0 0 0	
0 0 0 0 0	
0 0 0 0 0	
0 0 0 0 0	bottom
1 2 3 4 5	

1 [f] [e] → 0. [R/S] (first column) → 0 11 [R/S] (second column, lower position first, upper position at last) → 0. [R/S] → 0 11 [R/S] → 0. [R/S] → 2 (next register to be used for storage) f W/DATA → Crd Put an unprotected card in the card reader slot [B] → (look page 3) 1 [A] read both sides of datacard 1 8 [A] 16 [A] read both sides of datacard 2

Solution(s): 24 [A] 6 [A] 7 [A] 16 [A] 9 [A] 7 [A] read previously recorded datacard 1 [A]

Reference(s):

```
111111111
111111111
111111881
111111111
111111881
111111111
111111111
188888881
111181111
111181111
111181111
188888881
111111111
111111181
188888881
111181181
111181181
111118811
111111111
111181111
111181111
111181111
111181111
111181111
111111111
118888111
181181811
181181181
181181181
118811111
111111111
111111181
111111181
188881181
111118181
111111881
111111111
181111111
118811111
111181111
111118811
111111181
111111111
111118811
181181181
181181181
118181181
111888811
111111111
111111181
111111181
188881181
111118181
111111881
111111111
111111111
111111881
111111111
111111881
111111111
111111111
```



STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Load both sides of program card.			
2	For printing go to step 3. To encode			
	new symbols go to step 7			
3	Load datacard with codes for the wanted			
	symbols on it.			
4	To print a single space (1/5 of 5x7 mat)		B	output,0
5	To print a full space (full 5x7 matrix			
	plus single space) press	0	A	output, 0
6	To print a symbol, enter indirect num-			
	ber, n, of register in which the code			
	is stored (Look at page 7(8))	n	A	output, 0
	After the 5x7 matrix, a single space			
	is printed. If n is not between 0 and			
	24 an error message or a wrong output			
	occurs. Go to step 3, 4, 5 or 6 for a			
	new print.			
7	Choose register, n, in which the code			
	has to be stored. NOTE: n must be an	n	f e	0
	integer and $1 \leq n \leq 24$			
8	The 5 columns of the 5x7 symbol are entered as a 7-digit binary number. Each digit must be 0 if no printout is wanted ("don't print") and 1 if a printout is wanted ("print"). The first digit in the binary number is the bottom position of the 5x7 matrix. If one ore more of the first digits are 0, they need not be entered. For correct operation calculator must be in FIX 0 mode.			
		bin numb first col	E or R/S	0
	Enter in the same way other columns till	fifth col	E or R/S	n+1
9	If an error is made go to step 7			
10	If one whishes to continue with register			
	n+1 press		f e or R/S	0
	and go to step 8. Else go to step 7			
11	After inputting, codes can be saved on			
	a datacard.			
	Special operations:			
12	Convert decimal to binary	n_{DEC}	f a	n_{BIN}
13	Convert binary to decimal	n_{BIN}	f b	n_{DEC}
	Look at warning on page 1			
14	Convert binary number and printout	n_{BIN}	f c	Output, 0

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBL a	32 25 11	DEC→BIN
	0	00	
	STO 0	33 00	clear R_0 and Y
	x≷Y	35 52	
	GSB 4	31 22 04	
	GSB 4	31 22 04	
	GSB 4	31 22 04	
	GSB 4	31 22 04	
	GSB 4	31 22 04	
010	GSB 4	31 22 04	
	GSB 4	31 22 04	
	R↑	35 54	Get t
	RCL 0	34 00	
	EEX	43	
	7	07	
	X	71	
	RTN	35 22	
	*LBL 4	31 25 04	
	2	02	
020	÷	81	
	ENT	41	
	FRAC	32 83	
	x≷Y	35 52	
	R↓	35 53	Check new and
	x>Y	32 81	last fraction
	SF 2	35 51 02	
	R↓	35 53	
	x≷Y	35 52	relocate stack-
	R↓	35 53	registers
	R↓	35 53	
	1	01	Binary digit is
	F? 2	35 71 02	1 if new fraction>
	F? 2	35 71 02	old fraction else
	CLX	44	0
	STO+0	33 61 00	
	R↓	35 53	
	1	01	
	0	00	
	STO÷0	33 81 00	
040	CLX	44	Bump stack
	+	61	
	RTN	35 22	
	*LBL b	32 25 12	
	0	00	
	STO 0	33 00	Clear R_0
	R↓	35 53	
	EEX	43	
	6	06	
	÷	81	
050	GSB 3	31 22 03	
	GSB 3	31 22 03	
	GSB 3	31 22 03	
	GSB 3	31 22 03	
	GSB 3	31 22 03	
	GSB 3	31 22 03	
	GSB 3	31 22 03	

