

Program Title PIAL executerContributor's Name P.C. SchmaleAddress Roland Holstlaan 152aCity Delft Country Netherlands Postal Code 2204

Program Description, Equations, Variables This program (hereafter called Executer) provides the software needed to execute a compiled PIAL-program (hereafter called "program"), or in fact numerically coded instructions generated in any other way. Some aspects of the operation of the PIAL-Executer will be emphasized here:

1) Register occupation

The Executer itself uses only 3 data-registers: R_D and R_P in which the Y and X registers are copied, and R_0 , used as program counter and return address memory. R_1 through R_9 can be used by the program for variable-storage. The other registers R_{50} - R_{59} , R_A , R_B , and R_C can be used for program storage. If a program uses less than 9 variables, the free registers can be used for program storage. A certain data-program tradeoff is possible in this way.

2) Controlling the program counter

Via codes 50 and 60 the contents of the program counter can be controlled. Normally the FOR-NEXT statements take care of this. But you can also create a jump by storing directly in R_0 . Subroutines are also possible. The structure could be: call:1:=0; SKIP 4; (filler) and subroutine: NEXT..... 0:=1 + CONST 1 / CONST 2;

3) Using the Stack for storage

Except for the instructions NEXT, >xy, =xy, /xy, FOR and SKIP, the result of an assignment statement remains in the stack. You can use this fact to save steps. E.g. instead of 1:= CONST 1; 2:= CONST 1; you may program: 1:= CONST 1; 2:= ; and instead of 1:= SIN 2; 3:= 1/2; you can write: 1:= SIN 2; 3:= / 2; You can output a number in the display by programming a STOP at a time that the desired number is in the X-register (is always the result of an assignment statement).

Operating Limits and Warnings

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.



50 3 1 3

Program Description I

Page 2 of 6

Program Title _____

Contributor's Name _____

Address _____

City _____ Country _____ Postal Code _____

Program Description, Equations, Variables _____

4) selecting begin address

If the begin address of the program is 10.0 as in most cases, you can set the program counter to 10 by pressing A. Else you must store the begin address in R₀.

5) Single Stepping through the program execution

Normally you start execution by pressing C. If you press B however, only one program step at a time is executed. In the display you see during PAUSES: 1. the contents of the program counter; 2. (part of) the instruction in this location, and 1 and 2 again for a two digit code, and 1 and 2 several times during a search for NEXT. The steady display gives the result of an assignment statement, or some number generated by a jump instruction. With steady display you can always choose between S.STEP or RUN, also when the program halted on a STOP-statement.

6) The meaning of the instruction codes is given below:

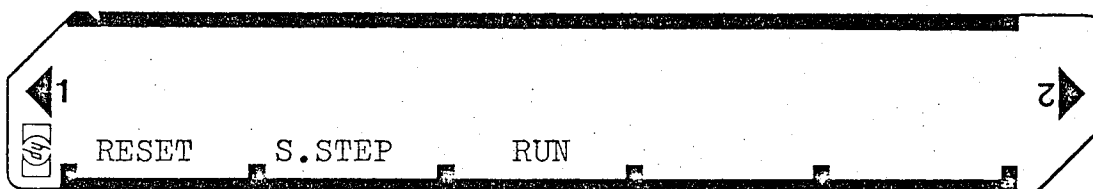
code/meaning	code/meaning	code/meaning
0 NEXT	80 STOP	90 SIN
1 +	81 >	91 COS
2 -	82 =	92 TAN
3 x	83 ≠	93 e ^x
4 /	84 PRINT	94 LN
5a STO a	85 FOR	95 Pi
6a RCL a	86 SKIP	96 10 ^x
7a CONST a	87 ABS	97 INT
a stands for 0 thr. 9	88 y ^x	98 CHS
	89 ARCTAN	99 √x

Operating Limits and Warnings _____

Execution time is much longer than that of a corresponding hp-97 program, so don't think too soon that the program is in an endless loop. Use S.STEP in cases of doubt.

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.

