♦MoHPC♠ *The Museum of HP Calculators*

HP 9100A/B programming

Warning: On these calculators the ENTER button reads programs from magnetic cards. It is called "the ENTER button" in this text. The key that enters numbers on the stack is labeled with an upward pointed arrow but is shown in this text as "ENTER^". The downward pointed arrow is shown as "drop" here.

Contents:

Features

- Memory space shared between program and registers with no sizing command (allows self-modifying code.)
- Up to 16 registers on the HP 9100A. Up to 32 on the HP 9100B.
- Up to 196 program steps on the HP 9100A. Up to 392 on the HP 9100B. (Unmerged - each keystroke takes a step.)
- An external memory expansion option adds up to 248 registers or 3472 program steps.
- Core memory requires no power to maintain state.
- Step number addressing.
- Unconditional and conditional branching based on register values.
- Subroutines on the HP 9100B only. (Possible by self-modifying code on the 9100A.)

Basic Programming

Entering And Running A Sample Program

A simple program is essentially just the keystrokes you would press to solve the program manually. The calculator remembers a sequence of keys and executes them in order at the touch of a key. In the simplest programs, there is only the addition of END keystroke.

To enter a program, make sure the calculator's PROGRAM/RUN switch is set to RUN and press END. This sets the program counter to 00. Now switch to PROGRAM mode.

With the PROGRAM/RUN switch set to PROGRAM the number you see on the left side indicates the step number. The HP 9100A shows each program step in the X register. The HP 9100B shows the current step in the Z register and the previous step in the X register. (Don't count on the previous step being meaningful on line 00.) As keystrokes are entered, their octal key codes will be displayed on the right. The first key you press will go into step 00.

Enter the program below to compute the area a circle:

```
ENTER^      This key is labeled with an up arrow
x           Square the radius
PI
x           Multiply by PI
END
```

Now set the PROGRAM/RUN switch to RUN and reset the program counter to 00 by pressing END. To run the program key in a radius and press CONTINUE (CONT on the HP-9100B) and the area will be displayed in the Y register almost instantly. The program can be run as many times as you like by entering new values and pressing CONTINUE. The display will blank during programs, but the calculator is so fast that the simple examples shown here will cause a barely noticeable blink. For many simple programs, that's all you need to know!

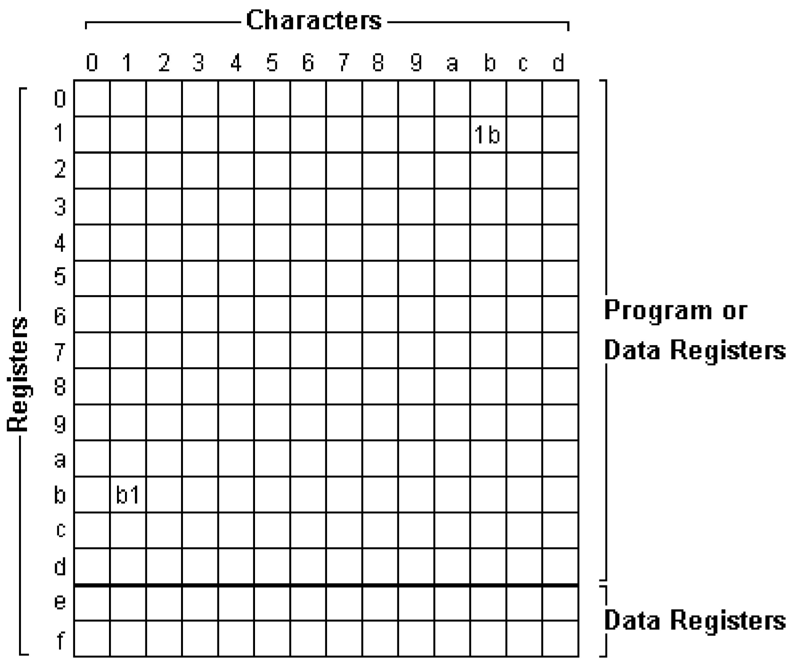
Stopping, Interrupting and Entering Data

Many programs only require data to be entered at the beginning as the one above did. Remember that you can use the stack and store values in registers before execution. (Use x->() or y->() to store numbers. a-f are recalled by just pressing the keys, 0-9 are recalled with y<>() or x<-() on the HP 9100B.)

