Calculate the EPS machine accuracy of your programmable calculator

Enrico Bellincioni

Abstract

In this short note I would like to present the algorithm to calculate the error due to the encoding of floating point numbers in a personal computer or programmable calculator, this error is the simplest to identify.

History

Until the mid-1970s the large firms that built electronic calculators used proprietary patents for both hardware and software. The results obtained by the algorithms had difficulty in being compared with each other. To help all researchers in the world came the intuition of Professor William Kahan (then a young researcher himself) who began to develop algorithms for the encoding of floating point numbers to be later proposed as an international standard IEEE754. As the choice of the coding base in those years the binary base was used, subsequently since 2008 the possible bases chosen are in addition to the binary one (present today in all PC CPUs) also the decimal base (which was also used for the super DM42 programmable calculator). It is not my goal of this brief note to go into the detailed description of how this encoding is done. On the net it is possible to determine a lot of material that explains the procedure. I leave as a reference for example Wikipedia

https://en.wikipedia.org/wiki/IEEE_754

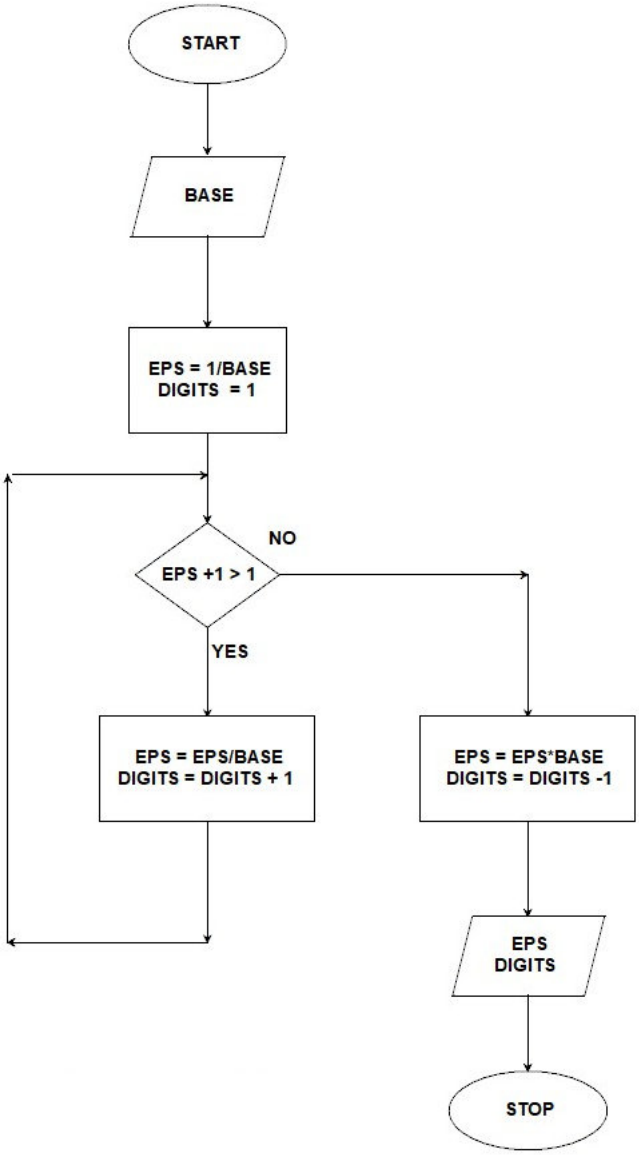
What does EPS represent ?

The term *EPS* is a shortened form of the Greek letter epsilon; it is used to represent the smallest machine value that can be added to 1.0 that gives a result distinguishable from 1.0. To deepen the concept

