HP-65 USERS CLUB EUROPE 5 0 6 8 5 Program Description I

Program Title DETERMINANTS

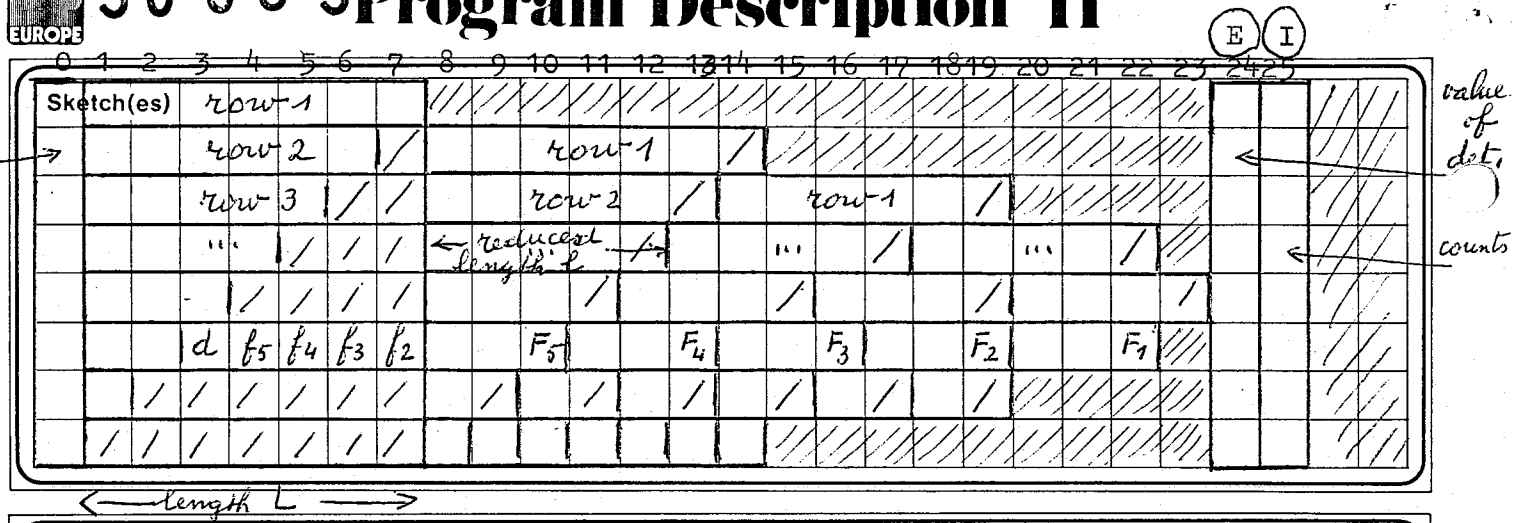
Contributor's Name RAYMOND BROECKX
Address Prieelstraat 12
City WILRIJK Country BELGIUM Postal Code 2610

Program Description, Equations, Variables

This program calculates a determinant of degree K from 2, up to 8. Its elements have to be entered only once. Since a determinant of degree 8 has 64 elements and your HP67 has only 26 storage places, the greater part of programming time is taken by re-arranging numbers already stored. Three storage places (STO 0, STO E, ST I) are not available for this work, since they have to remember: the original row-lenght and its slowly decreasing lenght after elimination of numbers already used (STO 0), the growing value of the determinant (STO E), or are used to count (ST I). Most of the time the STACK registers and LASTx are in operation too. Since at one time exactly 26 storage places are needed, one can say that this program uses the calculator's possibilities to the limit.
After first row has been entered, elements are divided by first element, so that only 7 (max) have to find a place. This first element is starting value of determinant, kept in STO E. The L=K-1 (=7) other elements are placed in STO 1,2,...,L. From now on, subroutines periodically perform the following operations: (1) move preceding rows to make room for a new row to be entered, (2) enter new row, (3) operate on new row (to the LEFT) in the diagonalisation-process, (4) divide amputated new row by first remaining element, (5) operate on old rows (to the RIGHT) in the diagonalisation-process, (6) fill gaps between remaining elements of rows, and start all over again. In step (4) the divisor also becomes a factor of the determinant-value of STO E. The sketch on next page should give you an idea of distribution of rows over storage-places...