You can also enter STOP instructions into the program to allow additional data to be entered. The user can enter the data and then press CONTINUE. If you create a program that runs for too long (possibly due to an infinite loop) you can stop it by pressing STOP and if necessary, resume it by pressing CONTINUE. If you press STOP in the middle of a program, you can see where the program is executing by switching to PROGRAM mode.

Memory

The HP 9100A has 16 registers of core memory. 14 of these registers (0-d) can be used to hold 14 program steps each which allows up to a maximum of 196 steps with two storage registers e and f. The memory layout is shown below:


Characters
0 1 2 3 4 5 6 7 8 9 a b c d
Registers
0 1 2 3 4 5 6 7 8 9 a b c d e f
1b
b1
Program or Data Registers
Data Registers

There is no command to determine how much space is dedicated to storage registers versus program code. Use is determined "on the fly" and you can overwrite code (intentionally or accidentally) by storing to a register via y->(), y<>() etc. Generally, you should start programs at step 00 and use registers starting from the top down in order to avoid a collision. Besides, the upper registers a-f can be recalled with one key whereas the only way to retrieve a value from registers 0-9 on the HP 9100A is to swap it via the y<>() instruction. The x<-() key makes this easier on the HP 9100B but it still requires one more keystroke than the alpha registers.

The HP 9100B has two pages of core labeled + and -. GO TOs are assumed to go to the current page unless a + or - precedes the address like GO TO + n n. Storage and recall operations assume the + page unless the minus page is specified like y->() - a to store in register a on the - page. (Precede the address with the minus key - don't use the CHG SIGN key.)

As an optimization on either calculator, consider placing program initialization code at the highest addresses of the program where it can be overwritten by register stores when it is no longer needed.

Stepping And Editing programs

STEP PGRM steps one instruction ahead. In RUN mode the next instruction is executed and the X register is then displayed making this instruction useful for debugging. It can also be used for moving forward without execution in PROGRAM mode. This command is not recordable.

GO TO nn can be used in RUN mode to position the calculator to step nn. The calculator can then be switched to PROGRAM mode for editing at that step. This function is recordable so the position of the PROGRAM-RUN switch must be properly set.

Entering an instruction causes the new instruction to overwrite the one currently displayed. (Most HPs add or overwrite new instructions *after* the one currently displayed.) To replace a single instruction, display the instruction and key in the correct instruction which will replace the error. If you find that you need to *add* instructions to a program, you can do one of the following:

1. Rewrite it from the point of the error on down.
2. Insert a GO TO nn at the point where the new instructions would be needed, perform the new steps elsewhere and use a GO TO to go back to the step after the first GO TO. Remember to include the instructions that were overwritten by the first GO TO in your new block of code. Or
3. Use the card reader to patch the program. (See Saving and Loading Programs below for an example of this method).

Here's an example of inserting a GO TO (method 2):

Original code;

```
...
10 -
11 y->()
12 4            Oops! Need to square x and take the log before storing!
13 5
14 *
...
```

Patched code:

```
...
10 -
11 GOTO ()()
12 4
```

```
13 0              Overwrite store 4 & PI with a jump to some unused memory
14 *              Return here
15 ...
...
40 ENTER^         The new steps
41 x              square x
42 log x
43 y->()          The steps overwritten by GO TO 40
44 4
45 5
46 GO TO ()()     Could have used GO TO SUB / RETURN on the HP 9100B
47 1
48 4              Return to the normal program flow
```

