



50555

Program Description I

Page 1 of 7

Program Title PIAL editorContributor's Name P.C.SchmaleAddress Roland Holstlaan 152aCity DelftCountry Netherlands Postal Code 2204

Program Description, Equations, Variables The PIAL-program you enter is represented by two-digit codes in the DATA-registers. The editing features work almost the same as those of the HP67 itself. To start the editor press A. Display shows 51. For storing the PIAL-program, the addresses 3.0, 3.2, 3.4until 24.8 are available. Normally the display shows you the address last used and the code for the PIAL-symbol that is stored there. The next entry is then stored on the next address. So if you want your program to begin at 3.0 you must go to 2.8. If you now press 2.8, A, you will see the entering loop start. During each PAUSE you can enter one PIAL-symbol. Every time the Editor comes back to the entering loop, you see the code for the just entered PIAL-symbol and the address where it is stored. You enter the PIAL-program in the same way as with the DIRECT PIAL COMPILER, only the condition-variables are entered differently (see USER INSTRUCTIONS). An address on which you want to store a PIAL-symbol must contain zeros before storing. So you must clear all registers before entering a new program. When you want to change something in an existing PIAL-program you can use the editing features: With SST, BST and GTO you can go to any desired part of the program. If you want to replace the displayed symbol by another, press CL STEP and enter the other symbol. If the displayed symbol must be removed and the following symbols shifted one step back, press DEL. If you want to insert a symbol after the displayed symbol, use INS. So, unlike the HP67, there is no automatic insert. A list of the meanings of the PIAL-symbol codes is given under "Sketches". The format of the display code is as follows:

12.40075
address (12.4) symbol code (75 = FOR)

Operating Limits and Warnings Don't disturb the contents of the stack after you pressed A and the display shows 51 (except when you want to use GTO or CL STEP). Trying to enter a not-PIAL-symbol results in Error. Other syntactic errors are not detected!

N.B. The INDIRECT PIAL COMPILER recognizes codes 00 and 50 as compiler-stop commands. So don't worry if the EDITOR places code 50 at the end of a PIAL-program. (This happens when you use DEL and the PIAL-program extends to address 24.6 or 24.8).

This program has been verified only with respect to the numerical example given in *Program Description II*. User accepts and uses this program material AT HIS OWN RISK, in reliance solely upon his own inspection of the program material and without reliance upon any representation or description concerning the program material.

NEITHER HP NOR THE CONTRIBUTOR MAKES ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND WITH REGARD TO THIS PROGRAM MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NEITHER HP NOR THE CONTRIBUTOR SHALL BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF THE FURNISHING, USE OR PERFORMANCE OF THIS PROGRAM MATERIAL.

Program Description II

Sketch(es)	code/meaning	code/meaning	code/meaning
	00 compiler stop	53 :=	90 SIN
	01 +	54 CONST	91 COS
	02 -	61 >	92 TAN
	03 x	62 =	93 e↑
	04 /	63 ≠	94 LN
	1a digit a	70 STOP	95 Pi
	20 PRINT	75 FOR	96 10↑
	30 NEXT	76 SKIP	97 INT
	50 compiler stop	87 ABS	98 CHS
	51 (	88 ↑	99 √
	52) ;	89 ARCTAN	

Sample Problem(s) 1) Enter the following PIAL-program, beginning at address 3.0 . This program is the PIAL version of the program on page 228 of the HP-97 Owner's Handbook (Using the I-register for indirect control, problem 3).

1:= CONST 1; 2:= STOP;

FOR

PRINT := 1/2; 1:= 1+2; PRINT := 2/1; 2:= 1+2;

NEXT

Record the edited PIAL-program on a DATA-card for later use with the INDIRECT PIAL COMPILER.

(The program prints the quotients of successive terms of a Fibonacci series, which converge to the golden ratio 0.618033989)

Solution(s) 1) Required keystrokes: (a comma means wait until display is steady or PAUSE occurs)

f CL REG f P<S f CL REG A, 2.8 A, 1, A, D, A,
E, 1, A, C, 2, A, D, f d, A, C, f c, f a, A,
D, 1, /, 2, A, C, 1, A, D, 1, +, 2, A, C,
f a, A, D, 2, /, 1, A, C, 2, A, D, 1, +, 2,
A, C, f b, R/S.

