Coding & Mathematics on TI-84 Plus CE

An Introductory Webinar, 23 June 2017, 7 pm Singapore time

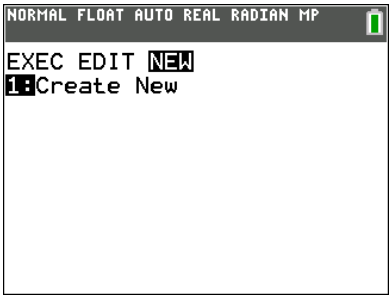
Panelist: Thomas Yeo

By the end of the webinar, participants should be able to:

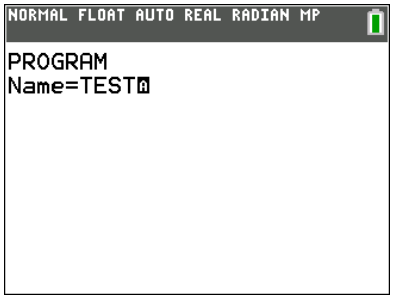
- Create, edit and execute programs on the TI-84 Plus CE
- Write code to display text on the screen
- Write code to process input from the user and output a result
- Write code involving simple decision making processes
- Write code to handle loops and recursions

Section 1: Creating a Program

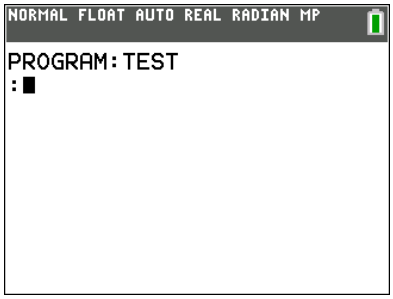
In this section, you will learn how to create your very first program and display simple text.



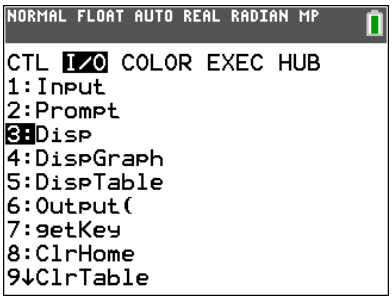
Press prgm and then go to NEW using the arrow keys. Press enter or 1 to create a new program.



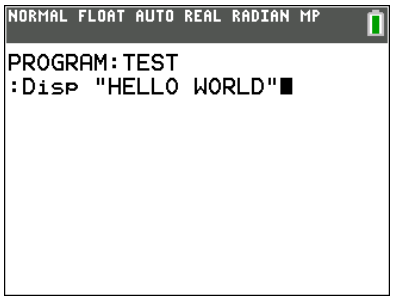
Input a name using the alphabet keys (printed in green). The maximum characters that can be accepted is 8. Press enter when done.



You will now be in the editor mode. We can now start with our very first program, which displays text on the screen.



Press prgm and go to I/O menu. Select **Disp**, which represents "display".



Press alpha + to input the quotes. Text to be displayed must be typed within a pair of quotes. Use the alpha lock 2nd alpha for easy typing. Press alpha again to exit alpha lock. Press enter to complete your very first line of code.

☺ <u>Useful notes:</u>

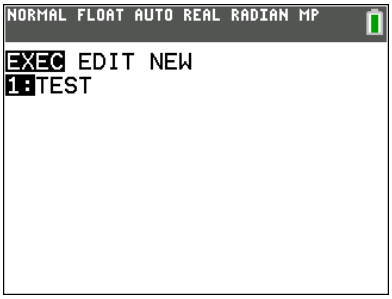
Most other programming platforms require you to save each time you make a change. However, when coding on the TI-84, the program is always automatically saved!

© 2016 – 2017 Thomas Yeo. All Rights Reserved.

