HEWLETT-PACKARD

KEYBOARD

VOL. 2 NO. 2

TO HEWLETT-PACKARD SYSTEM 9100 USERS:

The *HP KEYBOARD* is published to keep you informed of new HP calculator system software, consisting of programs and programming tips, and information on new peripherals for the growing System 9100.

Most of the programs published in *KEYBOARD* are submitted by HP calculator users like yourself. Your programs and programming ideas which you feel are of interest to a number of other 9100 users will be most welcome.

If you would like to see additional types of articles or material in *KEYBOARD*, please send your ideas to the editor.

SOFTWARE PACKETS AVAILABLE

In the U.S.A., you can order, through your local HP sales office, the following packets of System 9100 programs for your specialized applications:

Part Number	Description	Price
09100-70800	Stat-Pac Vol. 1	$19.50
09100-74100	Surveying Vol. 1	$15.00
09100-75598	Chemical Processing Vol. 1	$15.00

Most of the programs in these packets will work on either the 9100A or 9100B, and appropriate provisions are included for operation of the 9120A Printer and other peripherals. Your local HP calculator salesman will gladly give you more information or arrange a hands-on demonstration on request.

In Europe, you can order the following specialized libraries as well as those listed above:

Part Number	Description	Price
09100-77000	Bautechnische Programme I 9100A (German Structures)	$7.50
09100-78000	Vermessung I (German Surveying)	$15.00
09100-79500	Gear Testing and Design Programmes I (British)	$7.50

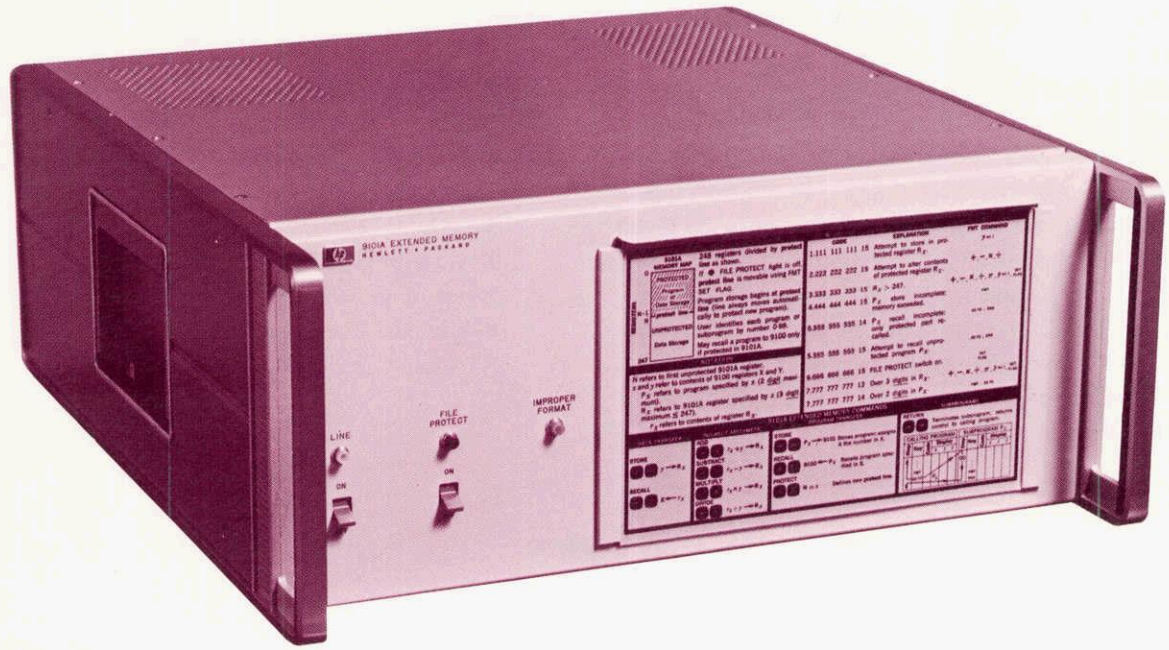
TABLE OF CONTENTS

COVER

The HP Model 9101A Extended Memory, featured in this issue of *KEYBOARD*, has been acclaimed by our customers as the most significant addition to the System 9100 since the 9100B Calculator was announced.

A colorful full-stage presentation announcing the HP Extended Memory is what our art director envisioned for this cover. When you read more about the 9101A's capabilities, you may agree that this type of debut is deserved.

ADDITIONAL MEMORY FOR GROWING PROBLEMS



The HP 9101A Extended Memory significantly increases the computing power of the 9100 Calculator System. The 248 registers of the 9101A, when added to a 9100A, multiply the A's capacity almost 17 times. When it is added to the 9100B, the system has nearly 9 times the storage available. The additional storage can be used for longer programs, more complex programs, storage of data and/or storage of a variety of often-used programs.

A 9100 – 9101A combination gives the system mini-computer capability while maintaining the benefits of the calculator. The man-machine interface remains unchanged---operation continues to be straightforward and easy to understand; information goes in and out easily; programming does not become difficult; and the system remains portable.

There are many computing devices that do not have the capability of the 9100 system and yet are more expensive. The problems with down time, turn-around time, compatibility, system programmers, expensive input/ output devices and computer operators are virtually eliminated on the 9100.

Considering the additional features a 9101A Extended Memory provides and the low-cost peripherals of the 9100 series, you have a powerful system that is hard to beat.

DESCRIPTION

The Hewlett-Packard 9101A is an Extended Memory unit offering expanded storage capacity for the HP 9100 family of calculators. When attached to any 9100A/B, the 9101A provides an additional 248 registers, giving you greater programming and data handling flexibility. These registers are of the same configuration as the calculator registers and usable to store 248 constants or up to 100 programs (totaling 3,472 program steps), or a combination of constants and programs.

A basic system configuration consisting of a 9100A/B calculator and 9101A Extended Memory unit requires no interfacing unit. Additional peripherals such as the 9120A Printer or 9125A Plotter may be added to the basic system using a 9102A Calculator Buffer.

CHARACTERISTICS OF THE HEWLETT-PACKARD 9101A EXTENDED MEMORY

248 REGISTERS --- 3,472 PROGRAM STEPS --- 100 PROGRAM STORAGE CAPABILITY --- INDIRECT ARITHMETIC SUBROUTINES --- INDIRECT ADDRESSING --- DIRECT ADDRESSING --- DIAGNOSTIC CODES

Ease of Operation: The 9101A is addressed by the FORMAT (FMT) instruction from the Calculator followed by a second instruction to define the operation.

User Convenience: Programs are automatically arranged in the Extended Memory. This relieves the investigator of having to remember register locations, and also utilizes the space to maximum efficiency. The memory has a definable protective region which prevents accidental erasures. Any time an illegal operation has been specified for the 9101A, a numeric diagnostic code appears in the calculator X register, identifying the incorrect operation.

Speed: The execution times for functions in the Extended Memory are comparable to the times to perform the same functions in the calculator alone.

Flexibility: All peripherals including the 9101A are plug-in attachments and can be added to the System at any time.

Typical Applications:

- Matrix Inversion, 14 x 14
- N Linear Equations in N Unknowns, $N \leqslant 14$
- Nth Order Polynomial Regression, $N \leqslant 11$
- Roots of an Nth Order Polynomial With Complex (or Real) Coefficients, $N \leqslant 24$

COMMANDS

Data Storage

Enter:

		Z
Data	56.321	Y
Register	246	X

Press: FMT y→( )

Data shown in Y register is stored in register 246. Data can be stored in and recalled from any register #0 - #247; capacity for up to 248 data points.

This saves you the trouble of entering the same data more than once.

Data Recall

Enter:

		Z
		Y
Register	201(51321)	X

Press: FMT π

The number of the register entered in X is replaced by the contents of that register. Indirect addressing can be accomplished by FMT π, FMT___. You can compute addresses or call a register which contains the register number desired.

You can design a program which will automatically sequence the called registers, saving many manual operations.

Indirect Arithmetic

Enter:

		Z
Data	5	Y
Register	243	X

Press: FMT +, (or –, or X, or ÷)

Contents of the Y register are added to contents of register 243. The sum is retained in register 243. A number in an unprotected Extended Memory register may be operated on (+, –, X, or ÷) similarly. Thus an Extended Memory register may be used as an accumulator.

This capability saves you up to 13 steps for each indirect arithmetic operation.

PROGRAMMING

Program Storage

Enter:

	Z
	Y
Program #	0 (10) X

Press: FMT FMT

Programs are transferred from the 9100 in their entirety with their program number designations. The programs are stored automatically and the last register used is displayed. This lets you verify proper storage instantly and see the amount of memory which is still available.

Program Recall

Enter:

	Z
	Y
Program #	2 (37) X

Press: FMT GO TO

FMT GO TO is used to recall a program into the 9100. Program sections are connected in this manner, allowing for long programs.
This saves you the need for connecting programs manually and manipulating intermediate results.

Memory Protect

Enter:

	Z
First register	Y
not protected	88 X

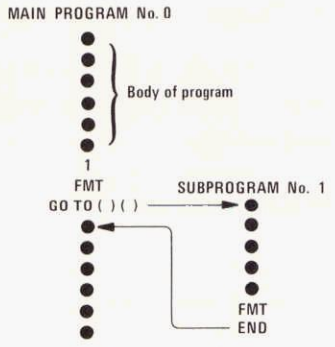
Press: FMT SET FLAG

The memory protect line can be manually set at any register. When a program is entered, the protect line automatically moves to include it. Accidental user errors and confusion are greatly reduced.

