



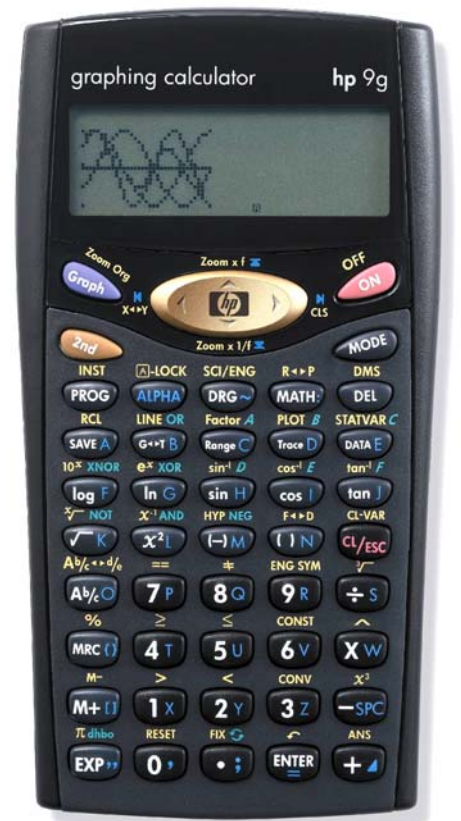
hp calculators

HP 9g Writing a Small Program

Programming the HP 9g

Practice Example: Fibonacci Numbers

Practice Example: Number of Days Between Dates



Programming the HP 9g

To paraphrase the HP 25 Owner's Handbook, the HP 9g means painless programming. Many of its commands use a C-like syntax, and all are described in the concise yet dense instruction sheet included with your calculator. This learning module won't be a photocopy of the manual, instead we will review the basics, emphasizing some of the pitfalls that the programmer might run up against. Then, we'll try to find out what programs are good for...

Programs are entered in the Program mode (MODE 3Z). In this menu you can create a new program (0), edit an existing one (2Y), delete one program (3Z), run a program (1X), and trace a program (^ 4T). Tracing is particularly useful when debugging a program because program execution halts after each instruction, which is displayed until ENTER is pressed. The RCL menu can be used to view the contents of any standard variable). There can be up to ten programs: program 0 through program 9. This number is small, admittedly, but keep in mind that the program capacity is 400 steps (the HP-25 had only 49 program steps, but gave rise to a great deal of interest, and numerous collections of programs were published).

Most programs can be run using the PROG key in the Main mode, but programs that contain Base-N functions will probably have to be run in the Program Run mode instead, because neither hex digits nor logical operators can't be entered in the Main mode. Program Run mode automatically sets the Base-N mode when running Base-N programs. The HP 9g learning module *Logical Operations with Base-N Numbers* contains an example of this type of programs. To interrupt a running program press CL/ESC , select Y and press ENTER . (You may need to press CL/ESC several times).

Most program instructions are in the INST menu (2nd INST), including all program control instructions. Remember that you can scroll this menu with the cursor keys and also that pressing 2nd INST again, displays the next submenu. Other commands are in the MATH menu, in the second (MATH MATH) and third (MATH MATH MATH) submenus in particular, which contain functions that are most useful in a program, such as MAX, MIN, Frac, INT, SGN and ABS. Not all the menus that are available in Main are displayable when editing a program, and therefore its functions can't be included in a program. Particularly, the angle unit and the display format have to be set by the user before running the program. Also, neither unit conversions (2nd CONV) nor Fraction/Mixed/Decimal conversions (2nd A b/c d/e and 2nd F D) are programmable. Unfortunately, the statistical functions are not programmable either.

Program lines are not numbered, but in these learning modules line numbers have been included to help you enter the program and locate a command. Remember that more than one command can be entered into the same line by separating them with a semicolon. Editing a program is similar to editing the entry line. Just bear in mind that CL/ESC does not clear the current line, but exits the Program Edit mode instead.

And now the sixty-four thousand dollar question: what are programs good for? There are at least three things programs are good for, namely: for speeding up calculations, for doing something that couldn't be done otherwise, and for enhancing the built-in functionality.

Practice example: Fibonacci numbers

Each term in the Fibonacci sequence (after the first two) is the sum of the preceding two terms. The n th Fibonacci number is given by:

$$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2} \quad \text{for } n > 1$$

We'll write an HP 9g program to calculate the n th term. Since subroutine nesting (i.e. GOSUB PROG) cannot exceed three levels, a loop version will be used instead.