Operating Limits and Warnings

The first element of the first row to be entered should not be zero. This and other small difficulties may be avoided by changing order of rows. A little practice is necessary here!



Sample Problem(s)

$$\begin{vmatrix} 6 & 8 \\ 7 & 10 \end{vmatrix} = 4.000000$$

$$\begin{vmatrix} 2 & 1 & 1 & 1 & 3 & 2 & 6 & 9 \\ 2 & 2 & 1 & 1 & 0 & 0 & 0 & 1 \\ 2 & 1 & 2 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 3 & 2 & 6 & 9 \\ 0 & 0 & 0 & 0 & 2 & 2 & 6 & 8 \\ 0 & 0 & 0 & 0 & 1 & 1 & 13 & 18 \\ 0 & 0 & 0 & 0 & 1 & 1 & 7 & 10 \end{vmatrix} = 8.000000 \quad (27 \text{ minutes})$$

BOTH EXAMPLES ARE EASY TO CONTROL BY HAND (EVEN FASTER THAN MACHINE!)
NOW TAKE MORE DIFFICULT NUMBERS (such as PI or other irrationals, or even two-decimal-numbers) AND TRY DOING THIS BY HAND...

Solution(s)

Reference(s)

1 DETERMINANTS 2

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	ENTER PROGRAM			
2	ENTER DEGREE K OF DETERMINANT	K	A	1
3	ENTER ONE BY ONE THE ELEMENTS a_i OF THE ROW n, THE NUMBER OF WHICH IS DISPLAYED	a_1	RS	a_1
		a_2	RS	a_1
		...		
		a_{K-1}	RS	a_1
		a_K	RS	n
4	REPEAT STEP 3 UNTIL ALL ROWS HAVE BEEN USED. AFTER EACH ROW, YOU'LL HAVE TO WAIT SOME TIME. AFTER LAST ROW, VALUE OF DETERMINANT IS SHOWN (IN DISP 6), INSTEAD OF NUMBER OF NEXT ROW.			D

USERS'CLUB EUROPE

50685 Program Listing I

STEP	KEY ENTRY	KEY CODE	COMMENTS
0011	*LBL A		Enter k here
2	DSPO		
3	1		
4	-		
5	STOO		L=k-1, also starting value of l
6	STI		i=L
7	1		
8	0		
9	:		
0101 0	STO+0		$l+\frac{L}{10}$ (two numbers kept in sto 0)
11	1		n
12	RS		enter a_1 (of first row)
13	STOE		det. ($a_1 \neq 0$!)
14	*LBL a		
15	RS		ent. next a_i
16	STOi		
17	↓		
18	STO:i		a_i / a_1 in sto i
19	DSZ		
0202 0	GTOa		
21	*LBL 0		make room for next row
22	C		p=L-l+1 (number
23	GSBc		l of preceding rows)
24	x		
25	STI		i=lp
26	*LBL 1		
27	RCLi		
28	GSBb		Δi=L (number of places
29	GSBd		to move)
0303 0	STOi		
31	GSBe		
32	DSZ		
33	GTO1		
34	GSBb		enter new row
35	STI		i=L
36	C		
37	1		
38	+		n=L-l+2
39	RS		enter a_1 (at the same
0404 0	*LBL 2		time factor f_1)
41	RS		enter next a_i
42	STO i		
43	↓		preserve f_1 in stack
44	DSZ		
45	GTO2		
46	C		"left" operation
47	GSBc		
48	STI		i=l
49	↓		f_i and p to be kept in stack
0505 0	*LBL 3		
51	B		adds Δi=pl+L-l to i
52	RCLi		x
53	GSBe		restores i
54	⇌		
55	x		Δx=fx
56	STO-i		x→x-fx
57	↓		
58	LSTx		
59	⇌		restores f_i and p in stack
0606 0	DSZ		
61	GTO3		
62	1		
63	-		p→p-1
64	ENT		
65	=0		if p=0 go to LBL 0
66	GTOO		
67	GSBc		otherwise:
68	STI		i=l
69	↓		
0707 0	GSBd		i=l+p
71	RCLi		x is new f_i
72	⇌		p
73	GSBe		restore i=l
74	GTO3		
75	*LBL 0		division routine
76	GSBc		
77	STI		i=l
78	1		
79	CFO		
0808 0	=		
81	SFO		SFO if l=1
82	RCLE		
83	RCLi		det→d x det
84	x		
85	STOE		
86	DSP6		
87	F?O		if FlO is set show det.
88	RS		
89	DSPO		otherwise:
0909 0	RCLi		d
91	DSZ		
92	*LBL 4		
93	STO:i		divide by d
94	DSZ		
95	GTO4		
96	C		"right" operation
97	ENT		p=L-l+1 to be kept
98	*LBL 5		in stack
99	GSBb		
1001 00	⇌		
101	GSBc		
102	x		
103	+		
104	STI		i=L+pl
105	RCLi		
106	STOD		factor F_i in sto D which
107	GSBc		is now free
108	STI		i=l
109	DSZ		i=l-1
1110	↑		p kept in stack
111	*LBL 6		
112	RCLi		