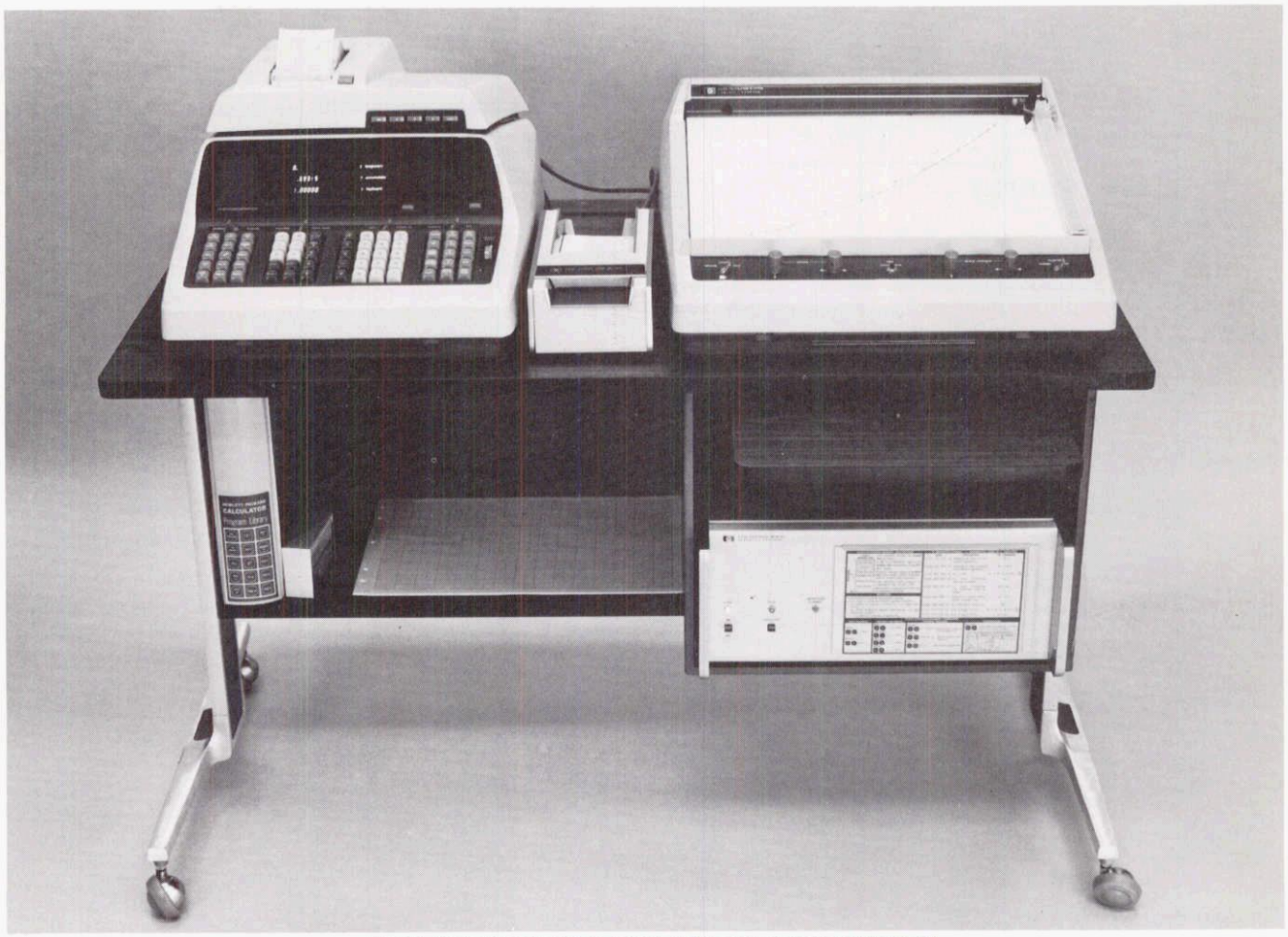
SUBPROGRAMMING

(Subroutine)

Programs that end in FMT END act like subroutines. The main program is called and resumed one step after the step calling the subprogram. THIS GIVES THE 9100A SUBROUTINE CAPABILITY.
Subprograms can call other subprograms, providing "nesting" capability. The 9101A can nest subprograms up to 14 deep.



As your problems grow--your system grows to solve them



HEWLETT-PACKARD 9100 CALCULATOR SYSTEM

9100A/B Calculator -- 9120A Printer

The 9100A, 16 registers - 192 program steps and the 9100B, 32 registers - 392 program steps have their capabilities greatly expanded by the 248 registers, 3,472 program steps of the Extended Memory. The 9100 - 9101A combinations will now solve lengthier and more complex problems. The 9101A adds subroutine capability to the 9100A Calculator.

9125A Calculator Plotter

The 9125A can plot any X, Y coordinates the calculator can generate. With the added capacity of the 9101A, more complex plots can be made.

9160A Mark Sense Card Reader

The 9160A can be used to rapidly enter data and program steps into the calculator, then into the Extended Memory.

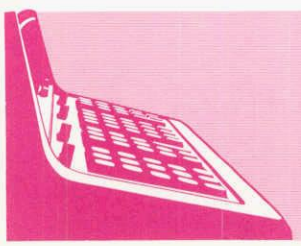
PRICE

9100A Calculator	$3950
9100B Calculator	4900
9125A Plotter	2475
9120A Printer	975
9160A Card Reader	590
9101A Extended Memory	3690
9102A Calculator Buffer	280
11163A System Desk	360

Rent and lease plans available.

Service Contracts available.

The HP 9102A Calculator Buffer allows you to use a 9100 Calculator and a 9101A Extended Memory with a 9120A Printer and/or 9125A Plotter. The buffer isolates the peripherals in a System and strengthens the output signals necessary for proper operation.



MATRIX INVERSION, N ≤ 14

EXTENDED MEMORY PROGRAM II

This program will determine the inverse, A^{-1}, of a square matrix, A. The matrix can be shown in the following form:

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1N} \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2N} \\ A_{31} & A_{32} & A_{33} & \cdots & A_{3N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ A_{N1} & A_{N2} & A_{N3} & \cdots & A_{NN} \end{pmatrix}$$

where the maximum value of N is 14.

The program uses a modified Gauss-Jordan elimination technique. New elements are overlayed as they are computed in order to save storage space. In addition, a pivot search is utilized to maximize accuracy and allow the inclusion of "zero" elements in the initial matrix.

The user inputs the value of "N" and the matrix elements, row by row. The elements are stored in the 9101A from register 247 through [247 - (N x N)]. After the matrix has been loaded, the user may change any of the elements that were incorrectly specified by using the corrector routine.

In the event that the matrix is singular, the calculator will execute a programmed error routine. This routine calls for division by zero placing a $9.999\ 999\ 999 \times 10^{99}$ in the Y-register and lighting of the error light. The program will also stop execution.

Maximum execution time (for N = 14) is approximately 4 minutes. Upon completion of the program, the calculator will printout the inverse matrix, row by row. The inverse matrix is now stored in the 9101A over the original matrix.

Reference: V. N. Faddeeva, *Computational Methods of Linear Algebra*, Dover Publications, Inc., New York, 1959. pp. 85-89

EQUIPMENT NEEDED ☐ 9100A ☐ 9100B ☑ 9120 ☐ 9125 ☐ 9160 ☑ 9101

☑ DEGREES ☐ RADIANS ☐ FLOATING ☑ FIXED DECIMAL WHEEL AT 4 PRESS x ☐ ☐ ON 9120

STEP	USER INSTRUCTIONS	DISPLAY		
	PROGRAM LOADING			
	File protect switch: OFF			
1.	PRESS: END			
2.	PRESS: CLEAR, FMT, SET FLAG	0	0	0
3.	ENTER PROGRAM P_0			
4.	PRESS: 0, FMT, FMT	5	0	0
5.	ENTER PROGRAM P_1			
6.	PRESS: 1, FMT, FMT	15	0	0
7.	ENTER PROGRAM P_2			
8.	PRESS: 2, FMT, FMT	22	0	0
9.	ENTER PROGRAM P_3			
10.	PRESS: 3, FMT, FMT	31	0	0
11.	ENTER PROGRAM P_4			
12.	PRESS: 4, FMT, FMT	36	0	0
	PROGRAM EXECUTION			
1.	PRESS: 0, FMT, GO TO	5		
2.	PRESS: CONTINUE	14	(Maximum N)	
3.	ENTER DATA: N	N		
4.	PRESS: CONTINUE	0		
5.	ENTER DATA: A_{ij}	A_{ij}	j (col)	i (row)
6.	PRESS: CONTINUE		j	i
	Return to Step No. 5 to enter the next element of the matrix.			
	After all elements have been entered:			
7.	Check the printout to be sure the correct values have been entered.			
8.	If incorrect values have been entered, proceed to the corrector option.			
9.	If all the elements are correct, PRESS: CONTINUE			
	The calculator will begin execution.			
	To rerun the program, return to Step No. 1.			

EQUIPMENT NEEDED ☐ 9100A ☐ 9100B ☑ 9120 ☐ 9125 ☐ 9160 ☑ 9101

☑ DEGREES ☐ RADIANS ☐ FLOATING ☑ FIXED DECIMAL WHEEL AT 4 PRESS x ☐ ☐ ON 9120

CORRECTOR OPTION

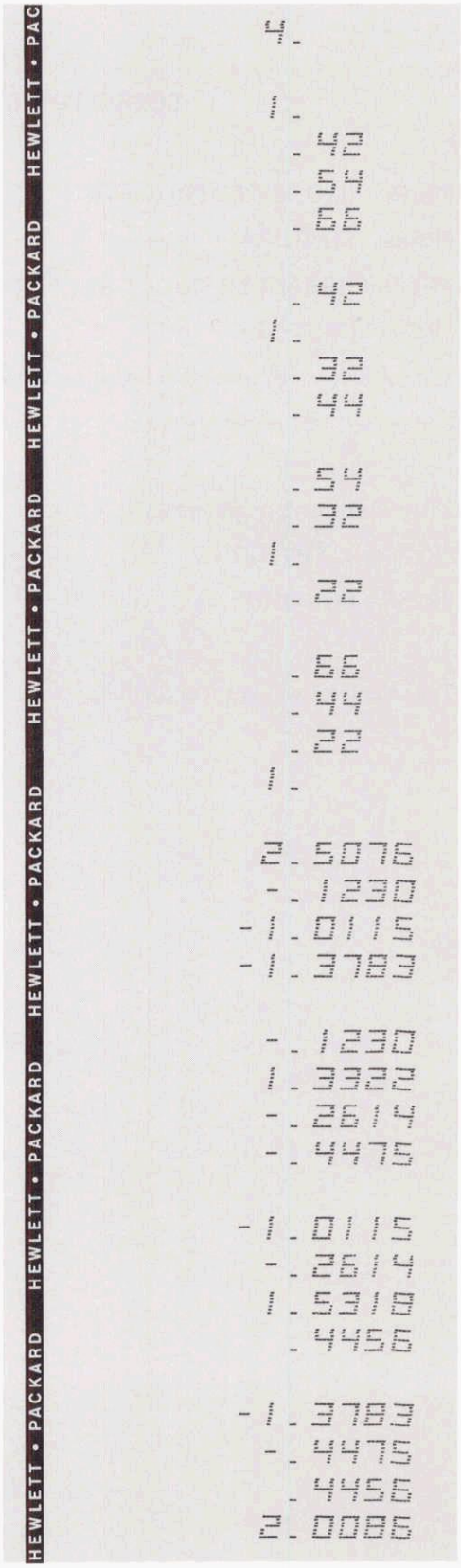
STEP	USER INSTRUCTIONS	DISPLAY		
1.	PRESS: END, ENTER PROGRAM			
2.	PRESS: CONTINUE	0	0	0
3.	ENTER CORRECT DATA: i, j, A_{ij}	A_{ij}	j	i
4.	PRESS: CONTINUE	0	0	0
	The calculator will store the correct A_{ij} value in the 9101A. To replace additional elements, return to Step No. 3.			
	After all corrections are made:			
5.	PRESS: 1, FMT, GO TO			
6.	PRESS: CONTINUE			
	The calculator will begin execution.			