Saving and Loading Programs

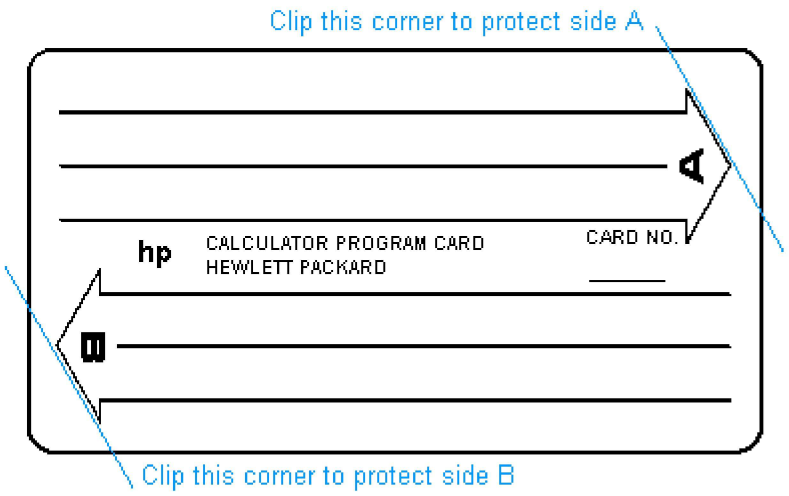
To save a program to magnetic card, set the PROGRAM-RUN is switch to RUN (this is different from most HPs). Hold the card with the label facing the you and with the the arrow of the track you want to record facing down and insert the card all the way into the read/writer slot.

Press GO TO n n to select a position to begin writing. (END is a handy shortcut if you want to start at 00.) Now press the RECORD button and the card will rise back out of the slot. The X register is destroyed when cards are written.

The first 14 registers (0-d inclusive) are written to the card (up to the END statement.) On the HP 9100A, the calculator's memory is stored on just one track of a card and the other side is available for a second program. The HP 9100B requires both tracks to save its entire core memory so after recording side A, reverse the card (still label toward you) to record side B. (The calculator does you the favor of doing a GO TO - 0 0 to get ready to write the second page after recording track A. Press GO TO - n n if you want to start from a different address on the second page.)

Note that while it's possible to write functioning programs without END statements, programs *recorded* without END statements may or may not read back properly. If programs steps are recorded in both pages of the HP 9100B, each page should have exactly one END statement.

You may cut off the corner of a magnetic card to prevent it from being overwritten. Don't clip more than shown below or you could loose data. Cards can be marked with a permanent felt tip pen or lightly with a soft pencil.


Clip this corner to protect side A
A
hp CALCULATOR PROGRAM CARD
HEWLETT PACKARD
CARD NO.
B
Clip this corner to protect side B

To load a program from a magnetic card, make sure the PROGRAM-RUN is switch set to RUN and place a card (label side toward the keyboard) all the way into the card reader/writer.

Press GO TO n n to select a position to begin reading. (END is a handy shortcut if you want to start at 00.) Now press the ENTER button next to the reader and the card will come back out of the slot. Remember to GO TO - n n (usually 0 0) and read side B if you've recorded both pages on an HP 9100B. The stack is undisturbed when cards are read.

Here's an example of using the card reader to insert three program steps (method 3 from stepping and editing above). Suppose you need to insert two steps at line 42. Make sure the PROGRAM/RUN switch is set to RUN and press GO TO 42 to position the calculator. Now record the memory from steps 42 and up onto the card by inserting the card and pressing the RECORD button. Now insert the two instructions at line 42 by pressing GO TO 42, switching to PROGRAM mode and keying the instructions.

Now (leaving the calculator positioned after the instructions just entered) switch the calculator back to RUN, insert the same program card and press the ENTER button next to the card reader. The calculator will read the rest of the program back into memory after the insertion. Note, however, that you'll need to adjust any GO TOs that reference addresses in the moved code.

Saving and Loading Data

It's possible, though a bit tricky to store data on a card along with the program. The tricky part is that you need to have your program end after the the data you want to save. For example if you wanted to preserve the data in registers up to c, you could reserve register d for the end of your program. At the end of your program, add GO TO d 0. Then switch to RUN mode and press GO TO d0 and switch back to PROGRAM mode and press END.

Then you must fill all program memory from the end of your program (right after the GO TO d 0) up to the first data register you want to save with CONTINUE operations which serve as No-ops. For example, if your program ended with GO TO d 0 at step 65 and your data data is in registers 8-c, you would place CONTINUE operations in steps 66-7d. (The reason for this is to insure that there isn't another END lurking in this unused space that would end the writing earlier.)