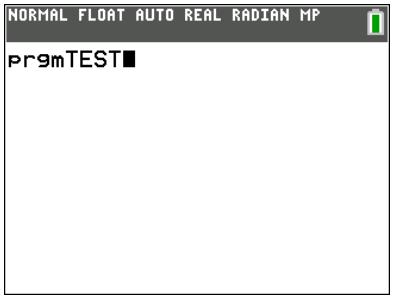
Updated on 22 June 2017

Section 2: Executing a Program

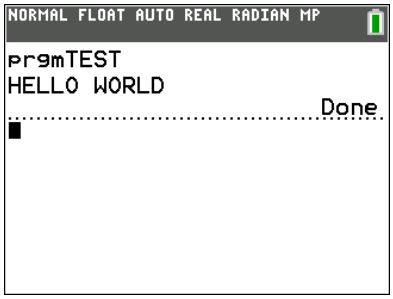
In this section, you will learn how to execute/run a written program on the main screen.



Press [2nd][mode] to exit to home screen. Press [prgm] to go to the Programs menu. Move the selector over the program you wish to run and press [enter].



A line will be automatically pasted on your home screen. Press [enter] to proceed.

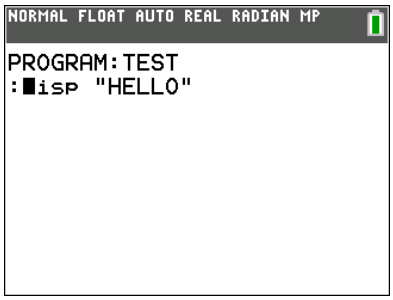


The program will be executed. Congratulations! You have just executed your very first program!

Section 3: Editing and Deleting a Program

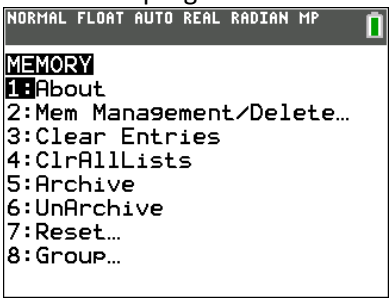
To edit a program:

At the home screen, press [prgm] to go to the programs menu. Press [▸] to go to the Edit menu and select the program to edit

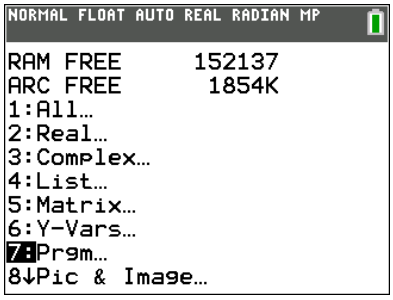


You can now edit the program.

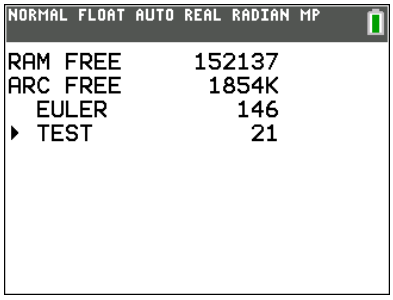
To delete a program:



At the home screen, press [2nd][+] to access the Memory menu. Select **2: Mem Management/Delete**.



Select **7: Prgm**.



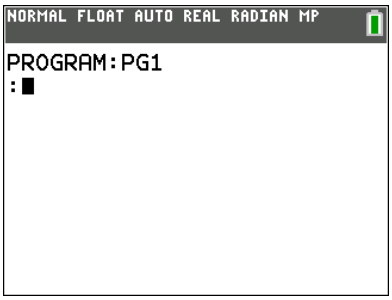
Select the program to delete and press the [del] key. Confirm to delete at the next screen.

© 2016 – 2017 Thomas Yeo. All Rights Reserved.
Updated on 22 June 2017

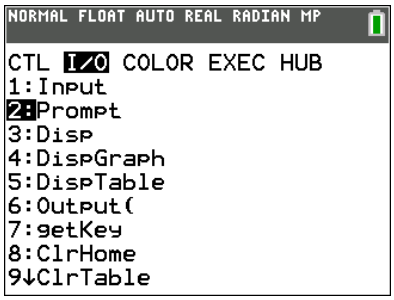
Section 4: Getting Input from the User and Processing it

In this section, you will learn how to get input from the user, process it and give an output. This is the part where some Mathematics can be included for learning purposes and also to train up the logical thinking skills of the students.