EXAMPLE

$$A = \begin{pmatrix} 1.0 & .42 & .54 & .66 \\ .42 & 1.0 & .32 & .44 \\ .54 & .32 & 1.0 & .22 \\ .66 & .44 & .22 & 1.0 \end{pmatrix}$$

$$A^{-1} = \begin{pmatrix} 2.5075 & -.1230 & -1.0115 & -1.3783 \\ -.1230 & 1.3322 & -.2614 & -.4474 \\ -1.0115 & -.2614 & 1.5318 & .4456 \\ -1.3783 & -.4474 & .4456 & 2.0085 \end{pmatrix}$$

SAMPLE PRINTOUT



EXTENDED MEMORY Program II

Program P_0

Step	Key	Code
00	CLR	20
01	1	01
02	4	04
03	STP	41
04	PNT	45
05	PNT	45
06	XTO	23
07	d	17
08	1	01
09	+	33
0a	AC+	60
0b	RCL	61
0c	UP	27
0d	CLX	37

Step	Key	Code
50	FMT	42
51	GTO	44
52	END	46

Step	Key	Code
40	8	10
41	XEY	30
42	-	34
43	DN	25
44	FMT	42
45	π	56
46	UP	27
47	b	14
48	X=Y	50
49	1	01
4a	5	05
4b	GTO	44
4c	2	02
4d	7	07

Step	Key	Code
90	XEY	30
91	FMT	42
92	YTO	40
93	CLX	37
94	XTO	23
95	f	15
96	2	02
97	FMT	42
98	GTO	44
99	4	04
9a	FMT	42
9b	GTO	44
9c	END	46

Step	Key	Code
40	DIV	35
41	DIV	35
42	e	12
43	UP	27
44	1	01
45	-	34
46	d	17
47	X	36
48	e	12
49	+	33
4a	2	02
4b	4	04
4c	8	10
4d	XEY	30

Step	Key	Code
20	2	02
21	4	04
22	8	10
23	UP	27
24	f	15
25	-	34
26	e	12
27	UP	27
28	1	01
29	-	34
2a	d	17
2b	X	36
2c	DN	25
2d	-	34

Step	Key	Code
70	-	34
71	DN	25
72	FMT	42
73	π	56
74	CNT	47
75	CHS	32
76	UP	27
77	c	16
78	UP	27
79	1	01
7a	-	34
7b	d	17
7c	X	36
7d	e	12

Program P_1

Step	Key	Code
10	STP	41
11	XTO	23
12	c	16
13	d	17
14	X=Y	50
15	CNT	47
16	SFL	54
17	c	16
18	IFG	43
19	CNT	47
1a	PNT	45
1b	PNT	45
1c	RDN	31
1d	1	01

Step	Key	Code
00	a	13
01	UP	27
02	1	01
03	+	33
04	d	17
05	X<Y	52
06	9	11
07	9	11
08	YTO	40
09	a	13
0a	CLX	37
0b	XTO	23
0c	f	15
0d	XTO	23

Step	Key	Code
50	b	14
51	UP	27
52	1	01
53	-	34
54	d	17
55	X	36
56	a	13
57	+	33
58	2	02
59	4	04
5a	8	10
5b	XEY	30
5c	-	34
5d	DN	25

Program P_2

Step	Key	Code
00	f	15
01	UP	27
02	1	01
03	+	33
04	YTO	40
05	f	15
06	d	17
07	X<Y	52
08	3	03
09	b	14
0a	e	12
0b	X=Y	50
0c	0	00
0d	2	02

Step	Key	Code
50	-	34
51	DN	25
52	FMT	42
53	X	36
54	CLX	37
55	XTO	23
56	c	16
57	3	03
58	FMT	42
59	GTO	44
5a	CLX	37
5b	XTO	23
5c	f	15
5d	1	01

Step	Key	Code
30	DN	25
31	FMT	42
32	π	56
33	UP	27
34	c	16
35	UP	27
36	1	01
37	-	34
38	d	17
39	X	36
3a	e	12
3b	XEY	30
3c	+	33
3d	2	02

Step	Key	Code
80	+	33
81	2	02
82	4	04
83	8	10
84	XEY	30
85	-	34
86	DN	25
87	FMT	42
88	X	36
89	GTO	44
8a	0	00
8b	0	00
8c	FMT	42
8d	END	46

Step	Key	Code
20	-	34
21	d	17
22	X	36
23	f	15
24	+	33
25	2	02
26	4	04
27	8	10
28	XEY	30
29	-	34
2a	DN	25
2b	FMT	42
2c	YTO	40
2d	f	15

Step	Key	Code
10	b	14
11	XTO	23
12	e	12
13	XTO	23
14	c	16
15	b	14
16	UP	27
17	1	01
18	+	33
19	d	17
1a	X<Y	52
1b	7	07
1c	5	05
1d	YTO	40

Step	Key	Code
60	FMT	42
61	π	56
62	UP	27
63	|Y|	55
64	c	16
65	X>Y	53
66	1	01
67	5	05
68	X=Y	50
69	1	01
6a	5	05
6b	YTO	40
6c	c	16
6d	b	14

Step	Key	Code
10	UP	27
11	1	01
12	-	34
13	d	17
14	X	36
15	e	12
16	+	33
17	2	02
18	4	04
19	8	10
1a	XEY	30
1b	-	34
1c	DN	25
1d	FMT	42

Step	Key	Code
60	FMT	42
61	GTO	44
62	END	46

Step	Key	Code
40	4	04
41	8	10
42	XEY	30
43	-	34
44	DN	25
45	FMT	42
46	π	56
47	X	36
48	c	16
49	UP	27
4a	1	01
4b	-	34
4c	d	17
4d	X	36

Program P_3

Step	Key	Code
30	UP	27
31	d	17
32	X=Y	50
33	3	03
34	b	14
35	CLX	37
36	UP	27
37	1	01
38	GTO	44
39	0	00
3a	a	13
3b	RCL	61
3c	d	17
3d	X=Y	50

Step	Key	Code
20	b	14
21	a	13
22	UP	27
23	1	01
24	X=Y	50
25	5	05
26	0	00
27	f	15
28	UP	27
29	1	01
2a	+	33
2b	a	13
2c	UP	27
2d	1	01

Step	Key	Code
70	XTO	23
71	e	12
72	GTO	44
73	1	01
74	5	05
75	e	12
76	UP	27
77	CLX	37
78	X=Y	50
79	DIV	35
7a	STP	41
7b	d	17
7c	UP	27
7d	X	36

Step	Key	Code
20	π	56
21	UP	27
22	CLX	37
23	X=Y	50
24	DIV	35
25	STP	41
26	e	12
27	UP	27
28	1	01
29	-	34
2a	d	17
2b	X	36
2c	f	15
2d	+	33

Step	Key	Code
00	c	16
01	UP	27
02	1	01
03	+	33
04	d	17
05	X<Y	52
06	8	10
07	c	16
08	YTO	40
09	c	16
0a	e	12
0b	X=Y	50
0c	0	00
0d	2	02

Step	Key	Code
50	f	15
51	XEY	30
52	+	33
53	2	02
54	4	04
55	8	10
56	XEY	30
57	-	34
58	DN	25
59	FMT	42
5a	-	34
5b	f	15
5c	GTO	44
5d	1	01

Step	Key	Code
40	7	07
42	f	15
43	AC-	63
44	GTO	44
45	0	00
46	8	10
47	CLR	20
48	XTO	23
49	a	13
4a	XTO	23
4b	b	14
4c	STP	41
4d	1	01

Step	Key	Code
30	-	34
31	DN	25
32	X<Y	52
33	5	05
34	0	00
35	YTO	40
36	f	15
37	d	17
38	UP	27
39	X	36
3a	f	15
3b	+	33
3c	2	02
3d	4	04

Step	Key	Code
80	a	13
81	+	33
82	2	02
83	4	04
84	8	10
85	XEY	30
86	-	34
87	e	12
88	UP	27
89	CLX	37
8a	X=Y	50
8b	DIV	35
8c	STP	41
8d	DN	25

Step	Key	Code
30	2	02
31	4	04
32	8	10
33	XEY	30
34	-	34
35	DN	25
36	FMT	42
37	DIV	35
38	GTO	44
39	0	00
3a	0	00
3b	DN	25
3c	1	01
3d	XEY	30

Step	Key	Code
10	CLX	37
11	UP	27
12	1	01
13	+	33
14	d	17
15	X<Y	52
16	6	06
17	1	01
18	YTO	40
19	f	15
1a	e	12
1b	X=Y	50
1c	1	01
1d	2	02

Step	Key	Code
60	1	01
61	e	12
62	UP	27
63	1	01
64	-	34
65	d	17
66	X	36
67	e	12
68	XEY	30
69	+	33
6a	2	02
6b	4	04
6c	8	10
6d	XEY	30

Storage

Plus Page

f
e
d
c
b
a
9
8
7
6
5
4
3
2
1
0

Minus Page

f
e
d
c
b
a
9
8
7
6
5
4
3
2
1
0

HEWLETT-PACKARD

Program P_4 Corrector Option

Step	Key	Code
00	CLR	20
01	1	01
02	UP	27
03	AC+	60
04	XEY	30
05	UP	27
06	1	01
07	-	34
08	d	17
09	X	36
0a	DN	25
0b	+	33
0c	2	02
0d	4	04
10	8	10
11	XEY	30
12	-	34
13	UP	27
14	RCL	61
15	RUP	22
16	FMT	42
17	π	56
18	XTO	23
19	c	16
1a	d	17
1b	X=Y	50
1c	4	04
1d	1	01
20	c	16
21	PNT	45
22	1	01
23	+	33
24	d	17
25	X<Y	52
26	3	03
27	0	00
28	YTO	40
29	f	15
2a	DN	25
2b	GTO	44
2c	0	00
2d	4	04
30	RCL	61
31	1	01
32	+	33
33	d	17
34	X<Y	52
35	4	04
36	7	07
37	1	01
38	XTO	23
39	f	15
3a	YTO	40
3b	e	12
3c	GTO	44
3d	0	00
40	4	04
41	c	16
42	PNT	45
43	PNT	45
44	GTO	44
45	2	02
46	2	02
47	CLR	20
48	STP	41
49	END	46

Step	Key	Code
00	CLR	20
01	STP	41
02	PNT	45
03	PNT	45
04	AC+	60
05	DN	25
06	1	01
07	-	34
08	d	17
09	X	36
0a	e	12
0b	+	33
0c	2	02
0d	4	04
10	8	10
11	XEY	30
12	-	34
13	f	15
14	XEY	30
15	FMT	42
16	YTO	40
17	GTO	44
18	0	00
19	0	00
1a	END	46

Step Key Code

Step Key Code

Step Key Code

Step Key Code

Step Key Code

HEWLETT-PACKARD

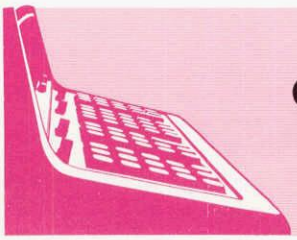
Storage

Plus Page

f
e
d
c
b
a
9
8
7
6
5
4
3
2
1
0

Minus Page

f
e
d
c
b
a
9
8
7
6
5
4
3
2
1
0



QUEUEING WITH FINITE CUSTOMERS

by Edgar J. Schmidt

This program computes the "average number in system, $\overline{n}$" and the "queue length, L_Q" when the following parameters are known:

N = Number of Available Customers
MTTR = Mean Service Time Per Call
MTBF = Mean Time Between Calls for Service Per Customer

The recursive equation applied is:

$$\frac{X}{1-X}\ (N-n+1)\ P_{n-1} = \left\{ \begin{array}{l} nP_n \text{ when } 0 \leqslant n \leqslant M \\ MP_n \text{ when } M < n \leqslant N \end{array} \right\}$$

where

$$X = \frac{MTTR}{MTTR + MTBF}$$

and M = Number of Service Channels (Assigned By Program)
P_n = Probability of n customers in the system.