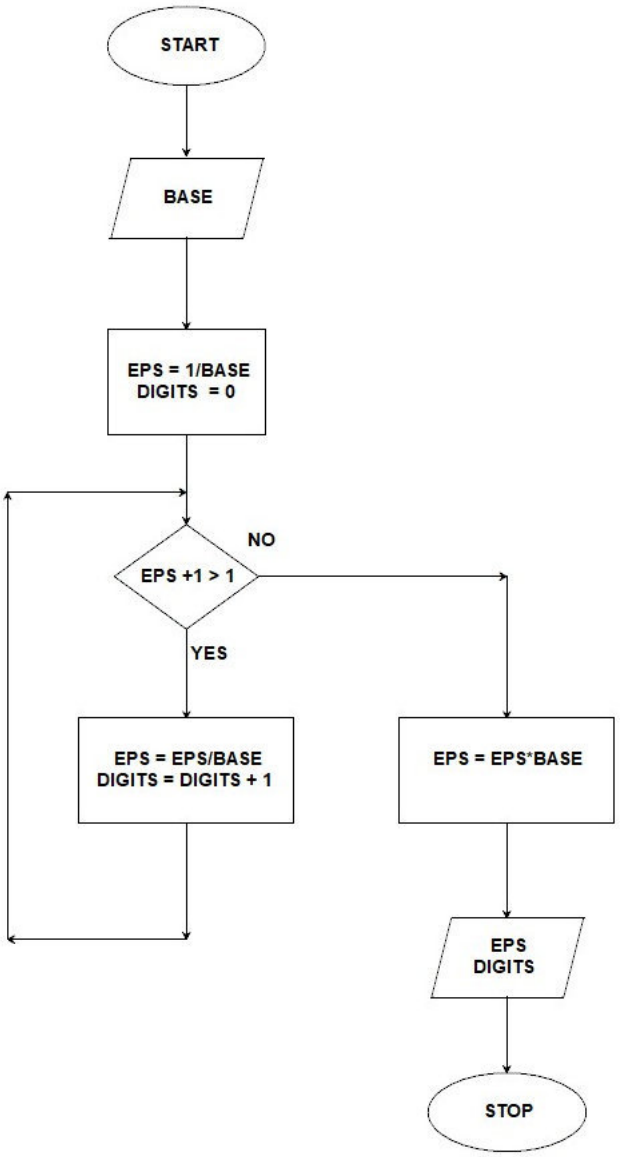
https://en.wikipedia.org/wiki/Machine_epsilon