Then you would switch to RUN mode, and save the program as usual. You can't save e and f since they are out of program space.

An alternative method is to leave your program unchanged and simply start the recording at the lowest data register to save. For example, to save a-d, press GO TO a 0 and then insert your card and press the RECORD button. The calculator starts writing and when it gets to the end of memory, it continues writing the card starting at step 00 because it still hasn't found an END. This is tempting because it is much simpler for writing **but remember that you must always press GO TO a 0 before reading this card back in.** You should make a note of this requirement on the card.

Keep in mind that if you're using one of these methods to store constants that are used just once or twice in the program, it may be more space efficient to simply write them into the code. For example 3.717 requires 5 characters as code but an entire 14 character register when stored as data.

Programming Techniques

Addressing

The HP-9100A & B use step addressing allowing you to jump to any line. Addresses are two digit numbers in base 14 of the form 01, 02, ... 09, 0a, 0b, 0c, 0d, 20, 21, ... 2c, 2d, 30, ... dd.

The HP 9100B has two pages of core labeled + and -. GO TOs are assumed to go to the current page unless a + or - precedes the address. (Storage operations are assumed to refer to + unless overridden by -.)

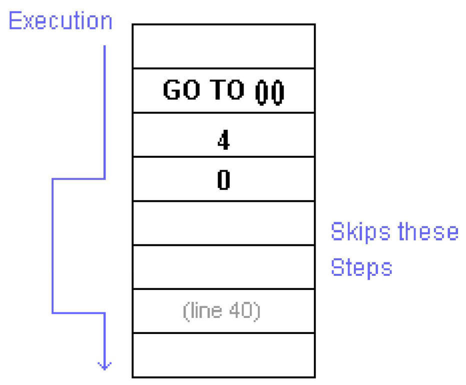
Jumps

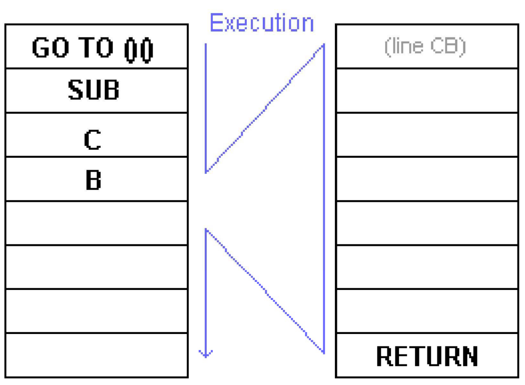
Both calculators have:

- GO TO nn where -dd <= nn <= (+)dd jumps to that step number. nn is a base 14 number (see the memory layout above) so each digit should be less than d.
- END sets the program counter to 0 0 and stops execution.

The HP 9100B adds:

- GO TO SUB nn saves the return address and branches to address nn. The function at nn must end with a RETURN. The HP 9100 can nest subroutines 5 levels deep. To refer to the other page, add a page specifier like GO TO SUB + 5 d.
- RETURN returns from a subroutine that was called by GO TO SUB.





Don't worry that SUB and RETURN are the same key. The calculator knows which you mean because SUB always immediately follows GO TO and RETURN never does.

The program below adds 1 to the X register, displays it for a second and then repeats. Key it in after pressing END and switching to PROGRAM mode.

```
1              top of the loop
+
PAUSE          display for 1/8 second
GO TO ()()
0
0              loop
```

Now switch to RUN mode, press END, key a number, press ENTER^ and press CONTINUE (CONT on the 9100B). When you get tired of it, press STOP. (You might have to press STOP a few times, because of the PAUSE.)

The following program calculates the volume of a cylinder by first calling a subroutine to calculate the area of its base and then multiplying by the length.

```
GO TO ()()
SUB
0
8              Call the subroutine at line 08 to calculate the area
x              Multiply by the length in y
STOP           Display it
GO TO ()()     This shouldn't be an END!
0
0              Ready for the next case
ENTER^         This key is labeled with an up arrow
x              Square the radius
PI             Multiply by PI
x              (Now have X=PI, Y=area, Z=length)
roll dn        Move area to X, length to Y
RETURN
END
```

Now switch to RUN mode and press END. Key in a cylinder length, press ENTER^, key in a radius and press CONTINUE. The volume will be displayed in the Y register.