The above recursive formula is iteratively applied to solve

$$\overline{n} = \frac{\Sigma\, nP_n}{\Sigma\, P_n} \quad \text{(average number in system)}$$

$$F = \frac{\frac{\overline{n}}{N} - 1}{X-1} \quad \text{(efficiency factor)}$$

and $L_Q = N\,(1-F)$. (queue length)

This program assigns successive integer values to M (number of service channels) and iteratively solves for $\overline{n}$ and L_Q for each M.

For certain highly unlikely combinations of N, MTTR and MTBF, an overflow condition will occur. However the program is protected against this eventuality and it re-cycles to the initial DISPLAY when an illegal combination is attempted.

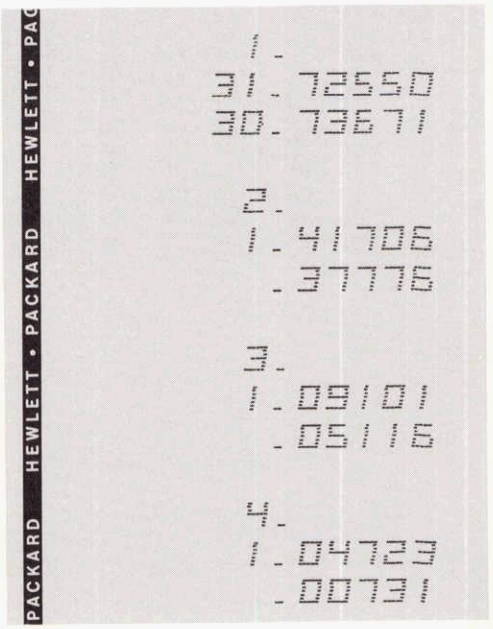
Computing times for each channel value (M) can range between 30-60 seconds.

EXAMPLE

Given a "system" with 625 units (customers) each having a 30,000 hour MTBF, and maintenance conditions such that the MTTR is 50 hours, find the average number in system in queue or being serviced ($\overline{n}$) and the queue length L_Q for varying M (number of channels).

Input data:
N = 625 (number of units–customers–available)
MTTR = 50 (mean time to restore)
MTBR = 30,000 (mean time between failures; at 'failure' it "calls for service")

Results:



Reference: *Finite Queueing Tables*, Peck and Hazelwood, Wiley, 1958.

EQUIPMENT NEEDED ☐ 9100A ☐ 9100B ☑ 9120 ☐ 9125 ☐ 9160 ☐ 9101

☑ DEGREES ☐ RADIANS ☐ FLOATING ☑ FIXED DECIMAL WHEEL AT 5 PRESS x y z ON 9120

STEP	USER INSTRUCTIONS	DISPLAY		
		X	Y	Z
1.	PRESS: END			
2.	ENTER PROGRAM			
3.	PRESS: CONTINUE	1	0	0
4.	ENTER DATA	MTBF	MTTR	N
5.	PRESS: CONTINUE	L_Q	$\bar{n}$	M
6.	To run another case, return to Step 3.			

Step	Key	Code	Step	Key	Code	Step	Key	Code	Step	Key	Code	Step	Key	Code	Step	Key	Code
00	CLR	20	20	a	13	40	DN	25	60	c	16	80	c	16			
01	XTO	23	21	UP	27	41	YE	24	61	DIV	35	81	X	36			
02	9	11	22	1	01	42	9	11	62	1	01	82	b	14			
03	1	01	23	UP	27	43	+	33	63	-	34	83	RUP	22			
04	STP	41	24	CHS	32	44	YTO	40	64	UP	27	84	a	13			
05	PNT	45	25	AC+	60	45	9	11	65	d	17	85	RUP	22			
06	PNT	45	26	RCL	61	46	EEX	26	66	XEY	30	86	PNT	45			
07	DIV	35	27	c	16	47	9	11	67	+	33	87	PNT	45			
08	YTO	40	28	X<Y	52	48	8	10	68	RDN	31	88	STP	41			
09	d	17	29	5	05	49	X<Y	52	69	CHS	32	89	c	16			
0a	DN	25	2a	7	07	4a	0	00	6a	X	36	8a	RDN	31			
0b	0	00	2b	DN	25	4b	0	00	6b	.	21	8b	GTO	44			
0c	YTO	40	2c	d	17	4c	DN	25	6c	9	11	8c	1	01			
0d	c	16	2d	X	36	4d	a	13	6d	X<Y	52	8d	1	01			
10	RUP	22	30	f	15	50	+	33	70	7	07	90	END	46			
11	1	01	31	X	36	51	YTO	40	71	c	16						
12	+	33	32	e	12	52	a	13	72	XEY	30						
13	YTO	40	33	UP	27	53	DN	25	73	DIV	35						
14	b	14	34	b	14	54	GTO	44	74	b	14						
15	RUP	22	35	X<Y	52	55	2	02	75	X	36						
16	+	33	36	XEY	30	56	2	02	76	c	16						
17	YTO	40	37	CNT	47	57	CLR	20	77	XEY	30						
18	f	15	38	DN	25	58	YE	24	78	INT	64						
19	EEX	26	39	DIV	35	59	9	11	79	GTO	44						
1a	9	11	3a	UP	27	5a	a	13	7a	1	01						
1b	3	03	3b	DN	25	5b	DIV	35	7b	0	00						
1c	CHS	32	3c	e	12	5c	YTO	40	7c	RDN	31						
1d	XTO	23	3d	X	36	5d	a	13	7d	-	34						

QUEUEING WITH INFINITE CUSTOMERS

by Edgar J. Schmidt

This program computes the "queue length" of a system when the following system parameters are known:

MTTR = Mean Service Time Per Call
MTBF = "Ensemble" Mean Time Between Calls for Service From Total Source of Customers

The equation applied is:

$$L_Q = (M-R)\left[\frac{\sum_{n=0}^{M-1}\frac{1}{n!}R^n}{\frac{M}{M-R}\frac{1}{M!}R^M}+1\right]^{-1}$$

where, M = Number of Service Channels (assigned by program)

$$R = \frac{MTTR}{MTBF}$$

L_Q is defined to be "average queue-length".

This program assigns successive integer values to M (number of service channels) and determines L_Q for each M. The user can optionally output

AV_S = Average Time In System
AV_W = Average Time In Waiting

EXAMPLE

Consider the problem posed by the program for Queueing With Finite Customers. To convert the finite model for an infinite model, determine the ensemble

$$MTBF = \frac{\text{unit MTBF}}{N}$$

For the example,

Unit MTBF = 30,000
N = 625

Thus

MTTR = 50
ensemble MTBF = 48

Results:

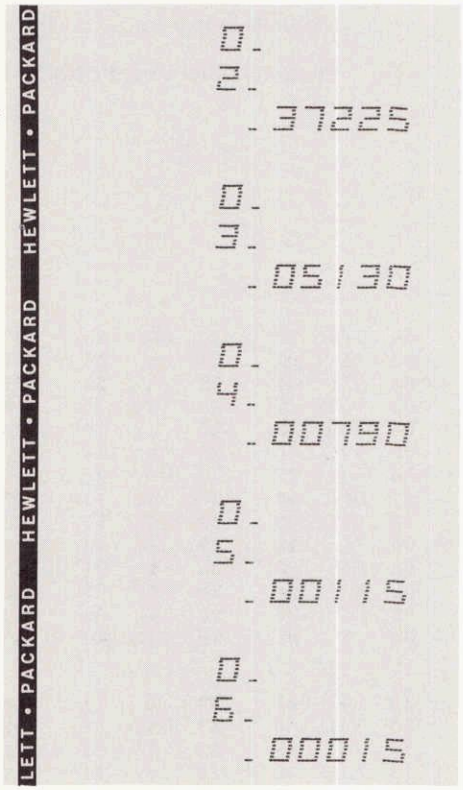
M = 2.
L_Q = .37225*

= 3.
= .05130

= 4.
= .00790

= 5.
= .00115

= 6.
= .00015



*PRESS: SET FLAG – CONTINUE to obtain:
AV_S = 68.61227 hrs.
AV_W = 18.61227 hrs.

Edgar J. Schmidt holds a BSME from Cooper Union (New York City). For the past eight years, he has been an engineering specialist in the Equipment and Systems Evaluation Branch, ITT Federal Electric Corporation, Paramus, New Jersey. His work involves applied mathematics and computer applications, primarily applied to the prediction and math modeling of communication system reliability, maintainability and availability.

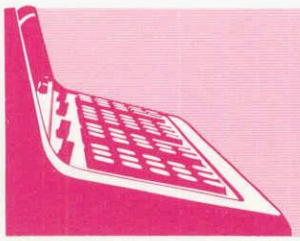
Reference: Operations Research–Methods and Problems, by Lasieni, Yaspan and Friedman

EQUIPMENT NEEDED ☐ 9100A ☐ 9100B ☑ 9120 ☐ 9125 ☐ 9160 ☐ 9101

☑ DEGREES ☐ RADIANS ☐ FLOATING ☑ FIXED DECIMAL WHEEL AT 5 PRESS x y z ON 9120

STEP	INSTRUCTIONS	DISPLAY X	Y	Z
1.	PRESS: END			
2.	ENTER PROGRAM			
3.	PRESS: CONTINUE	1	0	0
4.	ENTER DATA	MTBF	MTTR	
5.	PRESS: CONTINUE	L_Q	M	0
6.	If AV_S and AV_W are desired, PRESS: SET FLAG, CONTINUE. Otherwise return to Step 5 for next value of M and L_Q.	AV_W	AV_S	M
7.	To run another case, PRESS: END and return to Step 3.			

Step	Key	Code	Step	Key	Code	Step	Key	Code	Step	Key	Code	Step	Key	Code	Step	Key	Code
00	CLR	20	20	DIV	35	40	7	07	60	RDN	31	80	XEY	30			
01	1	01	21	c	16	41	d	17	61	X	36	81	X	36			
02	STP	41	22	X	36	42	UP	27	62	RDN	31	82	XEY	30			
03	PNT	45	23	YTO	40	43	c	16	63	XEY	30	83	+	33			
04	PNT	45	24	c	16	44	+	33	64	+	33	84	PNT	45			
05	YTO	40	25	b	14	45	YTO	40	65	RDN	31	85	PNT	45			
06	9	1	26	UP	27	46	d	17	66	X	36	86	STP	41			
07	DIV	35	27	e	12	47	RCL	61	67	DN	25	87	DN	25			
08	YTO	40	28	X=Y	50	48	a	13	68	DIV	35	88	GTO	44			
09	f	15	29	3	03	49	X>Y	53	69	a	13	89	0	00			
0a	DN	25	2a	4	04	4a	5	05	6a	XEY	30	8a	d	17			
0b	INT	64	2b	1	01	4b	1	01	6b	UP	27	8b	END	46			
0c	UP	27	2c	+	33	4c	XTO	23	6c	CLX	37						
0d	1	01	2d	YTO	40	4d	e	12	6d	RDN	31						
10	+	33	30	b	14	50	SFL	54	70	PNT	45						
11	YTO	40	31	GTO	44	51	1	01	71	PNT	45						
12	a	13	32	1	01	52	GTO	44	72	STP	41						
13	XTO	23	33	b	14	53	1	01	73	IFG	43						
14	d	17	34	IFG	43	54	7	07	74	7	07						
15	XTO	23	35	5	05	55	d	17	75	9	11						
16	e	12	36	5	05	56	RUP	22	76	GTO	44						
17	XTO	23	37	1	01	57	DIV	35	77	0	00						
18	b	14	38	UP	27	58	a	13	78	d	17						
19	XTO	23	39	CLX	37	59	DIV	35	79	UP	27						
1a	c	16	3a	AC+	60	5a	UP	27	7a	YE	24						
1b	f	15	3b	a	13	5b	f	15	7b	9	11						
1c	UP	27	3c	X=Y	50	5c	-	34	7c	YTO	40						
1d	b	14	3d	4	04	5d	1	01	7d	9	11						



UNIVERSAL FOUR OPERATIONS PROGRAM

by Claude Cardot

PART NO. 09100–70056

This program, which can be used either with the 9100A or 9100B, will perform a given arithmetic operation on a list of numbers. It will add, subtract, multiply or divide each number in the list by a given constant designated by the operator. The user enters the constant and the type of operation in a single entry. The author has found that the program saves a great deal of time in operations such as unit conversions, and it is not necessary to write a special program for each type of operation.