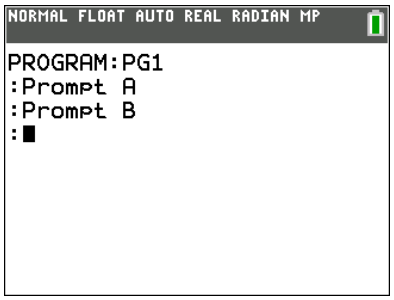
Example 1:
Create a simple program to calculate the hypotenuse *h* of a right angled triangle, where the other 2 sides are *a* and *b* respectively.



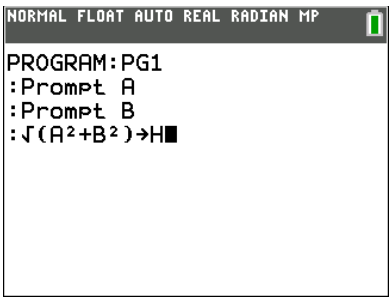
Create a new program and give it a name. I used PG1 here to represent program one.



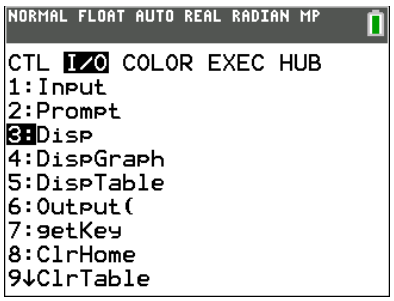
Press prgm and go to I/O menu. Select **2:Prompt**. This is to get input from the user.



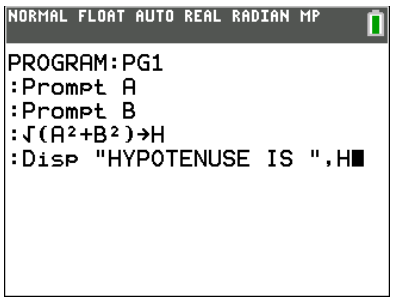
Type the variable name A and press enter to go to the next line. Insert another prompt to get input for B.



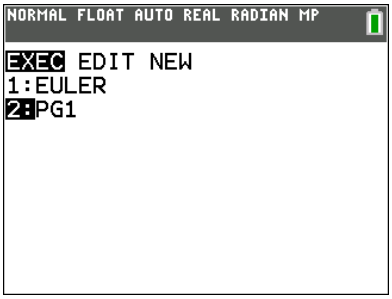
Type in the calculation for H using the calculator keys. To get the arrow, which represents store, press sto→. Press enter when done.



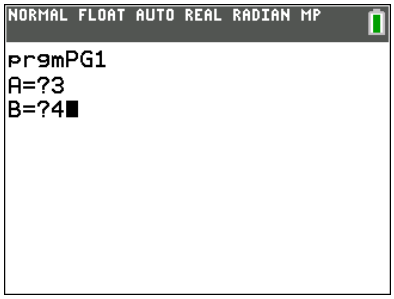
Now let's have a display statement to display the results of the calculation. Press prgm then **I/O** then **3:Disp**.



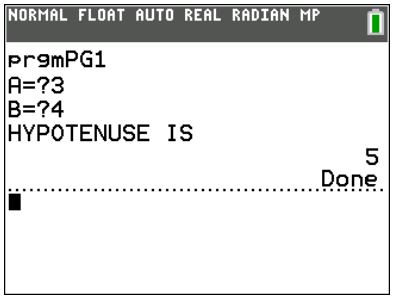
Type in some text as usual and put a comma , after the text, and type in the variable name H.



Go back to home screen and execute the program.



Type in the values of A and B.



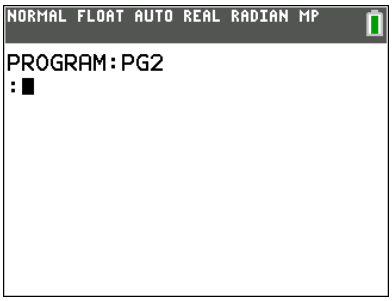
Final result is calculated and displayed.