The contents of $R_3 - R_9$ should now be:

1.153541152	3
1.253705275	4
2.053110412	5
5.211531101	6
1.252205312	7
0.411521253	8
1.101125230	9

Reference(s)

Sketch(es)

Sample Problem(s) 2) Change the program of 1) to the following using the editing features:

1 := CONST 1 ; 2 := STOP ;

FOR

3 := 1 / 2 ; 1 := 1 + 2 ; 4 := 2 / 1 ;

≠ 3 4

2 := 1 + 2 ;

NEXT

PRINT := 2 ; PRINT := 1 ; PRINT := 4 ;

STOP

Also record this edited PIAL-program on a DATA-card.

(Now the program runs until the golden ratio is found, then prints the two numbers that give the golden ratio, and finally prints the golden ratio decimally)

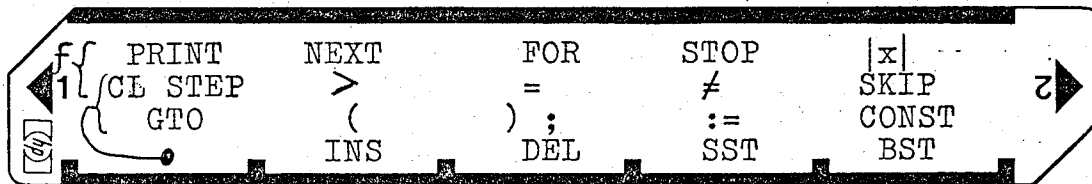
2)
Solution(s) Required keystrokes: (find out what editing functions are used)
A, 5.0 A, A, fa, 3, A, 7.4 A, A, f a, 4, D,
D, D, D, D, B, D, now you see that the
address 8.6 is free, E, A, f d, B, 3, B, 4,
A, 10.4 A, f a, A, D, 2, A, C, f a, A, D,
1, A, C, f a, A, D, 4, A, C, f d, R/S.

1.15354152	3
1.253705275	4
1.353110412	5
5.211531101	6
1.252145312	7
0.411526313	8
1.412531101	9

Now the contents of $R_3 - R_{13}$ should look like this:

1.252302053	10
1.252205311	11
5.220531452	12
7.000000000	13

Reference(s) For a description of the PIAL language, see program descr. I of the program DIRECT PIAL COMPILER. To convert an edited PIAL-program into an RPN-oriented code you need the program INDIRECT PIAL COMPILER. This is the same code as generated by the DIRECT PIAL COMPILER. Execution of this code is possible with the PIAL EXECUTOR.



STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Load side 1 and side 2			
2	Initialize:			
	For a new PIAL-program clear the registers		f CL REG	
			f PZS	
			f CL REG	
	For changing an old PIAL-program load the program by entering a DATA-card.			
3	Start entering loop		A	51
	Enter begin address - 0.2	add _{beg} 0.2	A	
4	During each PAUSE you can:			
	I. Enter one of the following symbols:			
	a) CONST		A E	
	b) any digit that occurs in a PIAL-program, including a condition var. 0 thr.9			
	c) :=		A D	
	d) (		A B	
	e))		A C	
	f) ;		A C	
	g) PRINT		f a	
	h) STOP		f d	
	i) an operator:			
	for ABS you use x		f e	
	for ↑ you use		y ^x	
	for e↑ you use		e ^x	
	for 10↑ you use		10 ^x	
	for √ you use		√x	
	for the other operators you use the corresponding keys on the keyboard.			
	j) FOR		f c	
	k) NEXT		f b	
	l) >		A f b	
	m) =		A f c	
	n) ≠		A f d	
	o) SKIP		A f e	
	N.B. A "-" between two keystrokes means you should wait until display shows 51.			
	-Continued on next page-			

[illegible]

REGISTRATION

⁰ address - register	¹ display - register	² store result of key pressed	³	⁴ P R O G R A M	⁵	⁶ R E G I S T E R	⁷	⁸ S	⁹ P R O -
^{S⁰} G R A M	^{S¹} R	^{S²} E G I	^{S³} S T E R S	^{S⁴}	^{S⁵} P R O G R A M	^{S⁶}	^{S⁷} R E G I S T E R S	^{S⁸}	^{S⁹}
^A P R O G R	^B A M	^C R E G I S T	^D E R	^E S	^F	^G	^H	^I indirect control.	