BASE for all calculators = 10

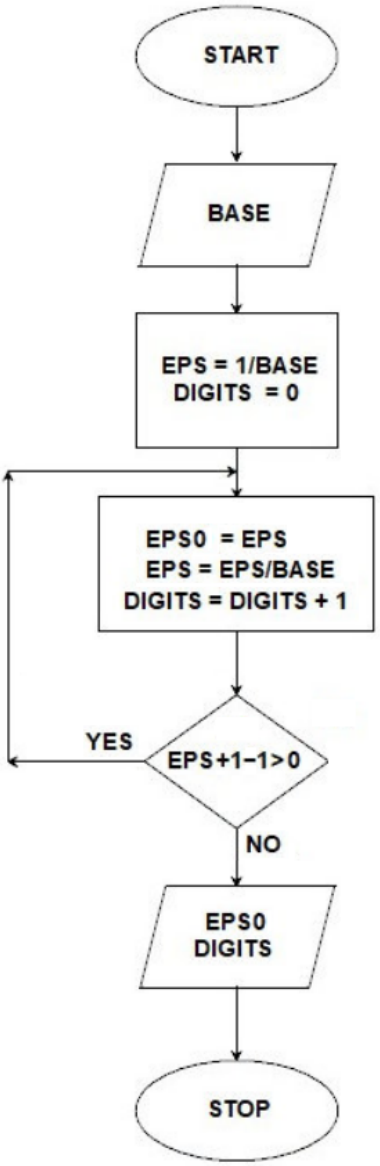
Flow chart First version



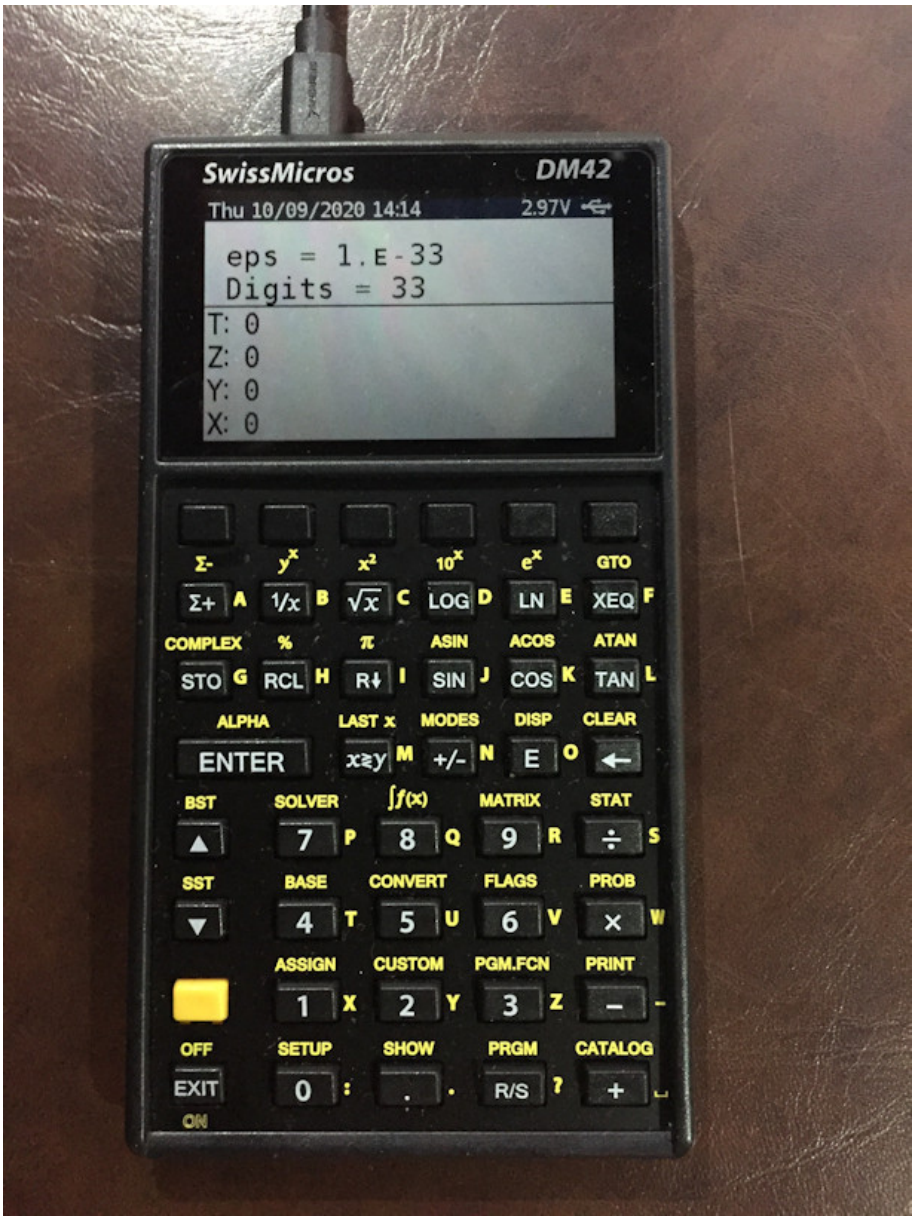
Flow Chart Second version



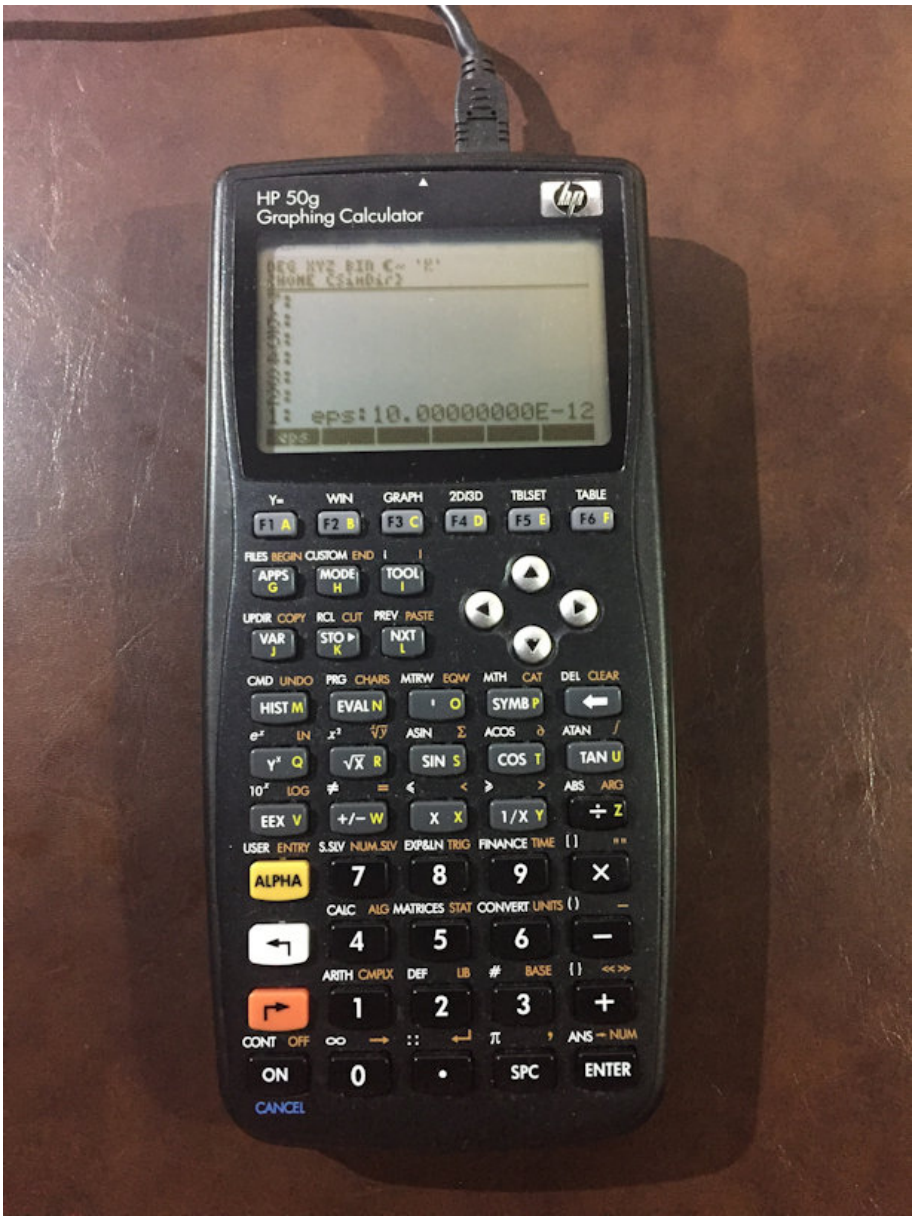
Flow Chart DM42 optimized version