© 2016 – 2017 Thomas Yeo. All Rights Reserved.
Updated on 22 June 2017

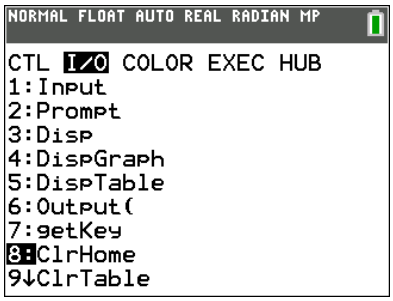
Section 5: Using Loops and Recursions

Sometimes we need the program to run the same program repeatedly for a number of times, for example in a sequence we may wish to run from *n*, *n* + 1, *n* + 2… up to a certain value. In terms of programming, we can use loops (FOR or WHILE) to help us achieve this.

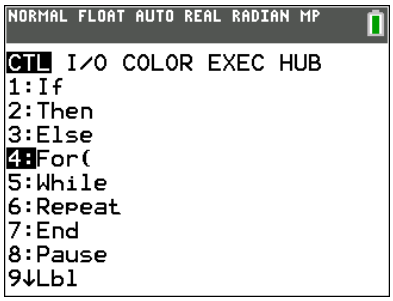
Example 2: (Using FOR loop, WHILE loops will not be discussed in this webinar)
Create a program to generate the first 8 powers of 3 and display it on the screen.



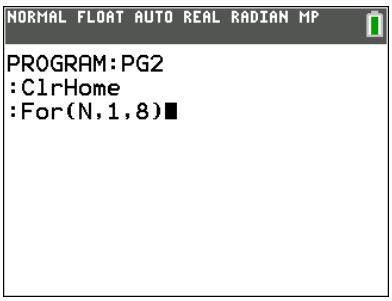
Create a new program and give it a name. I used PG2 here to represent program two.



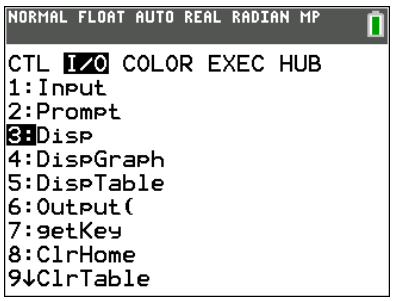
Let's clear the home screen first as we are displaying some output. Go to prgm, **I/O** and select **8:ClrHome**. This statement clears the screen each time the program is ran.



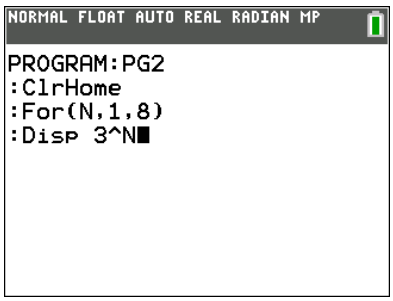
Insert a FOR loop command next. Go to prgm and select **4:For(**.



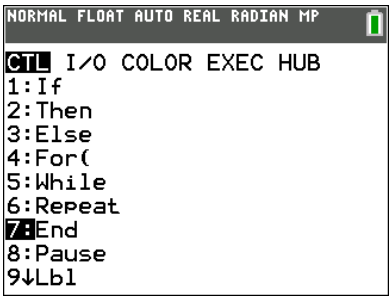
Type in the parameters as shown. Basically this means the loop will run 8 times from n = 1 to n = 8.



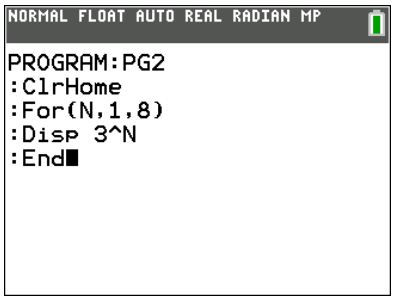
Now let's have a display statement to display the desired results. Press prgm then **I/O** then **3:Disp**.



