Lab 3

Introduction to the Operating Systems and the BASIC Programming Language

Starters:

Make a list of commands (and what they do) that you have used from the Command Prompt.

Make a list of BASIC commands that you have used in a BASIC program.

Instruction Conventions

➪	When the "process" icon appears in the box on the left, then this box will contain one or more instructions that you will need to follow. The instructions will be a specific as is practical but could be different for different users and computer configurations.
📝	When the "note" icon appears in the box to the left, this box will contain notes, hints, or tips that may be helpful to the lab activities.
⌨	<user input>
	When the keyboard icon appears in the box to the left, the box above will contain a line of input to be entered by the user, and this box will contain an explanation of what the user input will do.

Lab 3: BASIC Conditional Programming Outline

Part 0: Review of Operating System and BASIC Programming Topics : from the last class.

Part 1: Set up the computing environment

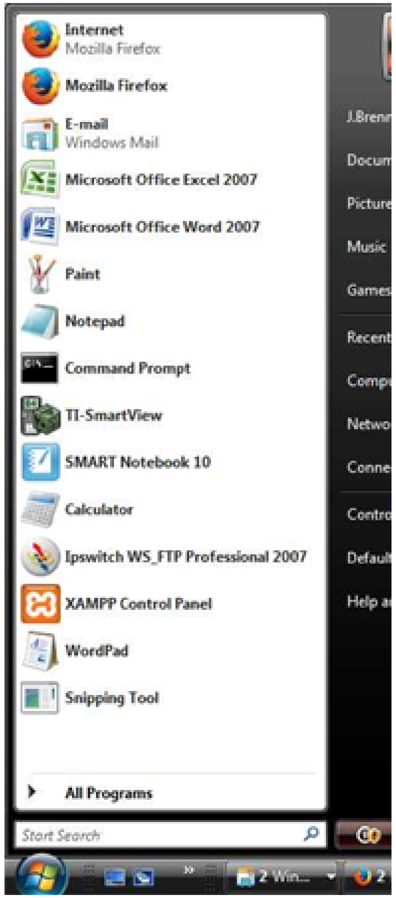
Part 2: Running and Editing the preliminary versions of the QUAD1.bas program to find the roots of quadratic equations.

Part 3: Add conditional programming to process error conditions and special values.

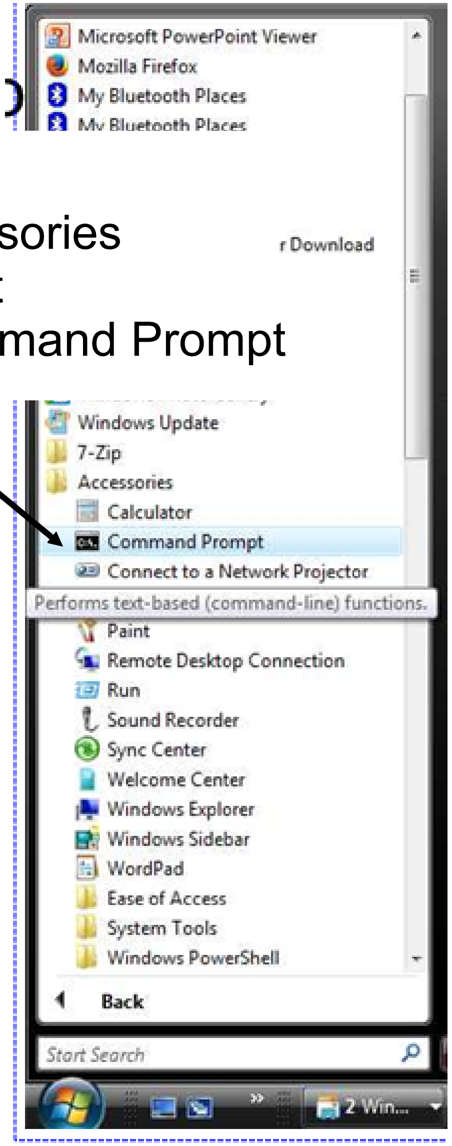
Part 4: Add iterations to repeat program segments.