STEP	KEY ENTRY	KEY CODE	COMMENTS
	R↓	35 53	
	RCL 0	34 00	
	2	02	
060	÷	81	
	RTN	35 22	
	*LBL 3	31 25 03	
	INT	31 83	
	STO+0	33 61 00	add binary digit
	R↓	35 53	to R_0
	2	02	
	STO×0	33 71 00	
	R↓	35 53	
	LSTx	35 82	get remaining
070	FRAC	32 83	binary numbers
	1	01	
	0	00	
	X	71	
	RTN	35 22	
	*LBL A	31 25 11	n→PRINT
	STO I	35 33	
	x=0	31 51	
	GTO 7	22 07	
	RCL (i)	34 24	Get code
080	LOG	31 53	
	6	06	
	4	04	
	x≤Y	32 71	code ≥ 10^{64}
	SF 2	35 51 02	
	R↓	35 53	
	INT	31 83	get exponent of
	ENT	41	code
	10^x	32 53	
	RCL (i)	34 24	
090	x≷Y	35 52	
	÷	81	1≤code<10
	1	01	
	F? 2	35 71 02	
	F? 2	35 71 02	
	CLX	44	
	−	51	subtr. 1 if F2 set
	EEX	43	
	9	09	make code an
	X	71	integer number
100	STO I	35 33	
	R↓	35 53	get exponent of
	GSB a	32 22 11	code and convert
	*LBL 7	31 25 07	
	ENT	41	enable roll down
	GSB 0	31 22 00	
	GSB 0	31 22 00	
	GSB 0	31 22 00	print five times
	GSB 0	31 22 00	
	GSB 0	31 22 00	
110 GTO	~~GSB 0~~	22 31 13	print space
	*LBL 0	31 25 00	
	R↓	35 53	

REGISTERS ALL USED

0 convert	1	2	3	4	5	6	7	8	9
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I

STEP	KEY ENTRY	KEY CODE	COMMENTS
	~~35 53~~		
	1	01	
	0	00	
	÷	81	
	INT	31 83	
	LSTx	35 82	
	FRAC	32 83	get first digit
	EEX	43	
120	3	03	
	X	71	
	RCL I	35 34	
	~~EEX~~	43	
	2	02	
	÷	81	
	INT	31 83	
	STO I	35 33	
	CLX	44	
	LSTx	35 82	
130	FRAC	32 83	get last two digits
	EEX	43	
	2	02	
	X	71	
	+	61	add
	GSB a	32 22 11	convert to binary
	*LBL c	32 25 13	
	7	07	
	0	00	
	X	71	
	1	01	
	1	01	
	1	01	
	1	01	
	1	01	convert to
	1	01	print format
	1	01	
	1	01	
	1	01	
	+	61	
150	DSP 0	23 00	
	PRTX	31 84	PRINT OUT
	CLX	44	
	RTN	35 22	
	*LBL B	31 25 12	SPACE
	0	00	
	GTO c	22 31 13	
	*LBL e	32 25 15	n→RESET
	STO I	35 33	
	CLX	44	
160	STO 0	33 00	clear R_0 and $R_{(i)}$
	STO (i)	33 24	
	*LBL 9	31 25 09	
	CLX	44	
	R/S	84	
	*LBL E	31 25 15	PATTERN→
	RCL 0	34 00	
	X≷Y	35 52	save R_0
168	GSB b	32 22 12	