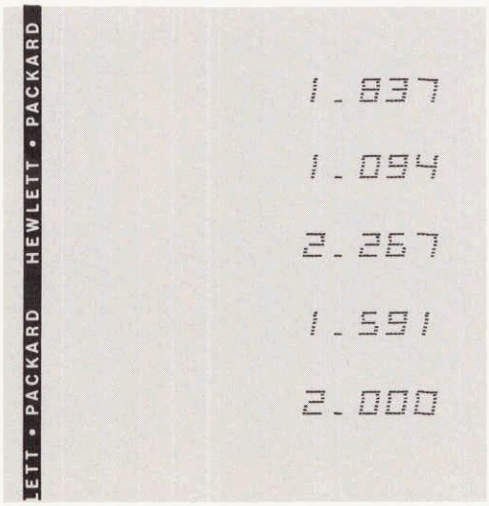
Although printout of each result is included, the program may be used without the printer by changing the print commands in steps 24, 36, 46 and 56 to CONTINUE commands.

EXAMPLES

1. $K = \pi \rightarrow X$

   Operation: ÷

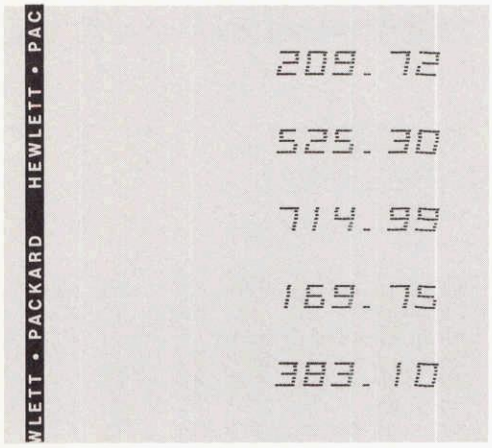
X_i	RESULT
5.770	1.837
3.438	1.094
7.123	2.267
4.999	1.591
6.283	2.000



2. $K = \sin 65^{o} \rightarrow X$

   Operation: x (Multiply)

X_i	RESULT
231.4	209.72
579.6	525.30
788.9	714.99
187.3	169.75
422.7	383.10



Claude Cardot is a supervisor of the Division des Applications Electroniques, Centre de Recherches de la Compagnie Generale d'Electricite, Laboratoires de Marcoussis, Marcoussis, France. Mr. Cardot is a graduate of Paris Ecole Polytechnique and National School of Telecommunications. He holds a Licencie es Sciences.

PART NO. 09100-70056

EQUIPMENT NEEDED ☐ 9100A ☐ 9100B ☑ 9120 ☐ 9125 ☐ 9160 ☐ 9101

☑ DEGREES ☐ RADIANS ☐ FLOATING ☑ FIXED DECIMAL WHEEL AT 3 PRESS ☐ ☐ z ON 9120

STEP	USER INSTRUCTIONS	DISPLAY X	Y	Z
1.	PRESS: END			
2.	ENTER Program			
3.	PRESS: CONTINUE	0	1	0
4.	ENTER DATA (Operating Number)	K		
5.	PRESS: +, –, X, or ÷ to designate desired operation			
6.	PRESS: CONTINUE	1	K	
7.	ENTER DATA	x_i		
8.	PRESS: CONTINUE	x_i	K	Result
9.	Return to Step 7 to enter next x_i			
10.	For new operating number K and/or new type of operation, PRESS: END,			
	CONTINUE and return to Step 3.			

Step	Key	Code	Step	Key	Code	Step	Key	Code	Step	Key	Code	Step	Key	Code	Step	Key	Code
00	CLR	20	20	STP	41	40	STP	41									
01	1	01	21	X	36	41	XEY	30									
02	RUP	22	22	XEY	30	42	UP	27									
03	STP	41	23	RDN	31	43	DN	25									
04	AC+	60	24	PNT	45	44	-	34									
05	-	34	25	STP	41	45	RUP	22									
06	1	01	26	RUP	22	46	PNT	45									
07	X=Y	50	27	f	15	47	STP	41									
08	2	02	28	RDN	31	48	GTO	44									
09	c	16	29	GTO	44	49	4	04									
0a	RCL	61	2a	2	02	4a	1	01									
0b	+	33	2b	1	01	4b	RCL	61									
0c	1	01	2c	RCL	61	4c	RUP	22									
0d	X=Y	50	2d	RUP	22	4d	1	01									
10	3	03	30	1	01	50	STP	41									
11	b	14	31	STP	41	51	XEY	30									
12	RCL	61	32	+	33	52	UP	27									
13	UP	27	33	RUP	22	53	DN	25									
14	1	01	34	f	15	54	DIV	35									
15	XEY	30	35	XEY	30	55	RUP	22									
16	DIV	35	36	PNT	45	56	PNT	45									
17	DN	25	37	STP	41	57	STP	41									
18	X=Y	50	38	GTO	44	58	GTO	44									
19	4	04	39	3	03	59	5	05									
1a	b	14	3a	2	02	5a	1	01									
1b	RCL	61	3b	RCL	61	5b	END	46									
1c	RUP	22	3c	RUP	22												
1d	1	01	3d	1	01												

CHARACTER PLOT PROGRAM

by Raymond J. Schneider, Jr.

PART NO.
09100–76011

This program, written for the Hewlett-Packard Model 9100A or 9100B Calculator with a 9125A Plotter, uses the calculator's digit keys to form a 3 x 3 matrix on which the form of any desired character may be typed.

This program, on a single side of a magnetic card, can be used to produce any desired amount of letters, numbers and symbols without re-entry. Arbitrary key strokes are possible, and inclusion of the zero key in its respective position allows plotting below the base line for commas, lower case p's, q's, and other characters.

Other features of this program allow printing in ascending, descending, and inverted configuration as well as normal horizontal lettering. A "carriage return" is included, as well as a back–space and reduced spacing before and after a narrow character, using only one keystroke. You also have a choice of oblique or rectangular characters.

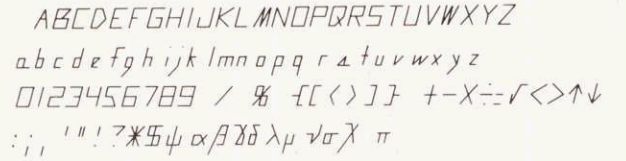
PRINCIPLE OF CHARACTER CODES:

The character codes are formed simply by typing the shape of the character on the digit keys; thus:

7	8	9
4	5	6
1	2	3
0		

M = 17593
E = 9745413
R = 1796453

Pressing a digit twice in succession causes the pen to lift, e.g., Q = 179621153 or 217962253, K = 177943, etc.



Editor's Note:

The HP 9101A Extended Memory library contains all variations of this program in one, and allows plotting up to 181 characters automatically in the sequence they are entered into the 9101A.

TO RECORD PROGRAMS FOR ALL FOUR WRITING DIRECTIONS:

1. Enter program and record on magnetic card, side A. Label: Horizontal

2. Make changes shown in Table I, in column 1. Record on side B. Label: Vertical ascending.

3. Re-enter side A. Make changes shown in column 2. Record on second card, side A. Label: Inverted.

4. Make changes shown in column 1. (This will give total changes listed in column 3.) Record on side B of second card. Label: Vertical descending.

TABLE I

STEP	1 ASCENDING KEY	CODE	2 INVERTED KEY	CODE	3 DESCENDING KEY	CODE
5.b			–	34	–	34
:						
6.0			–	34	–	34
:						
6.3	XEY	30			XEY	30
6.4	CHS	32			CHS	32
:						
6.b	XEY	30			XEY	30
6.c	CHS	32			CHS	32
:						
8.0			AC–	63	AC–	63
:						
8.2	XEY	30			XEY	30
8.3	CHS	32			CHS	32
:						
9.7			AC+	60	AC+	60
:						
a.8			AC+	60	AC+	60

PART NO. 09100-76011

EQUIPMENT NEEDED ☐ 9100A ☐ 9100B ☐ 9120 ☑ 9125 ☐ 9160 ☐ 9101

☑ DEGREES ☐ RADIANS ☐ FLOATING ☑ FIXED DECIMAL WHEEL AT 0 PRESS ☐☐☐ ON 9120

STEP	USER INSTRUCTIONS	DISPLAY
1.	ENTER PROGRAM for desired writing direction.	
2.	If rectangular rather than oblique characters are desired, PRESS GO TO 5–8, SWITCH to PROGRAM: PRESS CONTINUE, SWITCH RUN, PRESS END.	
3.	ENTER desired character height in mils. PRESS CONTINUE.	
4.	PRESS FMT ↑. Locate pen at desired starting point. NOTE: If you do not wish to move the plotter controls, store the desired starting coordinates manually, X→f, Y→e before continuing.	
5.	Type in first character. PRESS CONTINUE. Type next character, PRESS CONTINUE.	
6.	For word space, PRESS CONTINUE with no character entry.	
7.	To decrease space around a character, CHANGE SIGN of entry. Thus I = −82, decimal point = −2, etc.	
8.	For line spaces, ENTER character, PRESS SFL, CONTINUE. For double or triple line spaces, PRESS SFL, CONTINUE with no character entry except for last time.	
9.	To backspace, PRESS b, PRESS ACC − three times. (To increase space, ACC+.)	
10.	To use a zero as a first digit, precede with a decimal point. Zero cannot be the last digit of a character entry. Examples: Comma: .01 Semicolon: .0114	

Ray Schneider lives in Warminster township north of Philadelphia, Pennsylvania, with his wife Jessica and a young son. Ray graduated from St. Joseph's College in Philadelphia in 1966 with a BS in Engineering Physics. Since then he has worked at the Naval Air Development Center, signal processing data from magnetometer systems.

Besides programming the HP 9100, some of his other hobbies include photography, painting pictures, and playing tournament chess often enough to maintain a B rating, but he says, not enough to improve it.

Mr. Schneider is studying for a master's degree at Penn State's King of Prussia extension in the evenings.