Part 1: Set up the computing environment

1) From the start menu select All Programs

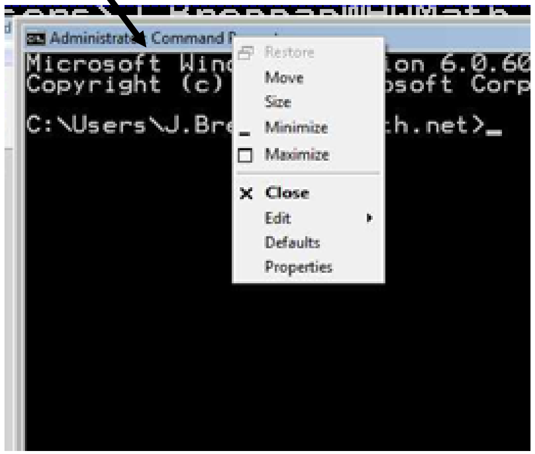


2) From Accessories Select Command Prompt



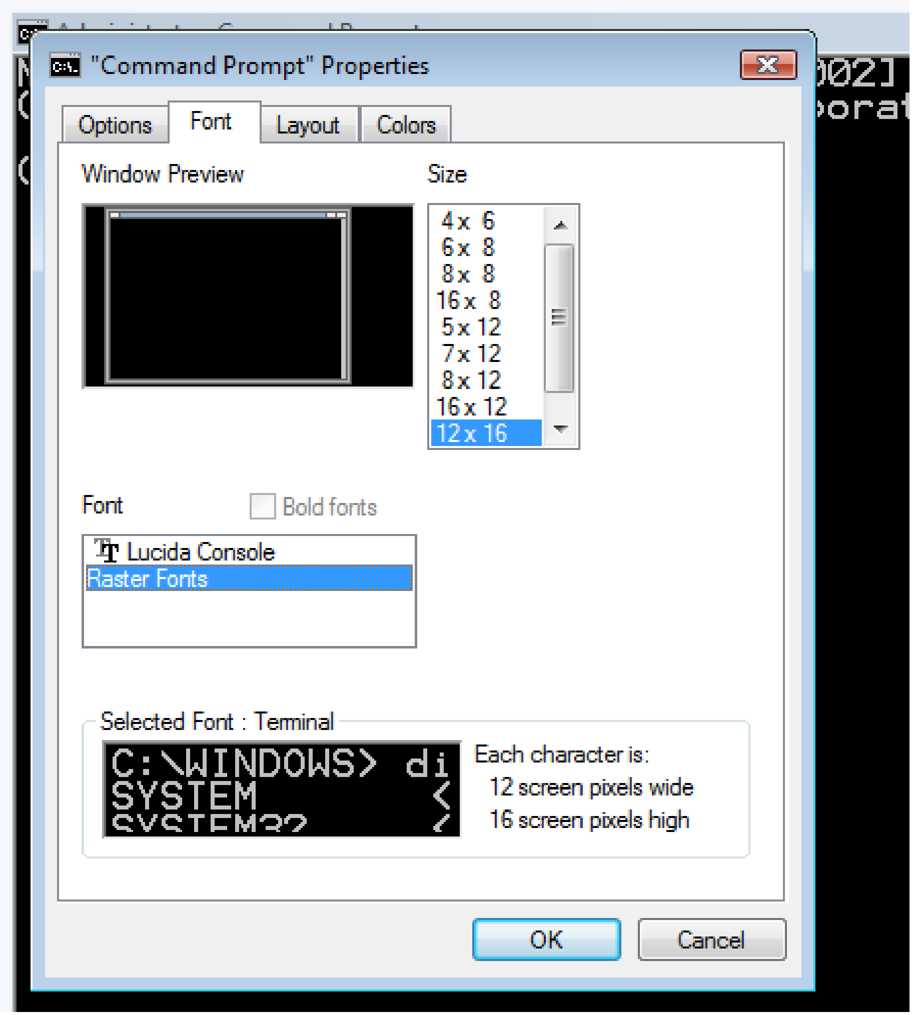
Part 1: Set up the computing environment

3) Right click on the top bar and select Properties



```
Microsoft Wind            ion 6.0.60
Copyright (c)             osoft Corp

C:\Users\J.Br             h.net>_
```

4) Select Font and then 12 x 16



Size
4 x 6
6 x 8
8 x 8
16 x 8
5 x 12
7 x 12
8 x 12
16 x 12
12 x 16

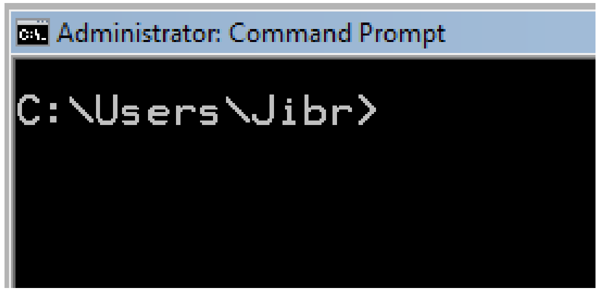
Font □ Bold fonts
Lucida Console
Raster Fonts

Selected Font : Terminal

```
C:\WINDOWS> di
SYSTEM       <
SYSTEM32     <
```

Each character is:
12 screen pixels wide
16 screen pixels high

Part 1: Set up the computing environment



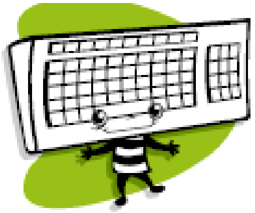
```
C:\Users\Jibr>
```

When you opened the Command Prompt window, it probably took you to the directory for your userid. You should have system permissions to read and write files from this directory.

In this case my current working directory is Jibr which is a subdirectory in the Users directory on my C: drive.

5) Insert the CD into the CD/DVD drive. Remember that this will be the D: drive on your computer.