REGISTERS

0 l.L	1 ←	2	3 of	4	5	6	7	8	9
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
A	B		C		D →	E det.		I counts	

50685

Program Listing II

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	RCLD		
114	x		$\Delta x = xF$
115	⇌		
	B		$\Delta i = L-l+pl$ added to i
	STO-i		$x \to x-\Delta x$
118	GSBe		old i restored
119	⇌		p
120	DSZ		
121	GTO6		
122	1		
123	-		$p \to p-1$
124	ENT		
125	≠0		if $p \neq 0$ go to LBL 5
126	GTO5		
127	C		otherwise start filling small gaps
128	STOD		
129	*LBL 7		$p = L-l+1$ in sto D which is now free
130	RCLD		
131	1		
132	-		
133	STOD		$p \to p-1$
134	=0		if $p=0$, go to LBL 0
135	GTO0		
136	GSBb		
137	GSBc		
138	RCLD		
139	x		
140	+		
141	STI		$i = pl+L$
	*LBL 8		
	ISZ		
144	RCLi		
145	DSZ		
146	STOi		
147	ISZ		
148	RCI		
149	RCLD		
150	-		$i-p$
151	D		$g_o = l(L-l+2)-1$
152	>		
153	GTO8		if $g_o > i-p$, go to LBL 8
154	GTO7		otherwise to LBL 7
155	*LBL 0		filling last big gap
156	D		
157	STOC		g_o in stoC which is now free
158	GSBb		
159	STI		$i = L$
160	*LBL 9		
161	ISZ		
162	RCLi		
163	C		
164	CHS		$\Delta i = -p = -(L-l+1)$
165	GSBd		
166	STOi		
167	LSTx		old i restored
168	STI		

STEP	KEY ENTRY	KEY CODE	COMMENTS
169	RCLC		
170	>		if $g_o > i$, go to LBL 9
171	GTO9		
172	1		otherwise $l \to l-1$
173	STO-0		
174	GTO0		and start all over again
175	*LBL C		Calculates $p = L-l+1$
176	GSBb		
177	GSB∅c		
178	-		
179	1		
180	+		
181	RTN		
182	*LBL b		Calculates L
183	RCL0		
184	FRAC		
185	1		
186	0		
187	x		
188	RTN		
189	*LBL c		Calculates l
190	RCL0		
191	INT		
192	RTN		
193	*LBL B		Calculates
194	C		$\Delta i = L-l+pl$ and
195	1		keeps p in stack
196	-		
197	⇌		
198	GSBc		
199	⇌		
200	x		
201	LSTx		
202	↓		
203	+		
204	*LBL d		adds Δi to i
205	RCI		
206	+		
207	STI		
208	↓		
209	RTN		
210	*LBL e		restores old i
211	LSTX		
212	STI		
213	↓		
214	RTN		
215	*LBL D		Calculates
216	C		$g_o = l(L-l+2)-1$
217	1		
218	+		
219	GSBc		
220	x		
221	1		
222	-		
223	RTN		

LABELS

A used	B used	C used	D used	E	
a used	b used	c used	d used	e used	
0 used	1 used	2 used	3 used	4 used	
5 used	6 used	7 used	8 used	9 used	

FLAGS

0 used	1
2	3

SET STATUS

FLAGS			TRIG	DISP	
	ON	OFF			
0	☐	☒	DEG ☒	FIX ☒	
1	☐	☒	GRAD ☐	SCI ☐	
2	☐	☒	RAD ☐	ENG ☐	
3	☐	☒		n 0	