Step	Key	Code
00	UP	27
01	.	21
02	1	01
03	6	06
04	X	36
05	YTO	40
06	b	14
07	CLR	20
08	STP	41
09	IFG	43
0a	8	10
0b	b	14
0c	X<Y	52
0d	9	11

Step	Key	Code
10	d	17
11	UP	27
12	1	01
13	X>Y	53
14	2	02
15	0	00
16	1	01
17	0	00
18	XTO	23
19	c	16
1a	DIV	35
1b	GTO	44
1c	1	01
1d	2	02

Step	Key	Code
20	CLX	37
21	X=Y	50
22	7	07
23	4	04
24	1	01
25	0	00
26	X	36
27	DN	25
28	UP	27
29	INT	64
2a	-	34
2b	YTO	40
2c	d	17
2d	UP	27

Step	Key	Code
30	YE	24
31	c	16
32	X=Y	50
33	6	06
34	a	13
35	XEY	30
36	1	01
37	-	34
38	3	03
39	DIV	35
3a	XEY	30
3b	UP	27
3c	INT	64
3d	-	34

Step	Key	Code
40	RDN	31
41	X	36
42	CLX	37
43	X>Y	53
44	EEX	26
45	+	33
46	RUP	22
47	XEY	30
48	-	34
49	.	21
4a	6	06
4b	4	04
4c	DIV	35
4d	b	14

Step	Key	Code
50	X	36
51	RUP	22
52	X	36
53	.	21
54	3	03
55	RUP	22
56	X	36
57	RDN	31
58	+	33
59	e	12
5a	RUP	22
5b	+	33
5c	f	15
5d	RUP	22

Step	Key	Code
60	+	33
61	CLX	37
62	RDN	31
63	CNT	47
64	CNT	47
65	FMT	42
66	DN	25
67	GTO	44
68	1	01
69	0	00
6a	RCL	61
6b	CNT	47
6c	CNT	47
6d	FMT	42

Step	Key	Code
70	UP	27
71	GTO	44
72	1	01
73	0	00
74	3	03
75	UP	27
76	CLX	37
77	IFG	43
78	.	21
79	5	05
7a	-	34
7b	b	14
7c	X	36
7d	DN	25

Step	Key	Code
80	AC+	60
81	RCL	61
82	CNT	47
83	CNT	47
84	FMT	42
85	UP	27
86	DN	25
87	DN	25
88	GTO	44
89	0	00
8a	8	10
8b	UP	27
8c	b	14
8d	UP	27

Step	Key	Code
90	.	21
91	1	01
92	6	06
93	DIV	35
94	CLX	37
95	XTO	23
96	f	15
97	AC-	63
98	RDN	31
99	DN	25
9a	GTO	44
9b	0	00
9c	c	16
9d	SFL	54

Step	Key	Code
a0	UP	27
a1	|Y|	55
a2	b	14
a3	UP	27
a4	2	02
a5	DIV	35
a6	CLX	37
a7	XEY	30
a8	AC-	63
a9	DN	25
aa	GTO	44
ab	1	01
ac	2	02
ad	END	46

Step	Key	Code

Storage

Plus Page

f
e
d
c
b
a
9
8
7
6
5
4
3
2
1
0

Minus Page

f
e
d
c
b
a
9
8
7
6
5
4
3
2
1
0

HEWLETT-PACKARD

PRINTER PAPER ECONOMY

You can save some time and printer paper when you want a 9120A printout of only a part of a program to verify steps. Address the calculator to an address a few steps preceding the portion in question by pressing GO TO, ( ) ( ); then press LIST on the 9120A. If you do not want to list the entire balance of the program, press STOP after the part you want is printed, otherwise the printout will continue until an END statement is reached.

Other helpful techniques you have developed will be welcomed by other HP Calculator system owners. Send them to the editor, *HP KEYBOARD*.

IT'S THE LITTLE THINGS THAT COUNT

Sometimes the obvious is not so noticeable. Did you realize that...

Your keyboard is one of the finest on the market. Each key has a positive touch or click made possible by our using Cherry Corp. gold cross switches. These keys make a slightly audible click, and also prevent switch bounce (prevent entering of more than one number or instruction per key depression). Result: A solid feel to the keyboard, plus a reduction of errors in entering data.

Many customers tell us the main reason they like the 9100 is its ease of operation and logical keyboard. They like the simplified arrangement which doesn't overwhelm the unfamiliar user with a maze of intricate keys, switches, or buttons. Result: More people can interface themselves quickly to the 9100. Compare this to other machines where only one or two people are the primary operators.

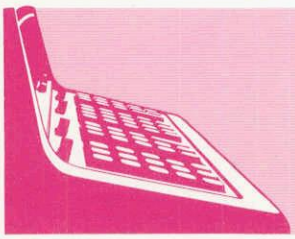
Insignificant zeroes on the display screen are automatically blanked. Result: You look at only the numbers you enter. You are not forced to look at an unnecessary string of zeroes.

"It's the little things that count."

ERRATA

An incorrect step appears in the Primary and Secondary Transmission Line Parameters program in the Winter 1970 *KEYBOARD*. Step (–)06 appears as CLX, 37. It should be UP, 27 to allow calculation of the propagation velocity.

The Programming Tip in the same issue shows a table labeled from left to right a, b, c, d. This should be d, c, b, a to correspond to the storage sequence in the program.



LINEAR PROGRAMMING

PART NO.
09100–74825

The basic ideas behind resource allocation using linear programming techniques are relatively straightforward. In fact, they are commonly taught in most introductory courses in economics, industrial engineering, etc. Unfortunately, the arithmetic involved with even simple problems is considerable.

This program uses the 9101 Extended Memory to implement the simplex method for solving linear programming problems. Students can now concentrate on setting up their problems and converting problem descriptions into precise mathematical statements. The HP Calculating System will do the arithmetic.

As an aside, the HP 9125A Plotter provides a very valuable tool for plotting graphical descriptions of the linear programming problem. This is particularly useful for beginning students just being acquainted with the technique. After seeing simple two dimensional problem statements, they can graduate to this program for solution of more realistic problems.

This program will handle up to eight constraints in eight unknowns. After entering the problem constraints, the program generates a slack variable for each unknown and uses this as the starting basic feasible solution. At each iteration the real problem variable which generates the largest increase in the profit function is introduced into the solution. A printout is provided of the problem variable introduced and the slack variable removed.

Minimization problems can be converted to maximization problems by multiplying the cost function by minus one. The program requires that the system of constraints be square, i.e., there must be as many constraint equations as there are unknowns. Dummy variables may be entered with zero coefficients to complete the array. Dummy variables entered should be assigned a profit function of -1×10^{99} to insure their being driven from the solution before the final iteration.

A number of improvements might be suggested. The current program makes no provision for removing the variable once it has been added to the solution and it is possible with certain sets of redundant constraints that the program will fail to reach the maximum. In some cases one might reduce the computation time by keeping track of the rows and columns independently, thus removing the requirement that the constraint array be square. Users with a 9100B can increase the number of unknowns to nine by storing programs six and eight on the minus page of the 9100B's memory rather than in the 9101A. Alternatively, these could be two passes on a 9100A. This will also require some appropriate changes in program constants for address calculations. All of these improvements might be suggested as student projects.

EQUATIONS USED:

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1j}x_j + \ldots + a_{1n}x_n \leq b_1$$

$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{ij}x_j + \ldots + a_{2n}x_n \leq b_2$$

$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mj}x_j + \ldots + a_{mn}x_n \leq b_m$$

$$\text{Maximize} \quad P = c_1x_1 + \ldots + c_jx_j + \ldots + c_nx_n$$

$$n = m \leq 8, \qquad x_j \geq 0 \qquad b_i \geq 0$$

SAMPLE DATA:

$$x_1 - x_2 \leq 3$$
$$x_1 + x_2 \leq 4 \quad \text{Constraints}$$
$$x_1 \leq 3$$

$$\text{Max: } 2x_1 + x_2 \quad \text{Objective function}$$

$$x_1 = 3$$
$$x_2 = 1 \quad \text{Solution}$$

$$10x_1 + 20x_2 + 8x_3 \leq 30{,}000$$
$$15x_1 + 5x_2 + 6x_3 \leq 30{,}000 \quad \text{Constraints}$$
$$10x_1 + 5x_2 + 25x_3 \leq 40{,}000$$

$$\text{Max: } 2x_1 + 3x_2 + 2.5x_3 \quad \text{Objective function}$$

$$x_1 = 1504.85$$
$$x_2 = 378.64 \quad \text{Solution}$$
$$x_3 = 922.33$$

PART NO. 09100-74825

EQUIPMENT NEEDED ☐ 9100A ☐ 9100B ☑ 9120 ☐ 9125 ☐ 9160 ☑ 9101

☑ DEGREES ☐ RADIANS ☐ FLOATING ☑ FIXED DECIMAL WHEEL AT 5 PRESS ☐ y ☐ ON 9120

STEP	USER INSTRUCTIONS	DISPLAY x	y	z
	PROGRAM LOADING			
	File protect switch: OFF			
	PRESS: END			
	PRESS: CLEAR, FMT, SET FLAG			
	ENTER PROGRAM 0			
	PRESS: 0, FMT, FMT	3	0	0
	ENTER PROGRAM 1			
	PRESS: 1, FMT, FMT	11	0	0
	ENTER PROGRAM 2			
	PRESS: 2, FMT, FMT	19	0	0
	ENTER PROGRAM 3			
	PRESS: 3, FMT, FMT	25	0	0
	ENTER PROGRAM 4			
	PRESS: 4, FMT, FMT	33	0	0
	ENTER PROGRAM 5			
	PRESS: 5, FMT, FMT	38	0	0
	ENTER PROGRAM 6			
	PRESS: 6, FMT, FMT	41	0	0
	ENTER PROGRAM 7			
	PRESS: 7, FMT, FMT	42	0	0
	ENTER PROGRAM 8			
	PRESS: 8, FMT, FMT	50	0	0
	PROGRAM EXECUTION			
1.	PRESS: 8, FMT, GO TO			
2.	PRESS: CONTINUE	1	0	0
3.	ENTER DATA: Number of variables	n		
4.	PRESS: CONTINUE	0	i	j
5.	ENTER DATA: a_{ij} (b_i when j = n + 1)	a_{ij}		
6.	Repeat instructions 4 and 5 until display reads:	0	1	0
7.	ENTER DATA	c_j		
8.	PRESS: CONTINUE	0	j	0
9.	Repeat instructions 7 and 8			
10.	Printed output will show j added to and i removed from the solution at each iteration.			
11.	After final iteration printed output will be: j, x_j			
12.	For new problem, return to instruction 1.			