Part 1: Set up the computing environment

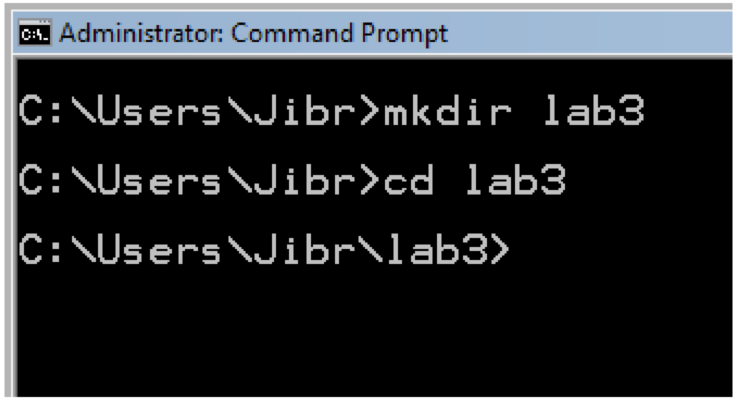


6) issue the following command to create a new subdirectory in your current working directory:

```
mkdir lab3
```

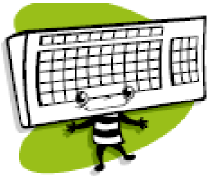
7) issue the cd command to change directories. This command will make lab3 your current working directory on your C: drive

```
cd lab3
```



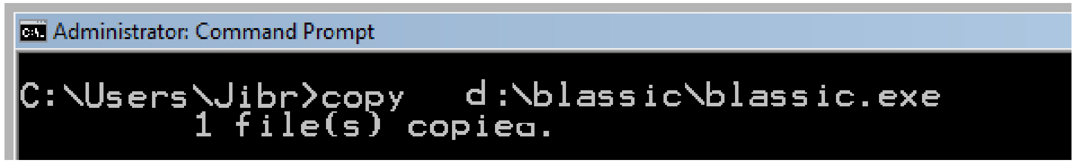
```
C:\Users\Jibr>mkdir lab3

C:\Users\Jibr>cd lab3

C:\Users\Jibr\lab3>
```

Part 1: Set up the computing environment



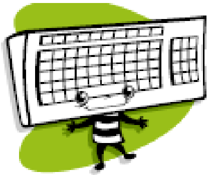
8) Copy the blassic program from the CD into your current working directory with the command:

```
copy  d:\blassic\blassic.exe
```



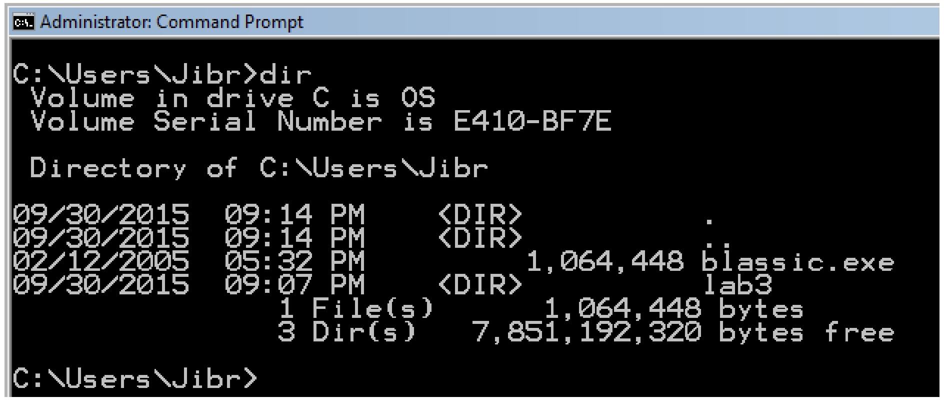
```
C:\Users\Jibr>copy   d:\blassic\blassic.exe
        1 file(s) copied.
```

When you copy a file and specify the name of the copy to file, but don't the command where to put the copy - it will put the copy in your current working directory, using the same file name.



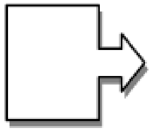
9) Verify that the blassic program has been copied by listing the files in your current directory:

```
dir
```



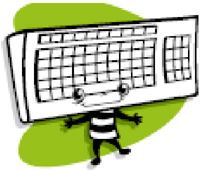
```
C:\Users\Jibr>dir
 Volume in drive C is OS
 Volume Serial Number is E410-BF7E

 Directory of C:\Users\Jibr

09/30/2015  09:14 PM    <DIR>          .
09/30/2015  09:14 PM    <DIR>          ..
02/12/2005  05:32 PM         1,064,448 blassic.exe
09/30/2015  09:07 PM    <DIR>          lab3
               1 File(s)      1,064,448 bytes
               3 Dir(s)   7,851,192,320 bytes free

C:\Users\Jibr>
```

Part 2: Running and Editing the preliminary versions of the QUAD1.bas program to find the roots of quadratic equations.