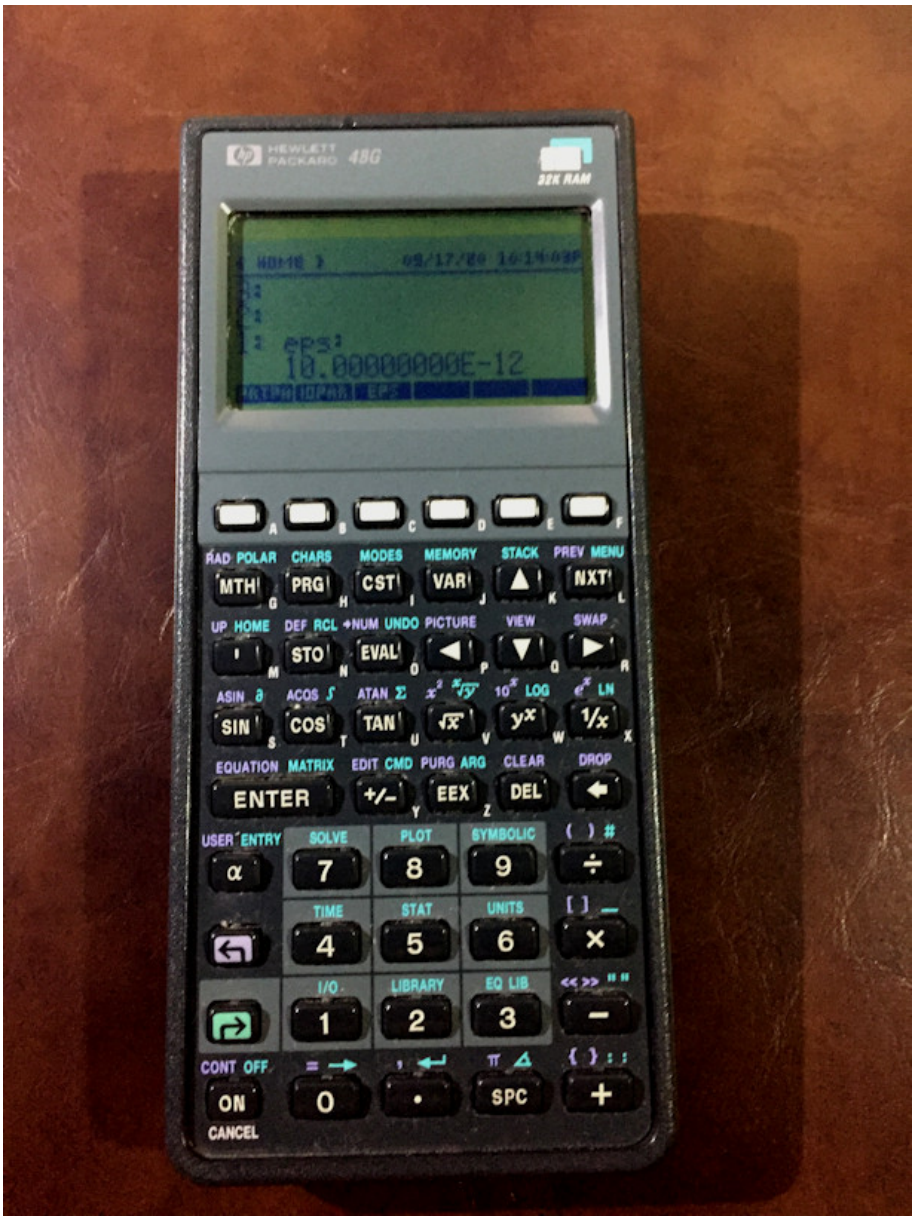
DM42


SwissMicros
DM42
Thu 10/09/2020 14:14
2.97V
eps = 1.E-33
Digits = 33
T: 0
Z: 0
Y: 0
X: 0
Σ- y^x x² 10^x e^x GTO
Σ+ A 1/x B √x C LOG D LN E XEQ F
COMPLEX % π ASIN ACOS ATAN
STO G RCL H R↓ I SIN J COS K TAN L
ALPHA LAST x MODES DISP CLEAR
ENTER x≷y M +/- N E O ←
BST SOLVER ∫f(x) MATRIX STAT
▲ 7 P 8 Q 9 R ÷ S
SST BASE CONVERT FLAGS PROB
▼ 4 T 5 U 6 V × W
ASSIGN CUSTOM PGM.FCN PRINT
1 X 2 Y 3 Z − -
OFF SETUP SHOW PRGM CATALOG
EXIT 0 : . R/S ? + ␣
ON