Program P_0

Step	Key	Code	Step	Key	Code	Step	Key	Code	Step	Key	Code	Step	Key	Code	Step	Key	Code
00	CLR	20	10	+	33	60	FMT	42	30	3	03	00	1	01	50	UP	27	40	X	36
01	1	01	11	1	01	61	π	56	31	2	02	01	6	06	51	1	01	41	b	14
02	XEY	30	12	5	05	62	RUP	22	32	+	33	02	7	07	52	AC+	60	42	+	33
03	AC+	60	13	1	01	63	YE	24	33	f	15	03	XTO	23	53	a	13	43	DN	25
04	f	15	14	XEY	30	64	c	16	34	X<Y	52	04	b	14	54	UP	27	44	UP	27
05	UP	27	15	FMT	42	65	CLX	37	35	0	00	05	FMT	42	55	f	15	45	FMT	42
06	a	13	16	π	56	66	RUP	22	36	8	10	06	π	56	56	X<Y	52	46	π	56
07	X	36	17	AC+	60	67	FMT	42	37	d	17	07	UP	27	57	1	01	47	RUP	22
08	e	12	18	RCL	61	68	YTO	40	38	UP	27	08	1	01	58	a	13	48	-	34
09	+	33	19	FMT	42	69	DN	25	39	8	10	09	5	05	59	CLR	20	49	DN	25
0a	1	01	1a	π	56	6a	X>Y	53	3a	1	01	0a	8	10	5a	4	04	4a	XEY	30
0b	6	06	1b	XEY	30	6b	FMT	42	3b	-	34	0b	-	34	5b	FMT	42	4b	FMT	42
0c	7	07	1c	FMT	42	6c	GTO	44	3c	c	16	0c	YE	24	5c	GTO	44	4c	YTO	40
0d	+	33	1d	π	56	6d	EEX	26	3d	XEY	30	0d	d	17	5d	END	46	4d	CLX	37

Program P_4

10	1	01	20	X	36	70	9	11	40	FMT	42	10	2	02	00	1	01	50	UP	27
11	XEY	30	21	DN	25	71	9	11	41	YTO	40	11	3	03	01	XEY	30	51	1	01
12	FMT	42	22	+	33	72	XTO	23	42	CLR	20	12	2	02	02	AC+	60	52	AC+	60
13	π	56	23	1	01	73	b	14	43	2	02	13	-	34	03	RCL	61	53	a	13
14	X=Y	50	24	UP	27	74	2	02	44	3	03	14	YTO	40	04	d	17	54	UP	27
15	2	02	25	a	13	75	3	03	45	2	02	15	c	16	05	X=Y	50	55	f	15
16	3	03	26	AC+	60	76	2	02	46	UP	27	16	d	17	06	5	05	56	X<Y	52
17	CLX	37	27	XEY	30	77	UP	27	47	2	02	17	UP	27	07	9	11	57	0	00
18	UP	27	28	1	01	78	2	02	48	4	04	18	CLX	37	08	f	15	58	8	10
19	1	01	29	5	05	79	FMT	42	49	0	00	19	AC+	60	09	UP	27	59	IFG	43
1a	AC+	60	2a	1	01	7a	GTO	44	4a	AC+	60	1a	RCL	61	0a	c	16	5a	7	07
1b	f	15	2b	+	33	7b	END	46	4b	RCL	61	1b	UP	27	0b	X=Y	50	5b	3	03
1c	UP	27	2c	e	12				4c	FMT	42	1c	c	16	0c	4	04	5c	RCL	61
1d	a	13	2d	X<Y	52				4d	π	56	1d	X=Y	50	0d	d	17	5d	AC-	63

Program P_2

20	X>Y	53	30	1	01	00	d	17	50	UP	27	20	2	02	10	a	13
21	0	00	31	8	10	01	UP	27	51	b	14	21	c	16	11	X	36
22	4	04	32	1	01	02	0	00	52	X=Y	50	22	a	13	12	d	17
23	RCL	61	33	5	05	03	9	11	53	5	05	23	X	36	13	+	33
24	PNT	45	34	8	10	04	+	33	54	d	17	24	DN	25	14	1	01
25	UP	27	35	XTO	23	05	DN	25	55	CHS	32	25	+	33	15	6	06
26	2	02	36	e	12	06	XEY	30	56	XEY	30	26	b	14	16	7	07
27	3	03	37	YE	24	07	AC+	60	57	+	33	27	+	33	17	+	33
28	2	02	38	b	14	08	RCL	61	58	DIV	35	28	DN	25	18	c	16
29	+	33	39	1	01	09	FMT	42	59	e	12	29	FMT	42	19	UP	27
2a	DN	25	3a	+	33	0a	π	56	5a	FMT	42	2a	π	56	1a	a	13
2b	FMT	42	3b	YTO	40	0b	XEY	30	5b	π	56	2b	IFG	43	1b	X	36
2c	π	56	3c	b	14	0c	FMT	42	5c	X	36	2c	CHS	32	1c	d	17
2d	UP	27	3d	AC+	60	0d	π	56	5d	e	12	2d	1	01	1d	+	33

30	PNT	45	40	e	12	10	UP	27	60	FMT	42	30	UP	27	20	1	01
31	PNT	45	41	FMT	42	11	CLX	37	61	YTO	40	31	c	16	21	6	06
32	RCL	61	42	π	56	12	X>Y	53	62	1	01	32	UP	27	22	7	07
33	AC-	63	43	RUP	22	13	2	02	63	UP	27	33	a	13	23	+	33
34	1	01	44	-	34	14	9	11	64	AC+	60	34	X	36	24	RUP	22
35	+	33	45	c	16	15	DN	25	65	2	02	35	d	17	25	FMT	42
36	YTO	40	46	X>Y	53	16	DIV	35	66	4	04	36	+	33	26	π	56
37	e	12	47	5	05	17	8	10	67	0	00	37	b	14	27	RUP	22
38	-	34	48	5	05	18	UP	27	68	UP	27	38	+	33	28	FMT	42
39	a	13	49	YTO	40	19	f	15	69	a	13	39	DN	25	29	π	56
3a	X>Y	53	4a	c	16	1a	+	33	6a	+	33	3a	FMT	42	2a	DIV	35
3b	0	00	4b	RCL	61	1b	DN	25	6b	f	15	3b	π	56	2b	UP	27
3c	4	04	4c	YTO	40	1c	FMT	42	6c	X<Y	52	3c	CHS	32	2c	7	07
3d	END	46	4d	d	17	1d	YTO	40	6d	4	04	3d	DIV	35	2d	FMT	42

Program P_1

00	CLR	20	50	1	01	20	b	14	70	b	14	40	f	15	30	GTO	44
01	XTO	23	51	6	06	21	X<Y	52	71	CLR	20	41	UP	27	31	c	16
02	b	14	52	7	07	22	2	02	72	3	03	42	a	13	32	XEY	30
03	CHS	32	53	FMT	42	23	9	11	73	FMT	42	43	X	36	33	a	13
04	EEX	26	54	YTO	40	24	YTO	40	74	GTO	44	44	c	16	34	X	36
05	9	11	55	CLR	20	25	b	14	75	END	46	45	+	33	35	b	14
06	9	11	56	b	14	26	f	15				46	0	00	36	+	33
07	XTO	23	57	UP	27	27	XTO	23				47	8	10	37	DN	25
08	c	16	58	a	13	28	d	17				48	7	07	38	FMT	42
09	0	00	59	X>Y	53	29	a	13				49	+	33	39	π	56
0a	7	07	5a	0	00	2a	UP	27				4a	DN	25	3a	X	36
0b	9	11	5b	9	11	2b	1	01				4b	FMT	42	3b	f	15
0c	XEY	30	5c	d	17	2c	AC+	60				4c	YTO	40	3c	UP	27
0d	b	14	5d	UP	27	2d	2	02				4d	CLX	37	3d	a	13

Storage

Plus Page

f e d c b a 9 8 7 6 5 4 3 2 1 0

Minus Page

f e d c b a 9 8 7 6 5 4 3 2 1 0

Program P_7

Step	Key	Code
60	1	01
61	+	33
62	YTO	40
63	e	12
64	-	34
65	a	13
66	X>Y	53
67	0	00
68	3	03
69	CLR	20
6a	d	17
6b	XTO	23
6c	e	12
6d	SFL	54

Step	Key	Code
30	UP	27
31	1	01
32	+	33
33	PNT	45
34	0	00
35	8	10
36	6	06
37	+	33
38	d	17
39	UP	27
3a	PNT	45
3b	PNT	45
3c	0	00
3d	7	07

Step	Key	Code
00	0	00
01	7	07
02	8	10
03	XEY	30
04	e	12
05	+	33
06	XEY	30
07	FMT	42
08	π	56
09	XTO	23
0a	b	14
0b	FMT	42
0c	END	46

Step	Key	Code
40	UP	27
41	CLX	37
42	GTO	44
43	1	01
44	c	16
45	2	02
46	3	03
47	1	01
48	UP	27
49	f	15
4a	+	33
4b	b	14
4c	XEY	30
4d	PNT	45

Step	Key	Code

Step	Key	Code

Step	Key	Code

Program P_8

Step	Key	Code
70	GTO	44
71	0	00
72	8	10
73	CLR	20
74	1	01
75	UP	27
76	c	16
77	AC+	60
78	UP	27
79	5	05
7a	FMT	42
7b	GTO	44
7c	END	46

Step	Key	Code
40	8	10
41	+	33
42	DN	25
43	FMT	42
44	YTO	40
45	1	01
46	5	05
47	9	11
48	XTO	23
49	d	17
4a	1	01
4b	FMT	42
4c	GTO	44
4d	END	46

Step	Key	Code
00	CLR	20
01	2	02
02	4	04
03	8	10
04	AC+	60
05	DN	25
06	1	01
07	AC-	63
08	RCL	61
09	FMT	42
0a	YTO	40
0b	UP	27
0c	0	00
0d	7	07

Step	Key	Code
50	PNT	45
51	FMT	42
52	YTO	40
53	a	13
54	UP	27
55	1	01
56	+	33
57	AC+	60
58	XTO	23
59	e	12
5a	f	15
5b	X<Y	52
5c	1	01
5d	d	17

Step	Key	Code

Step	Key	Code

Step	Key	Code

Program P_5

Step	Key	Code
00	a	13
01	X	36
02	d	17
03	+	33
04	1	01
05	6	06
06	7	07
07	+	33
08	DN	25
09	FMT	42
0a	π	56
0b	RDN	31
0c	7	07
0d	FMT	42

Step	Key	Code
10	GTO	44
11	f	15
12	XEY	30
13	a	13
14	X	36
15	b	14
16	+	33
17	DN	25
18	UP	27
19	FMT	42
1a	π	56
1b	RUP	22
1c	DIV	35
1d	RDN	31

Step	Key	Code
20	XEY	30
21	FMT	42
22	YTO	40
23	RCL	61
24	1	01
25	+	33
26	YTO	40
27	e	12
28	-	34
29	a	13
2a	X>Y	53
2b	0	00
2c	c	16
2d	c	16

Program P_6

Step	Key	Code
00	CLR	20
01	1	01
02	XEY	30
03	AC+	60
04	RCL	61
05	STP	41
06	CNT	47
07	UP	27
08	IFG	43
09	PNT	45
0a	PNT	45
0b	PNT	45
0c	1	01
0d	5	05

Step	Key	Code
10	8	10
11	RUP	22
12	+	33
13	CLX	37
14	RDN	31
15	FMT	42
16	YTO	40
17	RCL	61
18	1	01
19	+	33
1a	YTO	40
1b	e	12
1c	a	13
1d	X=Y	50

Step	Key	Code
20	SFL	54
21	SFL	54
22	1	01
23	-	34
24	a	13
25	X>Y	53
26	0	00
27	4	04
28	1	01
29	FMT	42
2a	GTO	44
2b	END	46

Step	Key	Code
30	6	06
31	7	07
32	+	33
33	a	13
34	RUP	22
35	X>Y	53
36	4	04
37	5	05
38	b	14
39	RUP	22
3a	PNT	45
3b	FMT	42
3c	YTO	40
3d	1	01

Step	Key	Code
60	CLR	20
61	a	13
62	UP	27
63	1	01
64	6	06
65	8	10
66	+	33
67	UP	27
68	0	00
69	7	07
6a	9	11
6b	AC+	60
6c	RCL	61
6d	FMT	42

Step	Key	Code
70	YTO	40
71	1	01
72	XEY	30
73	1	01
74	AC+	60
75	RCL	61
76	DN	25
77	X<Y	52
78	6	06
79	c	16
7a	6	06
7b	FMT	42
7c	GTO	44
7d	END	46

Step	Key	Code

Step	Key	Code

Storage

Plus Page

f	
e	
d	
c	
b	
a	
9	
8	
7	
6	
5	
4	
3	
2	
1	
0	

Minus Page

f	
e	
d	
c	
b	
a	
9	
8	
7	
6	
5	
4	
3	
2	
1	
0	

PROGRAMMING TIPS

ERROR-PROOFING PROGRAMS

Many calculator system programs afford the opportunity to include short self-checking sequences which will save operator time by alerting the operator when he is following improper procedures. Other indications of improper operation are self-evident.