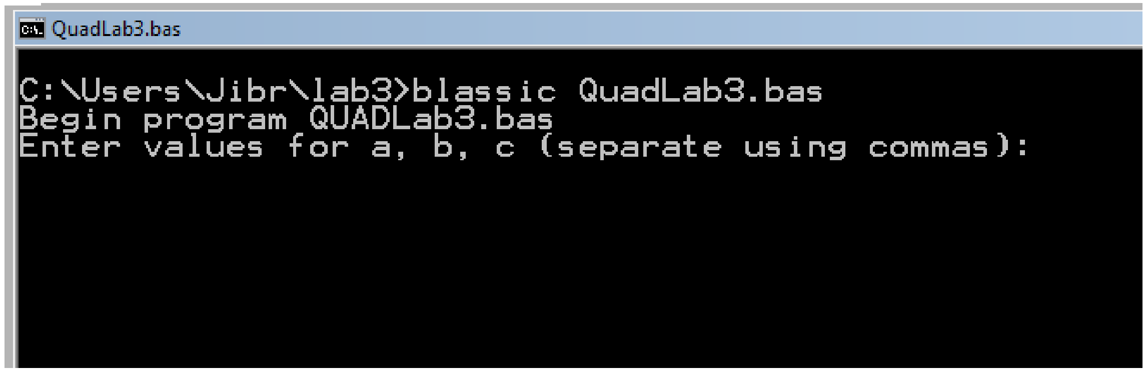
1) In a web browser, go to hwmath.net/IBC
2) Click on QuadLab3.bas which will open in a new tab
3) Select all of program lines of QuadLab3.bas and copy them to your clip board.
4) From the Windows start menu, select All Programs -> Accessories -> Notepad
5) Paste the contents of the clipboard into the blank file
6) Save this as QuadLab3.bas in the lab3 subdirectory that you created.

Keep notepad open.



7) Switch back to the Command Prompt window and issue the command:

```
blassic QuadLab3.bas
```



```
C:\Users\Jibr\lab3>blassic QuadLab3.bas
Begin program QUADLab3.bas
Enter values for a, b, c (separate using commas):
```

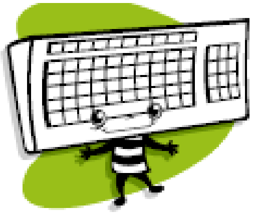
Part 2: Running and Editing the preliminary versions of the QUAD1.bas program to find the roots of quadratic equations.

8) Enter valid values to make sure you get the expected output. Run the program multiple times to make sure it is running as expected.

In the last class we ran blassic interactively - in this lab you will use blassic to execute a program and use Notepad to edit the program.

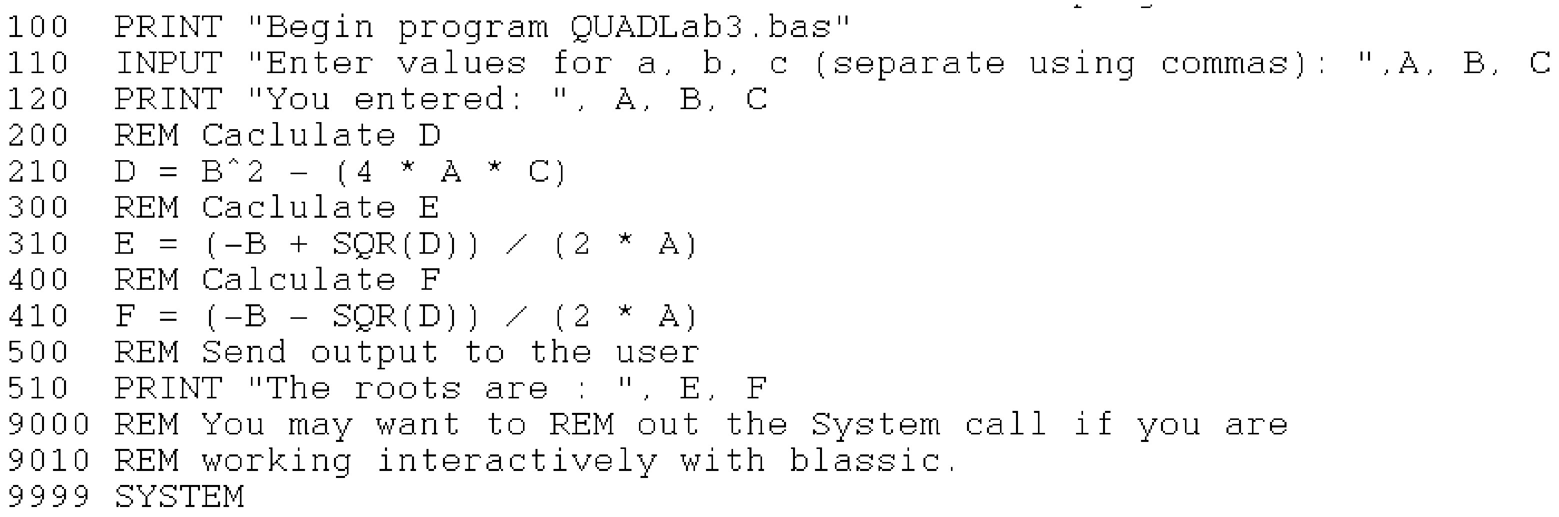
Notice that when you run blassic you do not need to specify the file extension .exe as part of the file name.

Part 2: Running and Editing the preliminary versions of the QUAD1.bas program to find the roots of quadratic equations.



9) Edit line 30 and put in your name

Part 3: Add conditional programming to process error conditions and special values.