Line No.	Command	Explanation	Keys
Line 1	INPUT N	Prompts for N	$\overline{2nd}$ INST \wedge \wedge \wedge 0) (ALPHA) (N) \overline{ENTER}
Line 2	F=0	Initializes F (current term)	(ALPHA) log F (ALPHA) \overline{ENTER} 0) \overline{ENTER}
Line 3	P=1	Initializes P (next term)	(ALPHA) 7P (ALPHA) \overline{ENTER} 1X \overline{ENTER}
Line 4	FOR(I=0;I <N; I++) {	The following two instructions are executed N times	$\overline{2nd}$ INST 3Z (ALPHA) cos I (ALPHA) \overline{ENTER} 0) $\overline{2nd}$ (ALPHA) cos I $\overline{2nd}$ < (ALPHA) (N) (ALPHA) cos I $\overline{2nd}$ INST $\overline{2nd}$ INST 2Y \overline{ENTER}
Line 5	SWAP(F,P)	Interchanges the contents of F and P	$\overline{2nd}$ INST \wedge \overline{ENTER} (ALPHA) 7P (ALPHA) 0) (ALPHA) log F \overline{ENTER}
Line 6	P=P+F }	Adds current term to next term. Updates P.	(ALPHA) 7P (ALPHA) \overline{ENTER} (ALPHA) 7P +J (ALPHA) log F \overline{ENTER}
Line 7	PRINT F	Displays Nth term.	$\overline{2nd}$ INST \wedge \wedge \wedge \overline{ENTER} (ALPHA) log F
Line 8	END	End of program.	Inserted automatically when the program is created.

Notice that the way of assigning a value to a variable is different from the way used in the entry line. In Program Edit mode, to store a value into a variable, we have to use *variable = value* (i.e. the (ALPHA) \overline{ENTER} key), whereas in the entry line the syntax is *value*→*variable* (i.e. the (SAVE A) key).

Example 1: Calculate the ratio between F_{15} and F_{16} .

Solution: Once the above program has been keyed in, run it by pressing (PROG) (0-9) \overline{ENTER} . The message “N=” is displayed in the entry line. Enter the first value (1X) (5U) \overline{ENTER} , notice that the value entered can be the result of a calculation, e.g. A+3). The result appears in the entry line: 610. To find the next term, just press \overline{ENTER} and the program restarts. Pressing (1X) (6V) \overline{ENTER} returns F_{16} in a couple of seconds: 987.

Answer: The ratio is $\frac{987}{610} = 1.618...$ quite similar to the golden ratio. In fact, it can be proved that:

$$\lim_{n \rightarrow \infty} \frac{F_{n+1}}{F_n} = \frac{1 + \sqrt{5}}{2}$$

But this is not the quickest way of calculating the *n*th term. There is an amazing formula to calculate it which does not require any iterations:

$$F_n = \frac{\left(\frac{1 + \sqrt{5}}{2}\right)^n - \left(\frac{1 - \sqrt{5}}{2}\right)^n}{\sqrt{5}}$$

and which can be easily stored in a program thanks to the (SAVE A) (PROG) technique (for more information, please refer to the HP 9g learning module *Solving Problems Involving Equations*). Let's clear N (0) (SAVE A) (N) \overline{ENTER}) and store the formula in, say, PROG 5 (press (MODE) (0) to finish the previous program):

(N) (N) (N) 1X +J \sqrt{K} 5U $\overline{2nd}$ \div 2Y $\overline{2nd}$ \wedge (ALPHA) (N) -SPC (N) (N) 1X -SPC \sqrt{K} 5U $\overline{2nd}$ \div 2Y $\overline{2nd}$ \wedge (ALPHA) (N) \div \sqrt{K} 5U (SAVE A) (PROG) 5U

Example 2: Calculate F_{400}

Solution: Press PROG 5 ENTER ENTER and now enter 400: 4 T 0 , 0 , ENTER . In a flash, the answer will appear in the result line. This calculation would take nearly one minute when using the first program. Two ENTER keys are needed to run the program because the first one retrieves the stored formula and the second one evaluates it, after prompting for N.