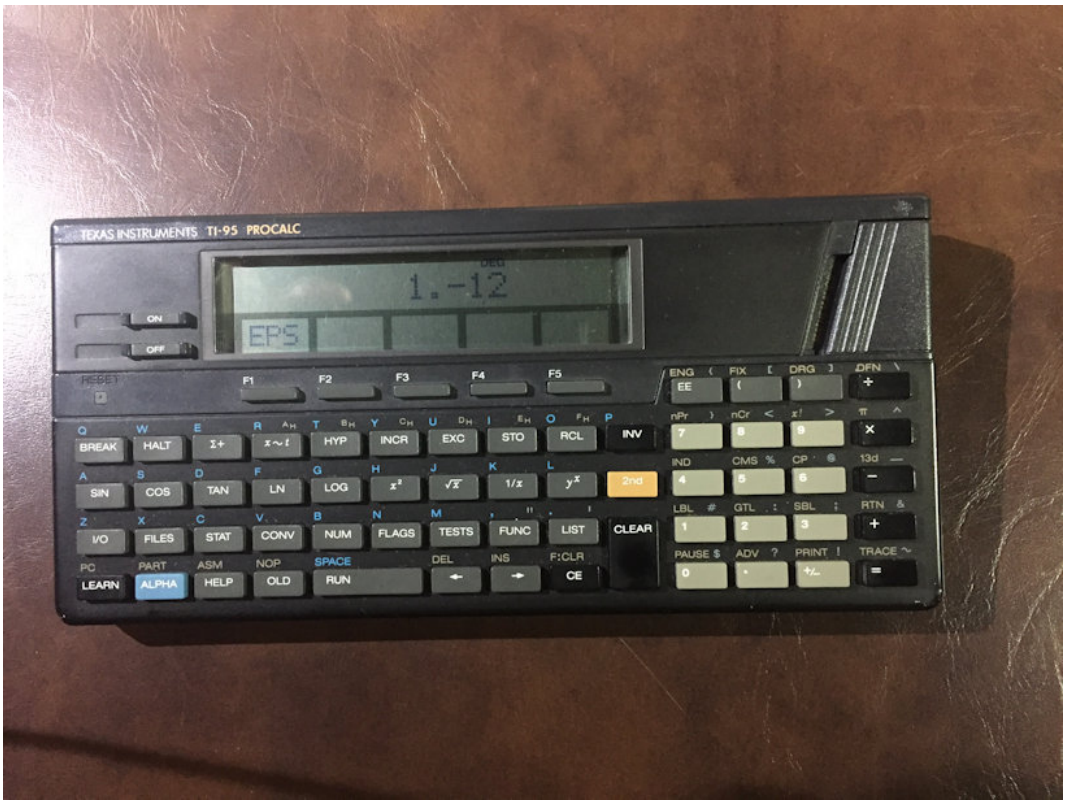
HP50G®


HP 50g
Graphing Calculator
eps:10.00000000E-12
eps
Y= WIN GRAPH 2D/3D TBLSET TABLE
F1 A F2 B F3 C F4 D F5 E F6 F
FILES BEGIN CUSTOM END
APPS G MODE H TOOL I
UPDIR COPY RCL CUT PREV PASTE
VAR J STO K NXT L
CMD UNDO PRG CHARS MTRW EQW MTH CAT DEL CLEAR
HIST M EVAL N ' O SYMB P
ASIN ACOS ATAN
SIN S COS T TAN U
ABS ARG
EEX V +/- W X X 1/X Y ÷ Z
USER ENTRY S.SLV NUM.SLV EXP&LN TRIG FINANCE TIME
ALPHA 7 8 9 ×
CALC ALG MATRICES STAT CONVERT UNITS
4 5 6 −
ARITH CMPLX DEF LIB # BASE
1 2 3 +
CONT OFF
ANS NUM
ON 0 . SPC ENTER
CANCEL