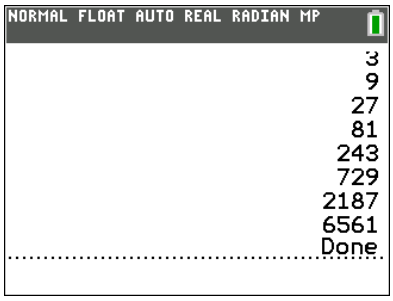
Type in the display command that will generate the powers of 3. We use n as the power so that with every run of the FOR loop, n is increased automatically.



Insert an End statement to close off the FOR loop. Press prgm and select 7:End.



This is the full program, which is simple to understand. Try executing the program.



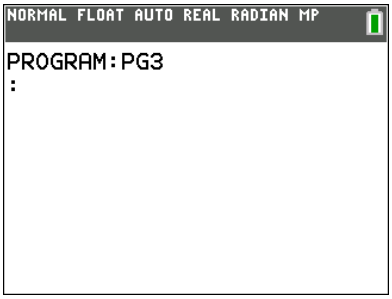
This is the result. As an extension, you can use Prompt statements to ask user for the base number and n.

© 2016 – 2017 Thomas Yeo. All Rights Reserved.
Updated on 22 June 2017

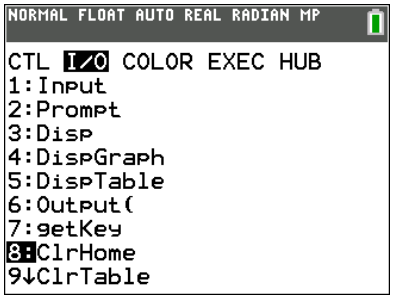
Section 6: Using Decision Statements

In a typical program, we will use decision statements to help make the program more robust and allow more things to be performed.

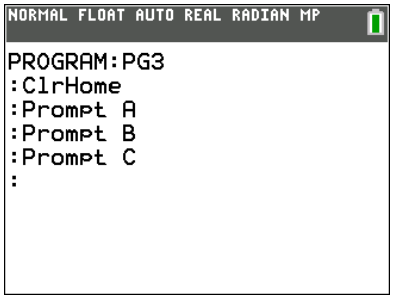
Example 3:
Create a program that allows the user to key in the coefficients of the terms in the polynomial $ax^2 + bx + c$. The program should determine the nature of the roots of the polynomial and display an appropriate message to the user.



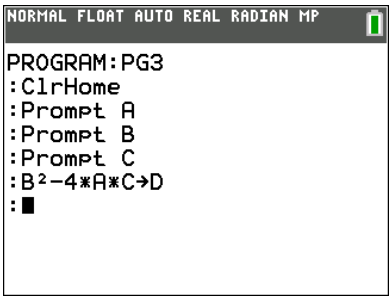
Create a new program and give it a name. I used PG3 here to represent program three.



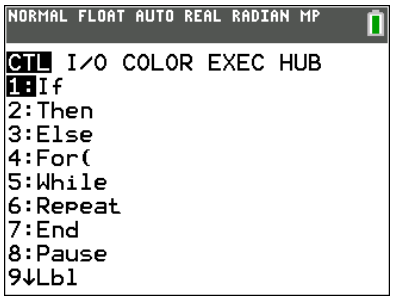
Let's clear the home screen first as we are displaying some output. Go to [prgm], **I/O** and select **8:ClrHome**. This statement clears the screen each time the program is ran.



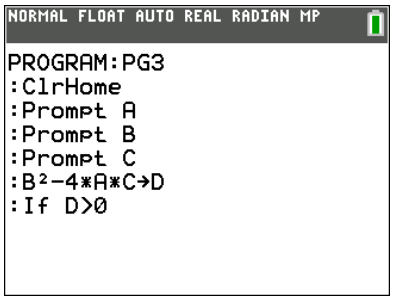
Insert 3 Prompt statements to get the input for values of *a*, *b* and *c*.