The self-check for proper setting of the DEGREES/RADIANS switch published in Vol. 1, No. 3 of the *KEYBOARD* elicited several other ideas for similar self-checking sequences. Two of these suggested by Donald S. McKenzie of the F. Yeager Bridge & Culvert Co., Port Huron, Mich., can be used to light the error light in the 9100A or 9100B. They detect improper setting of the DEGREES/RADIANS switch.

Sequence when program requires DEGREES:

STEP	KEY	CODE	IN DEGREES		IN RADIANS	
			X	Y	X	Y
00	1	01	1		1	
01	↑	27	1	1	1	1
02	tan X	71	.0175	1	1.5574	1
03	X>Y	53				
04	CLX	37			0	1
05	DIV	35			Error Light	

Sequence when program requires RADIANS:

STEP	KEY	CODE	IN DEGREES		IN RADIANS	
			X	Y	X	Y
00	1	01	1		1	
01	↑	27	1	1	1	1
02	tan X	71	.0175	1	1.5574	1
03	X<Y	52				
04	CLX	37	0	1		
05	DIV	35	Error Light			

Claude Cardot of Laboratoires de Marcoussis, Marcoussis, France, whose 4-Operations program also appears in this issue, submitted the following sequence for checking the DEGREES setting:

STEP	KEY	CODE	IN RADIANS		IN DEGREES	
			X	Y	X	Y
00	π	56	π		π	
01	tan X	71	0		.05489	
02	DIV	35	Error Light			

J. G. Mott of the U.S. Naval Weapons Center, China Lake, California sent us another short sequence for checking the DEGREES/RADIANS switch setting. This does not disturb the Y or Z register contents:

FOR DEGREES:	IN DEGREES	IN RADIANS
2	2	2
cos X	+.9994	−.4162
$\sqrt{X}$	(No Light)	Error Light

FOR RADIANS:	IN DEGREES		IN RADIANS	
	X	Y	X	Y
2	2		2	
cos X	+.4162		−.4162	
CHS	−.4162		+.4162	
$\sqrt{X}$	Error Light		(No Light)	

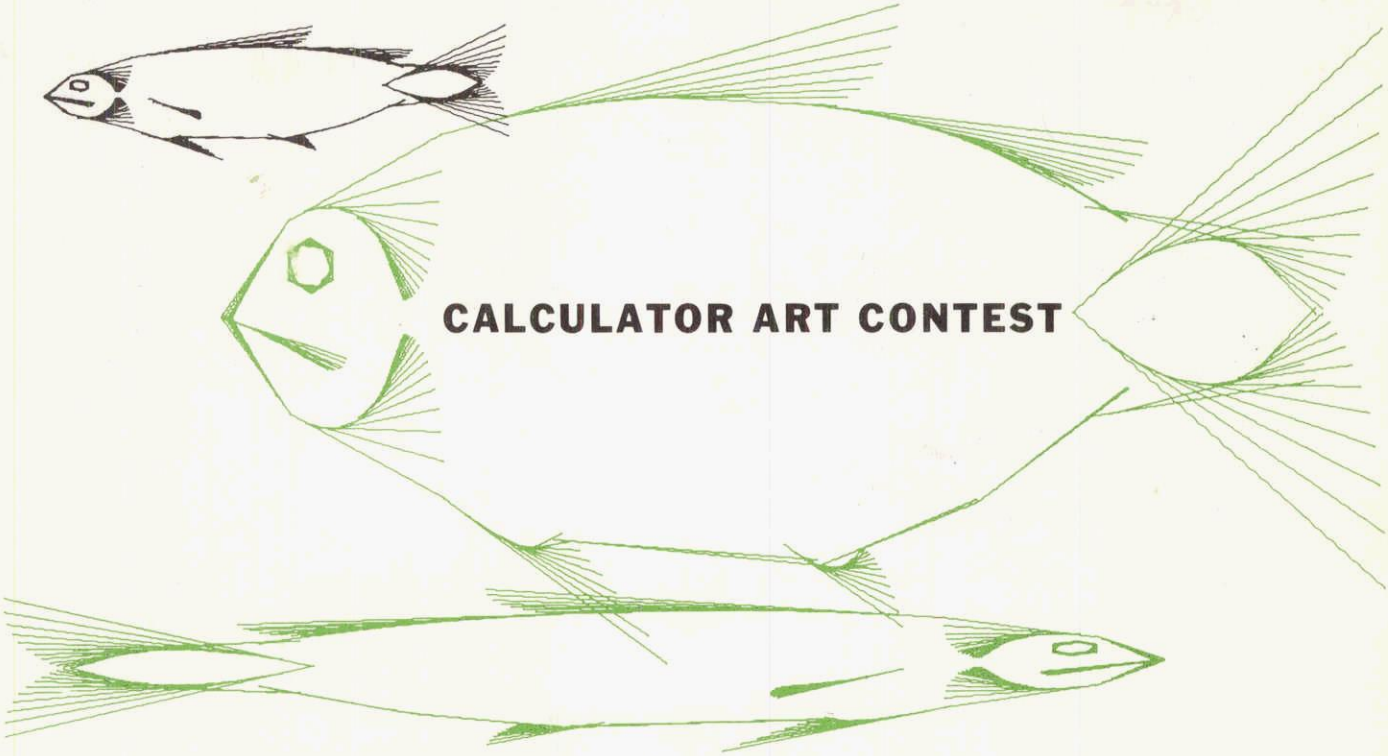
Other procedures than lighting the error light to show improper procedures may appeal to you, such as the following one suggested by John F. Ashcroft of Cobar, NSW, Australia. It produces a flashing display until the DEGREES/RADIANS switch is turned to the right position. As soon as the switch setting is corrected, the calculator shows readiness for the first data entry:

DEGREES program:			RADIANS program:		
STEP	KEY	CODE	STEP	KEY	CODE
00	CLR	20	00	CLR	20
01	PSE	57	01	PSE	57
02	π	56	02	π	56
03	sin X	70	03	sin X	70
04	X=Y	50	04	X>Y	53
05	0	00	05	0	00
06	0	00	06	0	00
07	CLR	20	07	CLR	20
08	1	01	08	1	01
09	STP	41	09	STP	41

If the PAUSE step is omitted, the display will go blank until the switch setting is corrected.

You may wish to include other error-proofing sequences in some of your programs. A useful 2-step sequence following a conditional statement which is met because of an operator error is CLX, log X. This sequence lights the error light. For example, if the program input data cannot be greater than 1.00 for a certain program, the sequence could be:

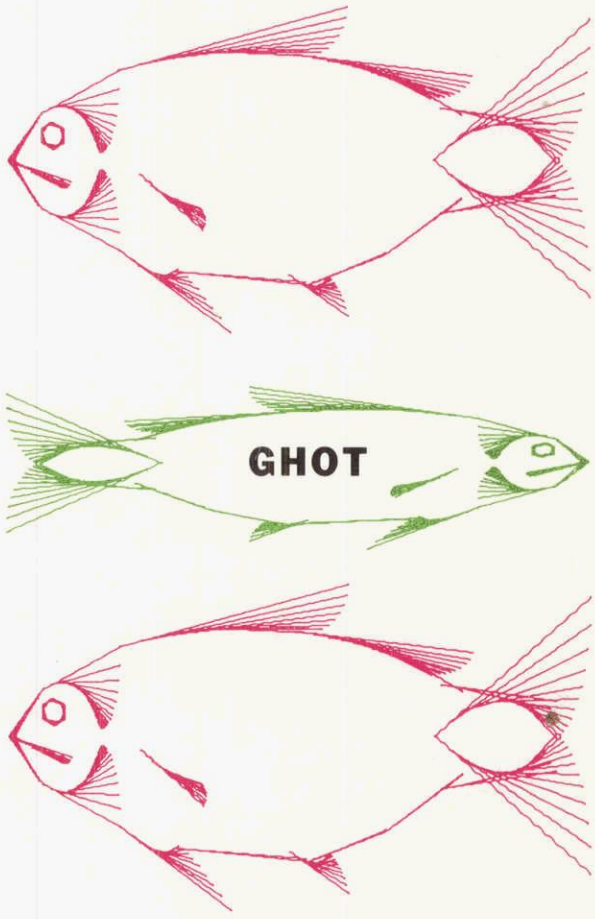
STEP	KEY	CODE	
00	CLR	20	
01	1	01	
02	↑	27	
03	STP	41	ENTRY
04	X>Y	53	
05	CLX	37	
06	log X	75	(Error Light)



CALCULATOR ART CONTEST

The HP System 9100 provides a fascinating diversion when it includes a 9125A Calculator Plotter--the ability to draw artistic designs. Some owners say this capability is exercised to the extent that their serious work is relegated to the wee hours of the morning! The design shown on this page is based on a program by Dr. Jack Walden of our R&D Laboratory. To allow calculator artists to display some of their creations, *KEYBOARD* is sponsoring a calculator art contest. Here are the rules:

1. Entries can be any pleasing designs made with an HP 9125 Calculator Plotter.
2. Entries should be in black or red, on white paper not larger than 11 x 17 inches.
3. Each entry should be accompanied by a short description of how the plot was made, and a copy of the program used for the plot.
4. First, second, and third-place winning entries, and as many runners-up as possible, will be published in the next issue of *KEYBOARD*.
5. Winning entries will also be published in *Journal of Recreational Mathematics,* published by Greenwood Periodicals, Inc., Westport, Conn.
6. Entries must be received by the *KEYBOARD* editor by December 31, 1970.



KEYBOARD

Volume 2 Number 2

APPLICATIONS INFORMATION FOR HEWLETT-PACKARD CALCULATORS

PUBLISHED AT P.O.BOX 301,LOVELAND,COLORADO 80537

Editor: A.B.SPERRY

Art Director: L.E.BRADEN