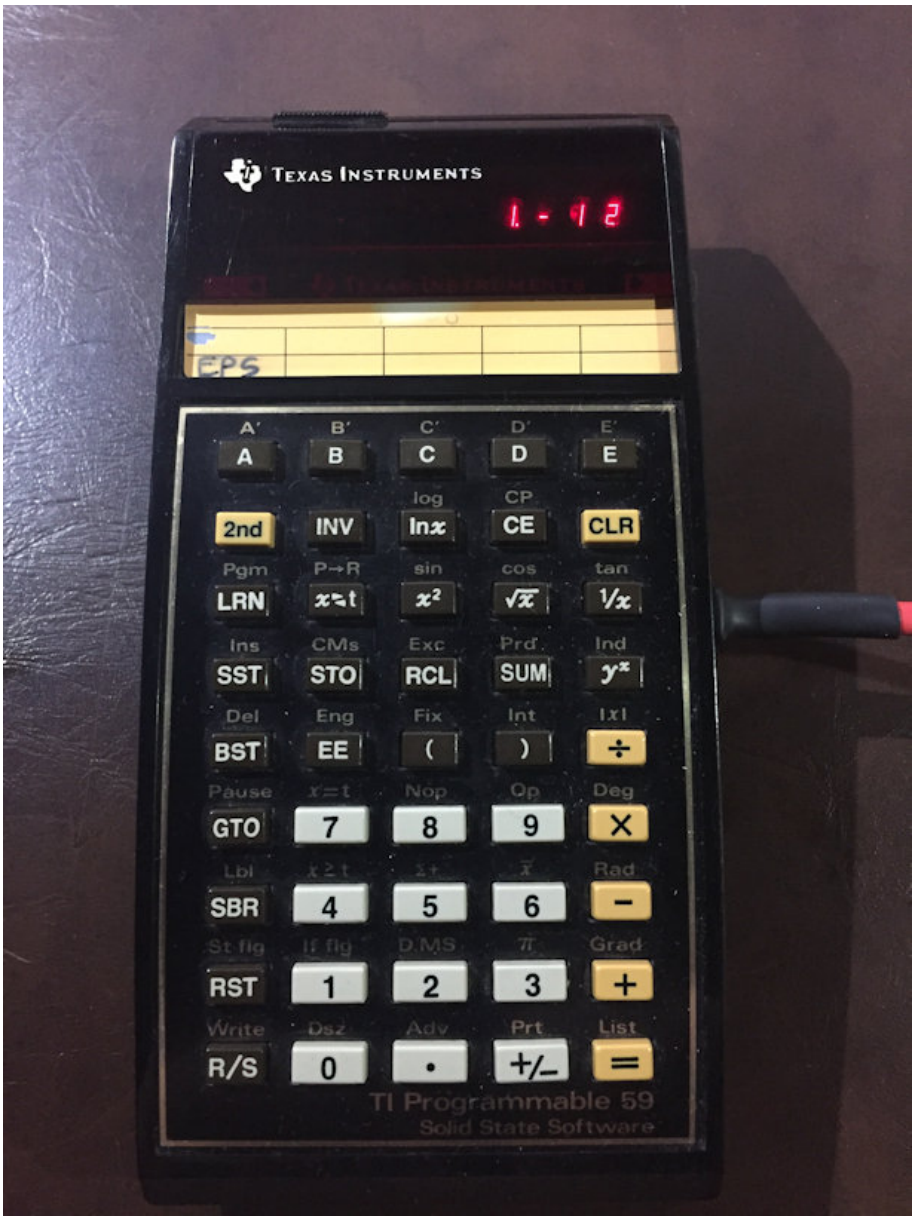
HP48G®


HEWLETT PACKARD 48G
32K RAM
eps:
10.00000000E-12
RAD POLAR CHARS MODES MEMORY STACK PREV MENU
MTH PRG CST VAR NXT
UP HOME DEF RCL →NUM UNDO PICTURE VIEW SWAP
STO EVAL
ASIN ACOS ATAN LOG LN
SIN COS TAN
EQUATION MATRIX EDIT CMD PURG ARG CLEAR DROP
ENTER +/- EEX DEL
USER ENTRY SOLVE PLOT SYMBOLIC
TIME STAT UNITS
I/O LIBRARY EQ LIB
CONT OFF
ON
CANCEL
SPC

TI95®


TEXAS INSTRUMENTS TI-95 PROCALC
1.-12
EPS
ON
OFF
RESET
F1
F2
F3
F4
F5
BREAK
HALT
HYP
INCR
EXC
STO
RCL
INV
SIN
COS
TAN
LN
LOG
2nd
I/O
FILES
STAT
CONV
NUM
FLAGS
TESTS
FUNC
LIST
CLEAR
LEARN
ALPHA
HELP
OLD
RUN
CE

TI59®


Texas Instruments
1 - 12
EPS
A' B' C' D' E'
A B C D E
log CP
2nd INV lnx CE CLR
Pgm P→R sin cos tan
LRN x≥t x² √x 1/x
Ins CMs Exc Prd Ind
SST STO RCL SUM y^x
Del Eng Fix Int |x|
BST EE ( ) ÷
Pause x=t Nop Op Deg
GTO 7 8 9 ×
Lbl x≥t x̄ Rad
SBR 4 5 6 −
St flg If flg D.MS π Grad
RST 1 2 3 +
Write Dsz Adv Prt List
R/S 0 • +/− =
TI Programmable 59
Solid State Software