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
113	SF0	16 21 00	DEL	169	+	-55	
114	GT08	22 08		170	*LBLc	21 16 13	=
115	*LBLB	21 12	INS	171	+	-55	
116	SF3	16 21 03	internal SST	172	*LBLb	21 16 12	>
117	GSBD	23 14		173	7	07	
118	0	00		174	+	-55	
119	*LBL8	21 08		175	*LBLE	21 15	CONST
120	F0?	16 23 00		176	+	-55	
121	GSB9	23 09		177	*LBLD	21 14	
122	GSB6	23 06		178	+	-55	
123	-	-45		179	*LBLC	21 13	
124	LSTX	16-63		180	+	-55	
125	X*Y	-41		181	*LBLB	21 12	
126	GSB7	23 07		182	GT03	22 03	
127	.	-62		183	*LBLa	21 16 11	
128	2	02		184	GSB6	23 06	CL STEP
129	ST+0	35-55 00		185	CHS	-22	
130	R1	16-31	end of program?	186	GSB7	23 07	
131	X#0?	16-42		187	GT0E	22 15	
132	GT08	22 08		188	*LBLe	21 16 15	SKIP
133	CF0	16 22 00		189	7	07	
134	.	-62	deduct starting	190	6	06	
135	1	01	address from display-	191	GT03	22 03	
136	ST=1	35-24 01	register and store	192	*LBL6	21 06	
137	RCL1	36 01	this in address register	193	RCL0	36 00	
138	INT	16 34		194	F2?	16 23 02	recall on the
139	x	-35		195	+	-55	address pointed at
140	ST00	35 00		196	STOI	35 46	by the address-register
141	GT0E	22 15	backstep	197	FRC	16 44	
142	*LBL9	21 09		198	1	01	
143	SF2	16 21 02	recall one address	199	0	00	
144	.	-62	further than add-	200	x	-35	
145	2	02	register tells.	201	1	01	
146	GT06	22 06		202	-	-45	
147	*LBLD	21 14		203	10*	16 33	
148	SF2	16 21 02	SST	204	RCLi	36 45	
149	*LBLE	21 15	BST	205	x	-35	
150	.	-62		206	FRC	16 44	
151	2	02		207	EEX	-23	
152	F2?	16 23 02		208	2	02	
153	CHS	-22		209	x	-35	
154	ST-0	35-45 00		210	INT	16 34	
155	GT01	22 01		211	RTN	24	
156	*LBLA	21 11	set up stack for	212	*LBL7	21 07	
157	1	01	efficient code -	213	RCL0	36 00	
158	ENT↑	-21	generation and halt	214	STOI	35 46	
159	ENT↑	-21	to except 2nd part of	215	FRC	16 44	
160	ENT↑	-21	instruction.	216	1	01	
161	5	05		217	0	00	
162	1	01		218	x	-35	
163	R/S	51		219	1	01	
164	*LBLA	21 11	GTO	220	+	-55	
165	ST00	35 00		221	10*	16 33	
166	CF3	16 22 03		222	÷	-24	
167	GT01	22 01		223	ST+i	35-55 45	
168	*LBLd	21 16 14		224	RTN	24	

LABELS				FLAGS		SET STATUS		
A	B	C	D	E	F	FLAGS	TRIG	DISP
GTO	INS	DEL	SST	BST	CONST	DEL		
PRINT	NEXT	FOR	STOP	TX	SKIP	1		
CL STEP	>	=	≠			ON OFF		
0 entering-loop	1 display-setup	2 digit entry	3 store new code	4	2 used	0 ☐ ☒	DEG ☒	FIX ☒
55 junctions-loop	6 recall	7 store	8 used	9 recall to 0,2	3 used	1 ☐ ☒	GRAD ☐	SCI ☐
						2 ☐ ☒	RAD ☐	ENG ☐
						3 ☐ ☒		n 5