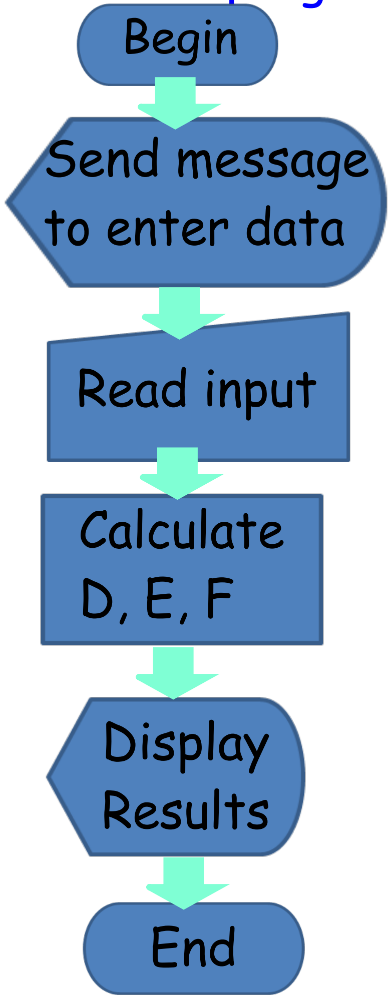


```
100  PRINT "Begin program QUADLab3.bas"
110  INPUT "Enter values for a, b, c (separate using commas): ",A, B, C
120  PRINT "You entered: ", A, B, C
200  REM Caclulate D
210  D = B^2 - (4 * A * C)
300  REM Caclulate E
310  E = (-B + SQR(D)) / (2 * A)
400  REM Calculate F
410  F = (-B - SQR(D)) / (2 * A)
500  REM Send output to the user
510  PRINT "The roots are : ", E, F
9000 REM You may want to REM out the System call if you are
9010 REM working interactively with blassic.
9999 SYSTEM
```

The program is currently executed sequentially, one statement (line) at a time. We will add some conditional statements to perform some error checking.

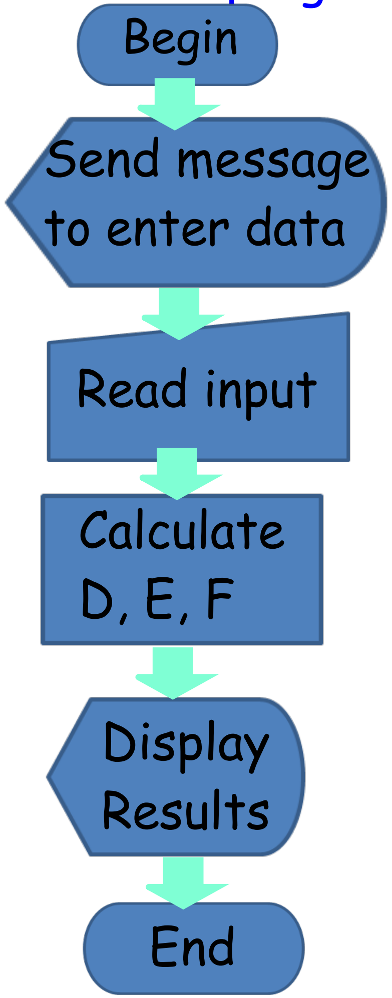
Part 3: Add conditional programming to process error conditions and special values.

A flow chart of the program is demonstrated here:



Part 3: Add conditional programming to process error conditions and special values.

A flow chart of the program is demonstrated here:



We have outline this using pseudo-code a couple of times

Prompt the use for A,B,C

Read the users input
Echo the users input
Calculate D, E, F

Send the user the roots D,E

Part 3: Add conditional programming to process error conditions and special values.

We'll add error checking to make sure the user entered a non-zero value for A, and add error processing to send the user a message and terminate the program if the user entered 0 for A.

```
Prompt the use for A,B,C

Read the users input
Echo the users input
If the user entered A = 0
   then
        send the user a message
        terminate the program.
   else
       Calculate D, E, F
       Send the user the roots D,E
end if
```

Part 3: Add conditional programming to process error conditions and special values.