Code for PC with MATLAB®

```
function [premac,digits] = premac()
digits = 1;
premac = 0.5;
while (premac+1 > 1)
    premac = premac/2;
    digits = digits+1;
end
premac = 2*premac;
digits = digits -1;
return
```

```
>> [premac,number_digits] = premac

premac =

   2.2204e-16

number_digits =

    52

>> eps              \% Builtin eps Matlab function

ans =

   2.2204e-16

>> premac2 = 2^(-number_digits)

premac2 =

   2.2204e-16
```

Code for DM42

```
00 { 79-Byte Prgm }
01 LBL "eps"
02 10              @ base = 10
03 1/X             @ 1/base = 0.1
04 STO 00          @ save to register 00
05 1               @ initialize digits = 1
06 STO 01          @ save counter to register 01
07 LBL a           @ Main Loop find EPS and Digits counter
08 RCL 00          @ read EPS
09 1               @ EPS + 1
10 +
11 1               @ 1
12 X=Y?            @ test if EPS + 1 = 1
13 GTO b           @ if test is true jump LBL b
14 10              @ if test is false
15 STO÷ 00         @ EPS <--- EPS/10
16 1               @ increment counter Digits
17 STO+ 01
18 GTO a           @ jump LBL a (Main Loop)
19 LBL b           @ EPS <--- EPS*10
20 10
21 STOx 00
22 1               @ decreases by one Digits counter
23 STO- 01
24 CLST            @ clear stack
25 " eps = "       @ save alpha eps =
26 ARCL 00         @ queues EPS value
27 ┣"[LF] Digits = "  @ save alpha new line Digits =
28 ARCL 01         @ queues Digits counter
29 AVIEW           @ display results
30 PROMPT          @ stop program
31 RTN
32 END
```

Code for DM42 optimized version

```
00 { 60-Byte Prgm }
01 LBL "eps"
02 10          @ base = 10
03 1/X         @ 1/base = 0.1
04 ENTER       @ save on the stack
05 ENTER       @ save on the stack
06 ENTER       @ save on the stack
07 STO 01      @ save on the register 01
08 STO- 01     @ clear register 01
09 LBL A
10 STO 00      @ storage 0.1 on the 01
11 x           @ product on the stack
12 1
13 STO+ 01     @ increment counter digits
14 +           @ addition
15 1           @ 1
16 -           @ subtract
17 X>0?        @ test if result is > 0
18 GTO A       @ if test is true jump LBL A
19 CLST        @ else clear stack menu
20 "eps = "    @ display eps
21 ARCL 00     @ queues register 00 to display
22 ├"[LF]digits = " @ queues digits 2nd line to display
23 ARCL 01     @ queues register 01 to display
24 AVIEW       @ see display
25 PROMPT      @ stop program
26 RTN
```

Code for HP50G® and HP48G® RPL

```
\<< .1 \-> e
  \<<
    WHILE 'e+1>1'
    REPEAT e 10 / 'e' STO
    END e 10 * "eps" \->TAG
  \>>
\>>
```

Code for TI95®

```
000 DFN F1:EPS@AA
007 HLT
008 LBL AA
011 1
012 STO A
014 0.1
017 STO B
019 LBL BB
022 RCL B
024 +
025 1
026 =
027 IF=A
029 SBL CC
032 RCL B
034 /
035 10
036 =
037 STO B
039 GTL BB
042 LBL CC
045 RCL B
047 *
048 10
049 =
050 HLT
```

Code for TI59®

Location	Key Code	Key Sequence
000	76	LBL
001	11	A
002	00	0
003	93	.
004	01	1
005	42	STO
006	00	00
007	01	1
008	32	X/T
009	76	LBL
010	12	B
011	43	RCL
012	00	00
013	85	+
014	01	1
015	95	=
016	67	X=T ?
017	13	if test is true jump to C
018	43	RCL
019	00	00
020	55	/
021	01	1
022	00	0
023	95	=
024	42	STO
025	00	00
026	61	GTO
027	12	B
028	76	LBL
029	13	C
030	43	RCL
031	00	00
032	65	*
033	01	1
034	00	0
035	95	=
036	92	INV SBR