STEP	KEY ENTRY	KEY CODE	COMMENTS
	X≷Y	35 52	
170	STO 0	33 00	Restore R_0
	R↓	35 53	
	EEX	43	
	2	02	
	~~#~~ X>Y	32 81	if ≥100
	GTO 5	22 05	
	−	51	subtr. 100 and
	1	01	put 1 in first
	STO+0	33 61 00	digit place
	*LBL 5	31 25 05	
180	R↓	35 53	
	STO+(i)	33 61 24	
	1	01	
	0	00	
	STO÷0	33 81 00	
	x²	32 54	
	STO÷(i)	33 81 24	
	RCL I	35 34	
	.	83	
	1	01	counter
190	+	61	
	STO I	35 33	
	FRAC	32 83	
	.	83	
	5	05	five inputs ?
	X≠Y	32 61	
	GTO 9	22 09	
	1	01	
	RCL 0	34 00	
	EEX	43	
200	5	05	
	X	71	convert first-digit
	GSB b	32 22 12	number to decimal
	10^x	32 53	
	RCL (i)	34 24	
	X≷Y	35 52	
	STO (i)	33 24	and store in (i)
	CLX	44	
	1	01	
	0	00	get "last two digit"
210	X	71	number
	X>Y	32 81	
	GTO 8	22 08	if <1
	+	61	add 1
	EEX	43	and multiply
	6	06	by 10^{64}
	4	04	
	X	71	
	*LBL 8	31 25 08	
	STOX(i)	33 71 24	
220	RCL I	35 34	
	RND	31 24	Get next n
	R/S	84	
	GTO e	22 31 15	

LABELS

A n→PRINT	B SPACE	C	D	E PATRN→
a DEC→BIN	b BIN→DEC	c PRINT	d	e n→RESET
0 PRINT	1	2	3 BIN→DEC	4 DEC→BIN
5 used	6	7 used	8 used	9 STOP

FLAGS

0	
1	
2 used	
3	

SET STATUS

FLAGS	ON	OFF
0	☐	☒
1	☐	☒
2	☐	☒
3	☐	☒

TRIG	
DEG	☒
GRAD	☐
RAD	☐

DISP	
FIX	☒
SCI	☐
ENG	☐
n	0

Datacard 1.

Symbol	n	Register contents
A	1	2.418171824 17
B	2	5.473732765 02
C	3	3.465656562 00
D	4	6.265652765 02
E	5	6.573737327 01
F	6	1.109090927 65
G	7	5.073736562 00
H	8	2.708080827 17
I	9	1.065276500 68
J	10	6.365656533 00
K	11	6.534200827 01
L	12	6.464646427 01
M	13	2.702040227 17
N	14	2.732280227 17
O	15	6.265656562 00
P	16	1.609092701 66
R	17	1.225092701 82
S	18	5.073737338 00
T	19	1.101270101 68
U	20	6.364646463 00
V	21	3.132643231 00
W	22	6.364486463 00
X	23	6.534283465 00
Y	24	1.102240201 68

Datacard 2.

Symbol	n	Register contents
1	1	6.464276668 04
2	2	6.669738198 00
3	3	4.975696533 00
4	4	1.627182024 08
5	5	4.973737339 00
6	6	4.873737460 00
7	7	1.305210101 68
8	8	5.473737354 00
9	9	3.041737306 00
0	10	2.834653428 00
Ø	11	6.269738162 00
Q	12	6.297816562 00
Z	13	6.769738197 00
,	14	1.048486400 64
.	15	1.000969600 64
/	16	1.106084864 64
?	17	1.609890102 64
!	18	1.000950000 64
:	19	1.000545400 64
;	20	1.054546400 64
x	21	3.420082034 00
÷	22	1.808420808 64
+	23	1.808620808 64
-	24	1.808080808 64