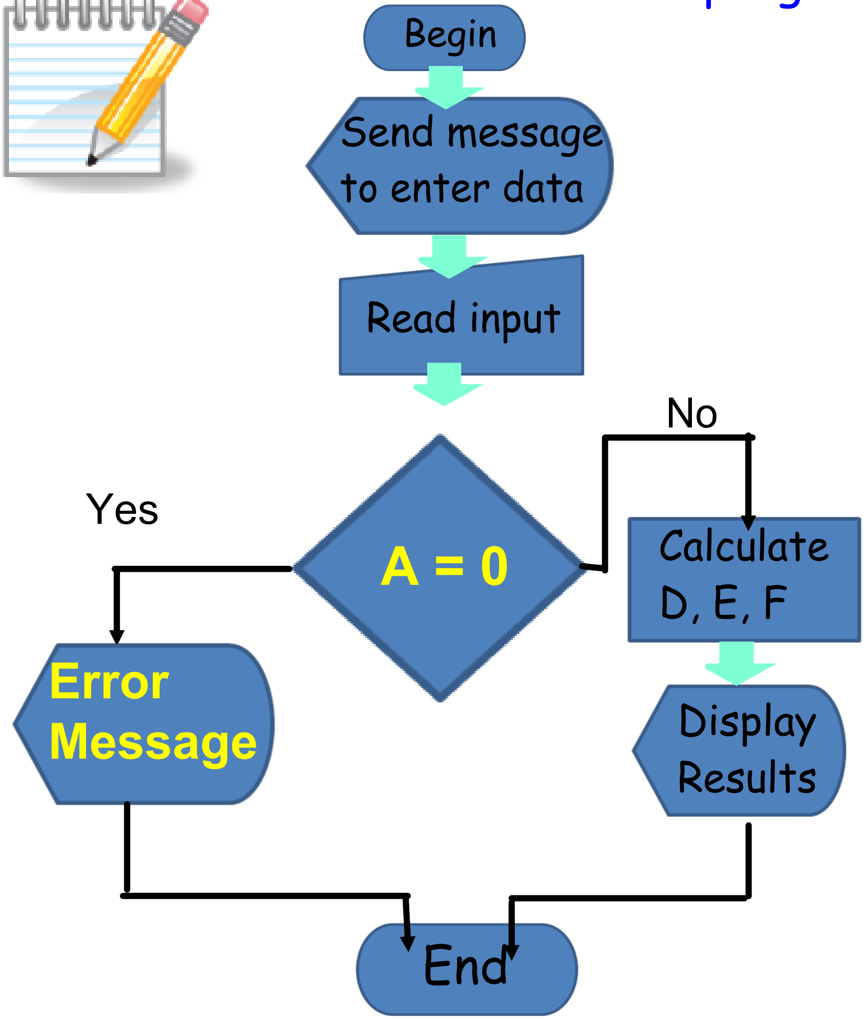
We'll add error checking to make sure the user entered a non-zero value for A, and add error processing to send the user a message and terminate the program if the user entered 0 for A.

```
Prompt the use for A,B,C

Read the users input
Echo the users input
If the user entered A = 0
   then
        send the user a message
        terminate the program.
   else
       Calculate D, E, F
       Send the user the roots D,E
end if
```

The only way to get to here is if A <> 0

Part 3: Add conditional programming to process error conditions and special values.

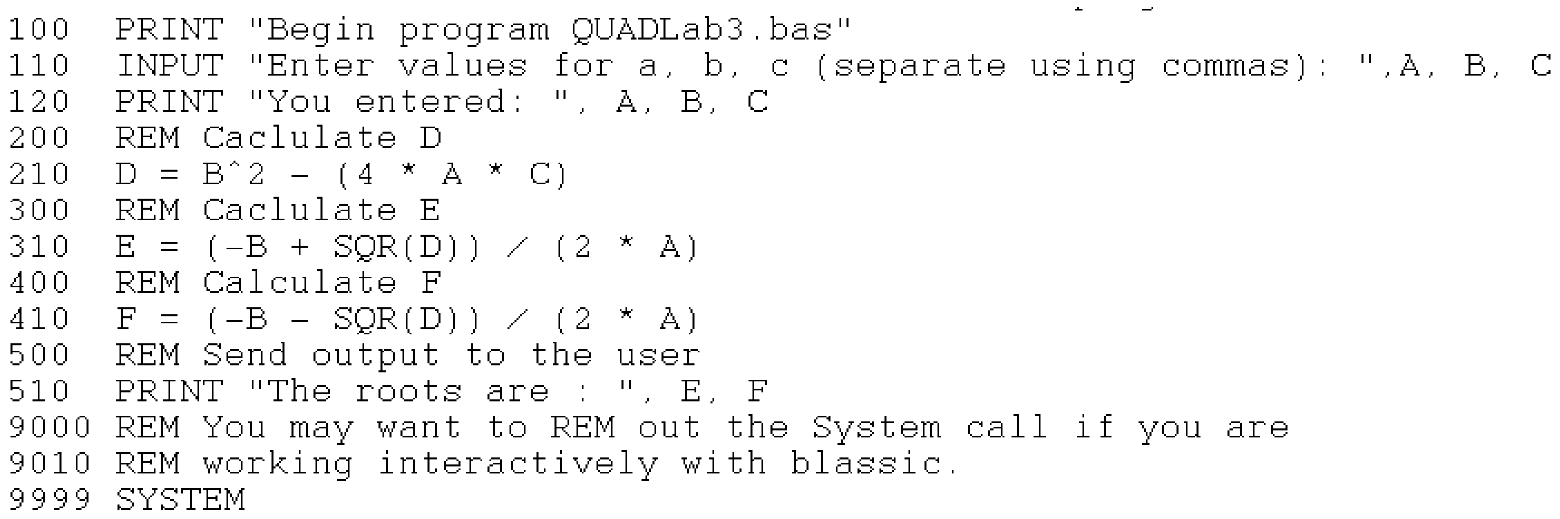
A flow chart of the program is demonstrated here:



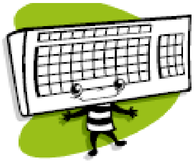
We have outline this using pseudo-code a couple of times

```
Prompt the use for A,B,C

Read the users input
Echo the users input
If the user entered A = 0
   then
        send the user a message
        terminate the program.
   else
       Calculate D, E, F
       Send the user the roots D,E
end if
```

Part 3: Add conditional programming to process error conditions and special values.



```
100  PRINT "Begin program QUADLab3.bas"
110  INPUT "Enter values for a, b, c (separate using commas): ",A, B, C
120  PRINT "You entered: ", A, B, C
200  REM Caclulate D
210  D = B^2 - (4 * A * C)
300  REM Caclulate E
310  E = (-B + SQR(D)) / (2 * A)
400  REM Calculate F
410  F = (-B - SQR(D)) / (2 * A)
500  REM Send output to the user
510  PRINT "The roots are : ", E, F
9000 REM You may want to REM out the System call if you are
9010 REM working interactively with blassic.
9999 SYSTEM
```