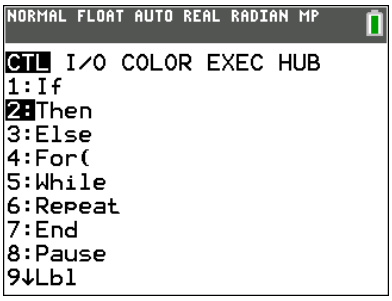
Next we input a calculation for the discriminant and store it in D.



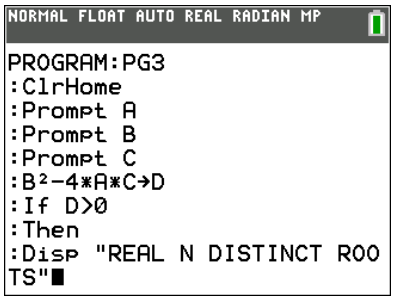
Go to [prgm] and select **1:If**.



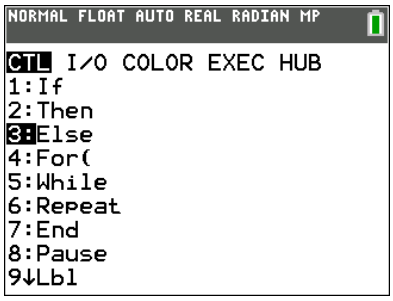
Type in the condition for the first check. The inequality symbol can be found in [2nd][math].



Insert a THEN statement from [prgm] then selecting **2:Then**.



Enter a display message for real and distinct roots.



Enter an ELSE statement to continue checking for other conditions. Press [prgm] then select **3:Else**.

© 2016 – 2017 Thomas Yeo. All Rights Reserved.
Updated on 22 June 2017

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:PG3
:Prompt C
:B²-4*A*C→D
:If D>0
:Then
:Disp "REAL N DISTINCT ROO
TS"
:Else
:If D<0
:
```

Now type in the second IF statement to check for d<0, following similar steps above.

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:PG3
:If D>0
:Then
:Disp "REAL N DISTINCT ROO
TS"
:Else
:If D<0
:Then
:Disp "NO REAL ROOTS"
:
```

Insert the THEN statement and display message accordingly.

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:PG3
TS"
:Else
:If D<0
:Then
:Disp "NO REAL ROOTS"
:Else
:Disp "REAL N REPEATED ROO
TS"
:
```

Enter another ELSE statement and the display for the last condition which has real and repeated roots.

```
NORMAL FLOAT AUTO REAL RADIAN MP
CTL I/O COLOR EXEC HUB
1:If
2:Then
3:Else
4:For(
5:While
6:Repeat
7:End
8:Pause
9↓Lbl
```

To close off the IF statement, insert the END statement from prgm then **7:End**.

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:PG3
:If D<0
:Then
:Disp "NO REAL ROOTS"
:Else
:Disp "REAL N REPEATED ROO
TS"
:End
:End
:
```

We need 2 END statements for the 2 IF statements.

```
NORMAL FLOAT AUTO REAL RADIAN MP
A=?1
B=?2
C=?-5
REAL N DISTINCT ROOTS
                    Done
```

Test the program and this is the result. Case 1: real and distinct roots.

```
NORMAL FLOAT AUTO REAL RADIAN MP
A=?2
B=?3
C=?11
NO REAL ROOTS
                    Done
```

Case 2: no real roots.

```
NORMAL FLOAT AUTO REAL RADIAN MP
A=?1
B=?-2
C=?1
REAL N REPEATED ROOTS
                    Done
```

Case 3: real and repeated roots.

```
NORMAL FLOAT AUTO REAL RADIAN MP
A=?1
B=?2
C=?-3
REAL N DISTINCT ROOTS
D IS
                      16
                    Done
```

You may wish to display the value of D as well, as shown here. Just add another display statement in the IF statement section.

Some tips:

- Avoid using the clear key where possible to avoid chunks of code being deleted. Instead use 2nd del to use the insert mode to add in code in between lines. This approach is also used to insert new lines in between lines of code.

© 2016 – 2017 Thomas Yeo. All Rights Reserved.
Updated on 22 June 2017