You may be wondering why the program used GO TO 0 0 in the middle rather than an end. The calculator requires any program that is written to a magnetic card to have exactly one END instruction at the bottom of the program. Even if you're not recording your program, it's good to get in the habit of using END only at the end of a program.

Here's an example of a factorial program for integers <= 69. Press END, switch to PROGRAM mode and enter:

```
ENTER^
ENTER^   loop here
2
IF x>y   If we're below 2, we're done
1        Go to the exit routine
1        at step 11 (remember addresses are base 14)
drop     (down arrow on the keyboard)
ENTER^
1        compute n-1
-
drop     (down arrow on the keyboard)
*        n*n-1 (leaving n-1 in the x register - very handy)
```

```
GO TO
0
1
drop     exit routine (step 11): Leave result in y
END
```

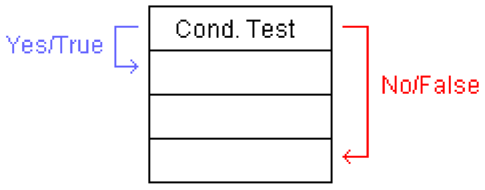
Now switch to RUN mode, press END, press 5 and press CONTINUE. The result of 120 will be displayed in the Y register. Try some larger numbers. Even the maximum computable input of 69 takes under a second.

Conditional Tests and Flags

The HP 9100 has three instructions for comparing X to Y.

- IF x < y
- IF x = y
- IF x > y

If the condition is false, the next two steps are skipped. If the condition is true, execution continues with the next step. When the condition is true, the calculator examines the next two steps and if they contain an address, then the calculator acts as if the Conditional test was a GO TO and branches to that address. If they are not an address, then the calculator simply executes them with no implied GO TO.



Two program fragments follow which show the conditional used both as an implied GO TO and as a conditional skip.

```
IF x < y      If x is equal to y
c             GO TO (implied) step C4
4
cos x         else compute the cosine of x
...

IF x = y      If x is equal to y
PI            Add PI to y
+
x<>y          Execute these steps regardless of conditional
cos x         compute cos of x or x+PI
...
```

The calculators have a single flag which is set by the SET FLAG instruction and tested by IF FLAG. IF FLAG acts like a conditional test skipping two steps if the FLAG is not set. The flag is cleared by testing it or whenever a CLEAR is executed. (Note that CLEAR also clears the stack and registers e and f.)

Input/Output

STOP can be used as an instruction or pressed from the keyboard. If a program is stopped, pressing CONTINUE (CONT on the HP 9100B) starts it. If the program is running, pressing

STOP stops it. A STOP can also be inserted into a program to allow the user to input or record data.

For example, this program accepts the length and radius of a cylinder. It computes the area of the base and stops to allow the answer to be recorded. When the user resumes, it computes the volume.

```
ENTER^
x       Radius squared
PI
x       Area of the base
STOP    Allow user to record it
drop    (Down arrow key)
x       Multiply by length still on the stack
END
```

Set the PROGRAM-RUN switch to RUN, press END, key in a length and press ENTER^. Then key a radius and press CONTINUE. After the area is displayed, press CONTINUE again to display the volume. (Both results are displayed in Y.)

For brief displays, one or more PAUSE instructions may be used to show the display for 1/8 of a second. If the PAUSE key is held down during program execution, a STOP will occur at the next PAUSE in the program. (Press Continue to resume the program in this case.) Using the PAUSE in this way allows you to watch the progress of a program and optionally make adjustments to the data.

The PRINT/SPACE key is used with the optional printer. If no printer is attached, it behaves like STOP so the user can record the display.

The illegal operation lamp is to the left of the display and will light on operations such as divide by zero, sqrt(n) where n<0 etc. It will not stop a running program but will remain lit until a key is pressed.

GTO Go back to the HP 9100A/B

GTO Go back to the software library

GTO Go back to the main exhibit hall