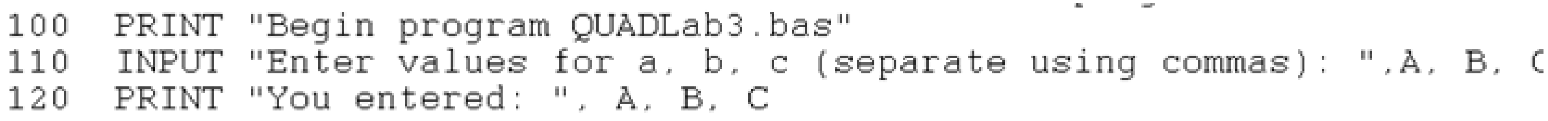


After line 120 we know what the user entered. We want to make sure that we only get to line 200 is if the user entered a value for A that is not zero. So enter this as line 130:

```
130   IF A <> 0 THEN 200
```

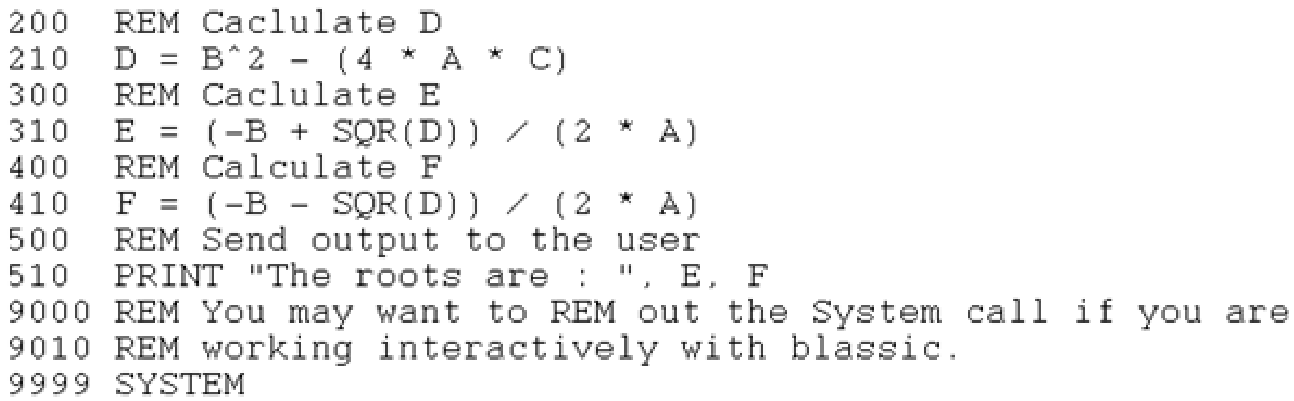
This statement says "If the value for A is not 0 then goto line 200" and means the user entered a valid value and it is okay to calculate the value of D.

Part 3: Add conditional programming to process error conditions and special values.

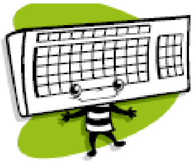


```
100  PRINT "Begin program QUADLab3.bas"
110  INPUT "Enter values for a, b, c (separate using commas): ",A, B, C
120  PRINT "You entered: ", A, B, C
```

```
130   IF A <> 0 THEN 200
```



```
200  REM Caclulate D
210  D = B^2 - (4 * A * C)
300  REM Caclulate E
310  E = (-B + SQR(D)) / (2 * A)
400  REM Calculate F
410  F = (-B - SQR(D)) / (2 * A)
500  REM Send output to the user
510  PRINT "The roots are : ", E, F
9000 REM You may want to REM out the System call if you are
9010 REM working interactively with blassic.
9999 SYSTEM
```