Answer: $F_{400} = 1.760236806 \times 10^{83}$. For $n < 50$, all results are integers, there's no need to avoid the roundoff caused by square roots and divisions in the formula. For $n > 49$, results are displayed in scientific notation.

Practice example: number of days between dates

Calculating the number of days between two given dates is one of the commonest calendar functions on HP calculators. However, it is not provided with your HP 9g. Implementing it is a simple task, though, and will be a good example of the sort of applications that you can add to the built-in functionality by writing programs.

Our program will return the actual number of days between the two dates (including leap days). The HP 9g programmability is powerful enough to match the behavior of the HP 12C, on which dates can be entered in two formats, namely Day.MonthYear and Month.DayYear. This feature is certainly nothing to shout about, but is a great opportunity to introduce a most useful programming technique: flags.

The HP 12C Owner's Handbook and Problem Solving Guide provides the algorithm, which is outlined below, along with the HP 9g program listing with indentations for ease of reading. The table on the next page provides explanations and keystroke sequences for each line.

$$\Delta\text{DYS} = f(\text{DT}_2) - f(\text{DT}_1), \text{ where}$$

$$f(\text{DT}) = 365y + 31(m - 1) + d + \text{INT}\left(\frac{Z}{4}\right) - x, \text{ and}$$

for $m \leq 2$:

$$x = 0 \text{ and } z = y - 1$$

and for $m > 2$:

$$x = \text{INT}(0.4m + 2.3) \text{ and } z = y.$$

```

INPUT D[1]
INPUT D[2]
FOR(I=1;I <=2; I++) {
    Y=Frac(D[I] E2) E4
    D=INT(D[I])
    N=INT(Frac(D[I] E2))
    IF ( M≠0) THEN {
        SWAP(D,N)
    }
    IF(N ≤ 2) THEN {
        X=0
        Z=Y - 1
    }
    ELSE {
        X=INT(.4N+ 2.3)
        Z=Y
    }
    D[I]=365Y + 31(N - 1) + D + INT(Z/4) - X }
I=D[2] - D[1]
IF (ABS(I)==1) THEN {
    PRINT I," DAY" }
ELSE {
    PRINT I," DAYS" }
END