[illegible]

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LELA	21 11	RESET program-counter to 10.0.	057	GSEb	23 16 12	
002	GSEe	23 16 15		058	RCLi	36 45	
003	STO0	35 00		059	GT0C	22 13	
004	RTN	24		060	*LBL7	21 07	CONST a
005	*LELC	21 13		061	GSEa	23 16 11	
006	FO?	16 23 00	test if in SSTEP mode and clear SSTEP mode.	062	GSEb	23 16 12	
007	SF2	16 21 02		063	RCLi	36 46	
008	SFO	16 22 00		064	GT0C	22 13	
009	SF?	16 23 02		065	*LBL9	21 09	gots codes beginning with 9.
010	R/S	51	stop if in SSTEP mode	066	GSEa	23 16 11	
011	GT0C	22 13		067	GSEb	23 16 12	
012	*LELB	21 12	set SSTEP mode	068	GT0C	22 16 13	
013	SFO	16 21 00		069	*LBL8	21 08	codes beginning with 8
014	*LBLC	21 13		070	GSEa	23 16 11	
015	STOE	35 15	save stack's contents.	071	GSEb	23 16 12	
016	R↓	-31		072	GT0i	22 45	
017	STOD	35 14		073	*LBL0	21 00	STOP
018	GSEa	23 16 11	goto label determined by digit on pr.counter address	074	R/S	51	
019	GT0i	22 45		075	GT0C	22 13	
020	*LELO	21 00	NEXT	076	*LBL1	21 01	> xy
021	GSEb	23 16 15	place address at bottom of return memory stack in program counter and lower return address memory stack.	077	GSEb	23 15	
022	STx0	35-35 00		078	X<Y?	16-35	
023	RCL0	36 00		079	GT0d	22 16 14	
024	FRC	16 44		080	GT0C	22 13	
025	STO0	35 00		081	*LBL2	21 02	= xy
026	EEX	-27		082	GSEb	23 15	
027	3	03		083	X=Y?	16-32	
028	x	-35		084	GT0d	22 16 14	
029	INT	16 34		085	GT0C	22 13	
030	ST+0	35-55 00		086	*LBL3	21 03	
031	GSEb	23 16 15		087	GSEb	23 15	± xy
032	ST+0	35-24 00		088	X=Y?	16-32	
033	GT0C	22 13		089	GT0C	22 13	
034	*LBL1	21 01		090	*LBL6	21 06	SKIP
035	GSEb	23 16 12	+	091	*LBLd	21 16 14	
036	+	-55		092	GSEa	23 16 11	found a NEXT?
037	GT0C	22 13		093	X=0?	16-43	
038	*LBL2	21 02	-	094	GT0D	22 14	
039	GSEb	23 16 12		095	5	05	
040	-	-45		096	X>Y?	16-34	one digit code?
041	GT0C	22 13		097	GT0d	22 16 14	
042	*LBL3	21 03	x	098	.	-62	
043	GSEb	23 16 12		099	1	01	
044	x	-35		100	ST+0	35-55 00	
045	GT0C	22 13		101	GT0d	22 16 14	
046	*LBL4	21 04	/	102	*LBLE	21 15	place [x] in X register and [y] in Y register if x=y, make Y register = 0.
047	GSEb	23 16 12		103	GSEa	23 16 11	
048	÷	-24		104	RCLi	36 45	
049	GT0C	22 13		105	RCLi	36 46	
050	*LBL5	21 05		106	R↑	16-31	
051	GSEa	23 16 11	STO a	107	GSEa	23 16 11	
052	GSEb	23 16 12		108	X=Y?	16-33	
053	ST0i	35 45		109	SF2	16 21 02	
054	GT0C	22 13		110	R↓	-31	
055	*LBL6	21 06		111	R↓	-31	
056	GSEa	23 16 11	RCL a	112	RCLi	36 45	

REGISTERS

COUNTER	1	2	3	4	5	6	7	8	9
RETURN ADDR	VAR	ABL	ES		VAR	ABL	ES	VAR	ABL
STACK									
S0	PR	S1	S2	S3	S4	S5	S6	S7	S8
	GRAM	REG	ISTE	RS	PROG	GRAM	REG	ISTE	RS
PROGRAM					copy Y register		copy X register		used

LABELS					FLAGS	SET STATUS			
RESET	S. STEP	RUN	used	used	S. STEP mode	FLAGS		TRIG	DISP
used	used	used	used	used	1	ON	OFF	DEG ☒	FIX ☒
STOP	1 $\frac{1}{\cos}$	2 $\frac{1}{\tan}$	3 $\frac{1}{x}$	4 $\frac{1}{\ln}$	2 used	0 ☐	☒	GRAD ☐	SCI ☐
STOP	5 $\frac{1}{\cos}$	6 $\frac{1}{\tan}$	7 $\frac{1}{x}$	8 $\frac{1}{\ln}$		1 ☐	☒	RAD ☐	ENG ☐
STOP	10 $\frac{1}{\cos}$	11 $\frac{1}{\tan}$	12 $\frac{1}{x}$	13 $\frac{1}{\ln}$	3	2 ☐	☒		n <u>9</u>
STOP	14 $\frac{1}{\cos}$	15 $\frac{1}{\tan}$	16 $\frac{1}{x}$	17 $\frac{1}{\ln}$		3 ☐	☒		