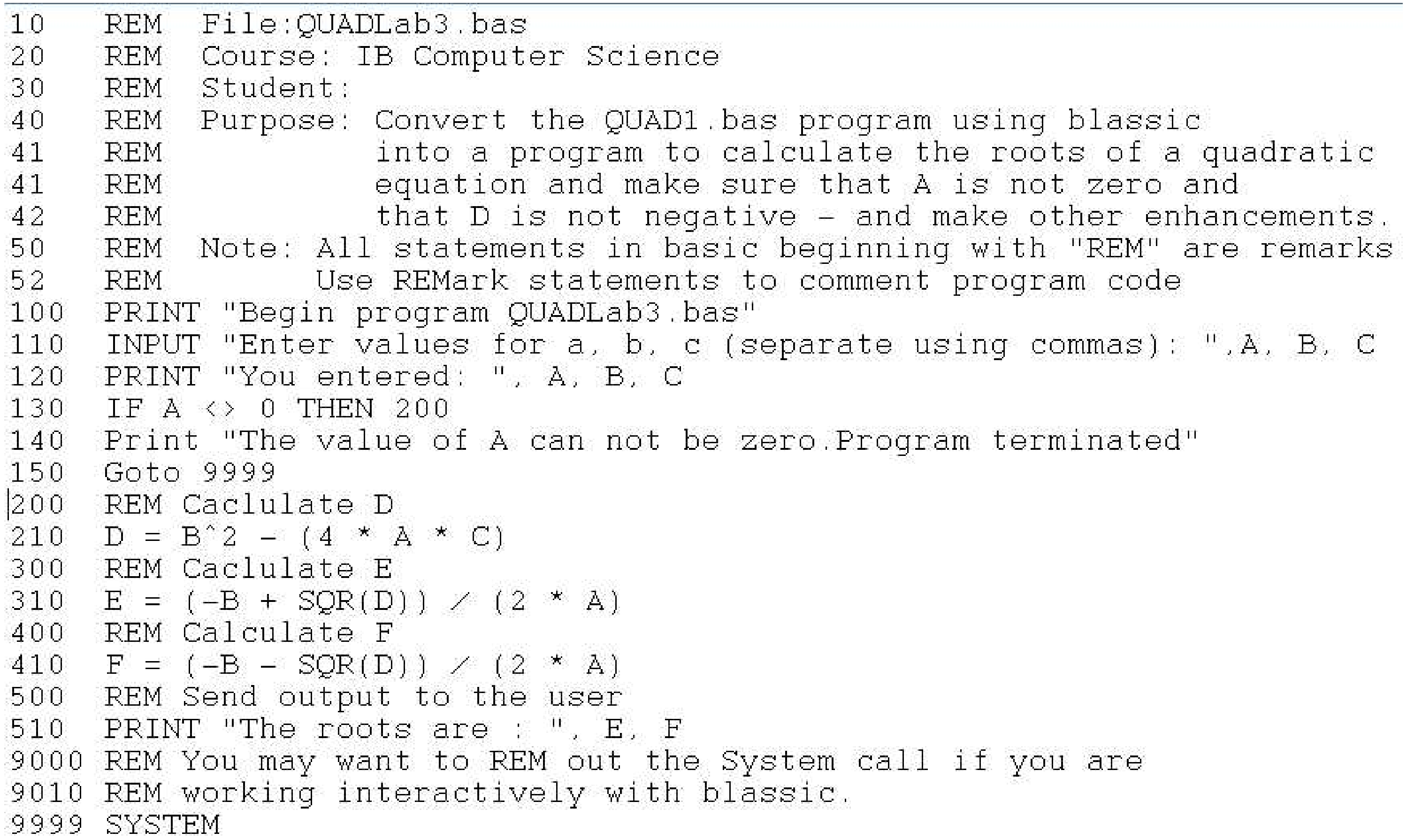
If the user entered a value that is not 0 then the program will jump to line 200.

If the user entered a value that IS equal to zero then the program will continue to execute the next line in the program. So if the value IS equal to zero - send the user a message and terminate program.

```
140   Print "The value of A can not be zero.Program terminated"
150   GOTO 9999
```

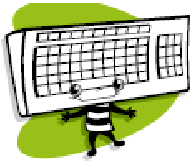
Part 3: Add conditional programming to process error conditions and special values.

Your program should be similar to the following:



```
10   REM  File:QUADLab3.bas
20   REM  Course: IB Computer Science
30   REM  Student:
40   REM  Purpose: Convert the QUAD1.bas program using blassic
41   REM           into a program to calculate the roots of a quadratic
41   REM           equation and make sure that A is not zero and
42   REM           that D is not negative - and make other enhancements.
50   REM  Note: All statements in basic beginning with "REM" are remarks
52   REM        Use REMark statements to comment program code
100  PRINT "Begin program QUADLab3.bas"
110  INPUT "Enter values for a, b, c (separate using commas): ",A, B, C
120  PRINT "You entered: ", A, B, C
130  IF A <> 0 THEN 200
140  Print "The value of A can not be zero.Program terminated"
150  Goto 9999
200  REM Caclulate D
210  D = B^2 - (4 * A * C)
300  REM Caclulate E
310  E = (-B + SQR(D)) / (2 * A)
400  REM Calculate F
410  F = (-B - SQR(D)) / (2 * A)
500  REM Send output to the user
510  PRINT "The roots are : ", E, F
9000 REM You may want to REM out the System call if you are
9010 REM working interactively with blassic.
9999 SYSTEM
```

Part 3: Add conditional programming to process error conditions and special values.



Save your program.

Run your program, test to see if the program does what is expected if the user enters 0 for A.

Test your program to make sure that it runs with valid values.

Read your program. What do you think will happen if the user enters 0 for A, and line 150 said ?

```
150   GOTO 110
```

Read your program. What do you think will happen if the user enters 0 for A, and line 150 said ?

```
150   GOTO 120
```

Part 3: Add conditional programming to process error conditions and special values.

Tasks:

- Add a block of code to see if the calculated value of D is equal to 0, if it is, then tell the user there will be only one Root, and then display only that root.

- Add a block of code to see if the calculated value of D is less than to 0, if it is, then tell the user there will be not be any real roots, and don't let the program calculate E or F or display their values to the user.

Part 4: Add iterations to repeat program segments.

Instead if ending the program and returning to the system, do you think you can change your program to process more input?

How could you get the program to end if the user did not want to find any more quadratic solutions?