```

Line	Command	Explanation	Keys
Line 1	INPUT D[1]	Prompts for date 1	$\overline{2nd}$ INST $\wedge \wedge \wedge \wedge$ 0, (ALPHA) Trace D (ALPHA) (M+) 1 X > ENTER
Line 2	INPUT D[2]	Prompts for date 2	$\overline{2nd}$ INST $\wedge \wedge \wedge \wedge$ 0, (ALPHA) Trace D (ALPHA) (M+) 2 Y > ENTER
Line 3	FOR(I=1;I <=2; I++) {	The following instructions are executed twice, once for each date.	$\overline{2nd}$ INST 3 Z (ALPHA) cos I (ALPHA) ENTER 1 X > (ALPHA) cos I $\overline{2nd}$ ≤ 2 Y > (ALPHA) cos I $\overline{2nd}$ INST $\overline{2nd}$ INST 2 Y >> ENTER
Line 4	Y=Frac(D[I] E2) E4	Stores the year in Y	(ALPHA) 2 Y (ALPHA) ENTER (MATH) (MATH) (MATH) ENTER (ALPHA) Trace D (ALPHA) (M+) (ALPHA) cos I > EXP>> 2 Y > EXP>> 4 T ENTER
Line 5	D=INT(D[I])	Stores the day in D	(ALPHA) Trace D (ALPHA) ENTER (MATH) (MATH) (MATH) 1 X > (ALPHA) Trace D (ALPHA) (M+) (ALPHA) cos I >> ENTER
Line 6	N=INT(Frac(D[I] E2)	Stores the month in N	(ALPHA) () N (ALPHA) ENTER (MATH) (MATH) (MATH) 1 X > (MATH) (MATH) (MATH) ENTER (ALPHA) Trace D (ALPHA) (M+) (ALPHA) cos I >> EXP>> 2 Y > ENTER
Line 7	IF (M≠0) THEN {	If Month.DayYear format	$\overline{2nd}$ INST ENTER (ALPHA) (←) M $\overline{2nd}$ ≠ 0, > > ENTER
Line 8	SWAP(D,N) }	then swaps day and month	$\overline{2nd}$ INST \wedge ENTER (ALPHA) Trace D (ALPHA) 0, > (ALPHA) () N >>> ENTER
Line 9	IF(N ≤ 2) THEN {	If January or February	$\overline{2nd}$ INST ENTER (ALPHA) () N $\overline{2nd}$ ≤ 2 Y > > ENTER
Line 10	X=0	then stores 0 in the aux variable x	(ALPHA) 1 X (ALPHA) ENTER 0, ENTER
Line 11	Z=Y - 1}	and 'year -1' in z	(ALPHA) 3 Z (ALPHA) ENTER (ALPHA) 2 Y -SPC 1 X > ENTER
Line 12	ELSE {	if any other month	$\overline{2nd}$ INST 2 Y ENTER
Line 13	X=INT(.4N+ 2.3)	then stores this value into the aux variable x	(ALPHA) 1 X (ALPHA) ENTER (MATH) (MATH) (MATH) 1 X > . ; 4 T (ALPHA) () N +, 2 Y . ; 3 Z > ENTER
Line 14	Z=Y }	and the year in z.	(ALPHA) 3 Z (ALPHA) ENTER (ALPHA) 2 Y > ENTER
Line 15	D[I]=365Y + 31(N - 1) + D + INT(Z/4) - X }	Calculates f(DT). End of the for loop.	(ALPHA) Trace D (ALPHA) (M+) (ALPHA) cos I > (ALPHA) ENTER 3 Z 6 Y 5 U (ALPHA) 2 Y +, 3 Z 1 X > () N (ALPHA) () N -SPC 1 X > +, (ALPHA) Trace D +, (MATH) (MATH) (MATH) 1 X > (ALPHA) 3 Z ÷ S 4 T > -SPC (ALPHA) 1 X > ENTER
Line 16	I=D[2] - D[1]	Calculates the final result.	(ALPHA) cos I (ALPHA) ENTER (ALPHA) Trace D (ALPHA) (M+) 2 Y > (ALPHA) Trace D (ALPHA) (M+) 1 X > ENTER
Line 17	IF (ABS(I)==1) THEN {	If the difference is ±1 day	$\overline{2nd}$ INST ENTER (MATH) (MATH) (MATH) 3 Z (ALPHA) cos I > $\overline{2nd}$ == 1 X >> ENTER
Line 18	PRINT I, " DAY" }	then displays "±1 day"	$\overline{2nd}$ INST $\wedge \wedge \wedge \wedge$ ENTER $\overline{2nd}$ (ALPHA) cos I 0, > EXP>> -SPC Trace D SWE A 2 Y (ALPHA) >>> ENTER
Line 19	ELSE {	otherwise	$\overline{2nd}$ INST 2 Y ENTER
Line 20	PRINT I, " DAYS" }	uses the plural form.	$\overline{2nd}$ INST $\wedge \wedge \wedge \wedge$ ENTER $\overline{2nd}$ (ALPHA) cos I 0, > EXP>> -SPC Trace D SWE A 2 Y ÷ S (ALPHA)

HP 9g Writing a Small Program

Line 21	END	End of program	Inserted when the program was created
---------	-----	----------------	---------------------------------------

Once the program has been entered (it takes up 195 steps), you can run it in either the MAIN mode ($\langle \text{MODE} \rangle \langle 0 \rangle$) or the Program Run mode ($\langle \text{MODE} \rangle \langle 3 \rangle \langle 1 \rangle$). The program starts by prompting for the DT_1 and DT_2 dates. You don't need to enter the earlier date first, but bear in mind that it calculates the number of dates between the second date and the first one – a negative result means that DT_2 was the earlier date. Note that the dates are stored in $D[1]$ and $D[2]$, which corresponds to the E and F variables respectively. The reason is twofold. Those names work as mnemonics for the user, and array variables make calculations easier, because they are written for $f(D[i])$ (i being an index controlled by the FOR loop), instead of calculating $f(DT_1)$ and then repeating all the calculations for $f(DT_2)$.

Dates can be entered in two formats: DD.MMYYYY and MM.DDYyyy. The program needs to know which format is being used (e.g. does 5.062004 mean May 6 or June 5?). We can take advantage of the fact that a small M is displayed whenever the M variable (i.e. the running memory) contains a non-zero value. In other words, we will use the M variable as a *user flag* (a switch that is either on or off and whose state is tested to make a decision. Its meaning depends entirely on how the program uses it). Just because M can stand for Month, the off or cleared state corresponds to the DD.MMYYYY format – when M is displayed, dates are assumed to be in the MM.DDYyyy format.

Notice that there's no need to extract the day and month twice. Lines 4 through 6 extract the year, day and month and store them in Y, D and N respectively (since M acts as a flag, it can't be used to store the month). If M is set (i.e. has a non-zero-value), the contents of the D and N variables must be interchanged, which is performed by the SWAP command.

Lines 17 through 20 display the result in the entry line. If the number of days is one (or minus one), the word "DAY" is appended to that number, whereas if it's more than one (or zero) it is "DAYS" that is appended.

Example 3: Calculate the number of days between February 7th 1974 and July 9th 2004. Use the MM.DDYyyy format to enter these dates.

Solution: Before running the program, make sure the M annunciator is lit by storing any non-zero value into M. For example, press $\langle 1 \rangle \langle X \rangle \langle M+ \rangle$. Now run the program: $\langle \text{PROG} \rangle \langle 0 \rangle \langle 9 \rangle \langle \text{ENTER} \rangle$ and enter the earlier date when $D[1]=$ is displayed: $\langle 2 \rangle \langle Y \rangle \langle . \rangle \langle 0 \rangle \langle 7 \rangle \langle P \rangle \langle 1 \rangle \langle X \rangle \langle 9 \rangle \langle R \rangle \langle 7 \rangle \langle P \rangle \langle 4 \rangle \langle T \rangle \langle \text{ENTER} \rangle$. Now it is $D[2]=$ that is displayed. Enter the second date, then: $\langle 7 \rangle \langle P \rangle \langle . \rangle \langle 0 \rangle \langle 9 \rangle \langle R \rangle \langle 2 \rangle \langle Y \rangle \langle 0 \rangle \langle 0 \rangle \langle 4 \rangle \langle T \rangle \langle \text{ENTER} \rangle$. The result will appear in a couple of seconds.

Answer: 11110 days.

Example 4: Will 2010 be a leap year?

Solution: Press $\langle \text{ENTER} \rangle$ to restart the program and do a new calculation. The dates are now February 28th 2010 and March 1st 2010, so press $\langle 2 \rangle \langle Y \rangle \langle . \rangle \langle 2 \rangle \langle Y \rangle \langle 8 \rangle \langle 0 \rangle \langle 2 \rangle \langle Y \rangle \langle 0 \rangle \langle 1 \rangle \langle X \rangle \langle 0 \rangle \langle \text{ENTER} \rangle$ and $\langle 3 \rangle \langle Z \rangle \langle . \rangle \langle 0 \rangle \langle 1 \rangle \langle X \rangle \langle 2 \rangle \langle Y \rangle \langle 0 \rangle \langle 1 \rangle \langle X \rangle \langle 0 \rangle \langle \text{ENTER} \rangle$.

Answer: The difference between those dates is one day, so 2010 won't be a leap year. (Alternatively, you can calculate the number of days between Feb 29 and Mar 1, which is zero: they're actually the same date!)

Example 5: How many days are there between September 11th 2001 and March 11th 2004? Use the DD.MMYyyy format this time.

Solution: Press $\langle \text{MODE} \rangle \langle 0 \rangle$ to exit the program and $\langle M+ \rangle \langle M+ \rangle \langle \text{CL}/\text{ESC} \rangle$ to clear M. Start the program, and enter the dates by pressing: $\langle 1 \rangle \langle X \rangle \langle 1 \rangle \langle X \rangle \langle . \rangle \langle 0 \rangle \langle 9 \rangle \langle R \rangle \langle 2 \rangle \langle Y \rangle \langle 0 \rangle \langle 0 \rangle \langle 1 \rangle \langle X \rangle \langle \text{ENTER} \rangle$ and $\langle 1 \rangle \langle X \rangle \langle 1 \rangle \langle X \rangle \langle . \rangle \langle 0 \rangle \langle 3 \rangle \langle Z \rangle \langle 2 \rangle \langle Y \rangle \langle 0 \rangle \langle 0 \rangle \langle 4 \rangle \langle T \rangle \langle \text{ENTER} \rangle$.

Answer: 912 days.