Included in this packet:

Menu Short Cut Pictures

Quadratic Formula Program 1.0

Given a, b, c determine if there are two real roots, one real double root or two imaginary root.

Programming Skills:

- Input: How to get information from the user and store it as a variable.
- Data types: float
- Print: How to display a message
- If Statements
- Exponents: In Python use ** instead of ^

Quadratic Formula Program 2.0

Given a, b, and c calculate the roots

Programming Skills:

- Replicating a program: How to make a copy of a file and modify it.
- Variables: How to calculate and store information as a variable.
- Complex Function: Use the complex function to calculate complex values.
- Print: Combine strings and variables in a print statement.

Quadratic Formula Program 3.0

*This program isn't needed to make the investigation program.

Given a, b, and c calculate the roots using a function.

Programming Skills:

- def: Write a function using the def function.

Quadratic Investigation Program

Allows the user to enter a, b and c for 5 equations. Calculate the mean for the roots of each equation. Store the coefficients and the mean in lists. Display the information in a table.

Programming Skills:

- Loops: Use a for loop to repeat code.
- Lists: Create a list. Add to a list. Display list values.
- Print: Use tab (\t) to create columns when printing.

Quadratic Investigation Worksheet

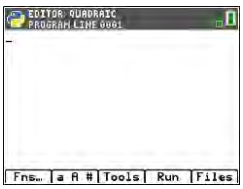
A worksheet that use the quadratic investigation program to develop the formula -b/(2a) for finding the vertex of a parabola.

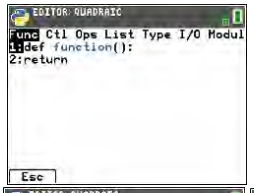
Quadratic Investigation Worksheet Answers

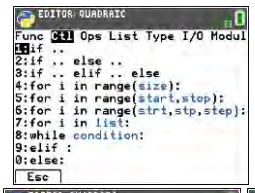
Menu Cheat Sheet:

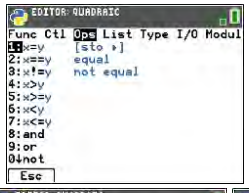
From the home screen

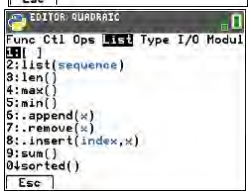
Choose functions

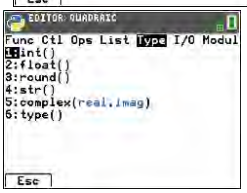


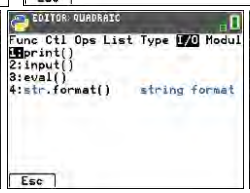


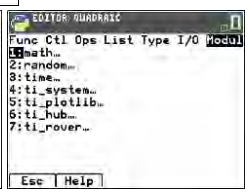




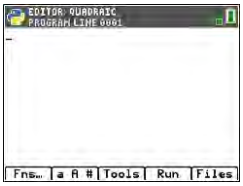


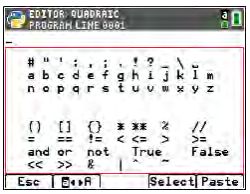




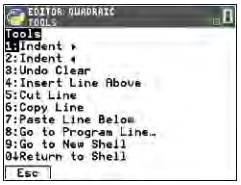


Choose a A #





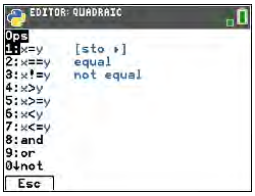
Choose Tools



Notice there is both a copy line and a paste line.

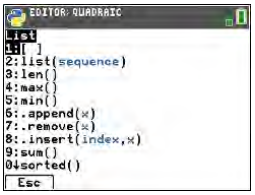
2nd Math – Test menu

Same menu from Fns->Ops



2nd List

Same menu from Fns-> List

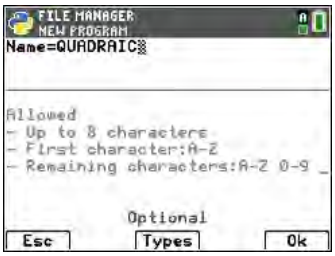


Quadratic Formula 1.0

Start a New Program

File Manager Shows All Files
Select New (Zoom)

Give it a name
Select OK (Graph)

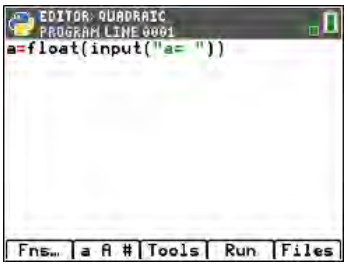


First get the coefficients a,b,c from the user.
In Python, use the command input() to request information.

By default, **var = input("prompt")** will store information as a string variable.
To store data as an integer, use **var = int(input("prompt"))**
To store data as an integer, use **var = float(input("prompt"))**

Everything is menu driven in Python. Instead of typing the commands, use the menus.

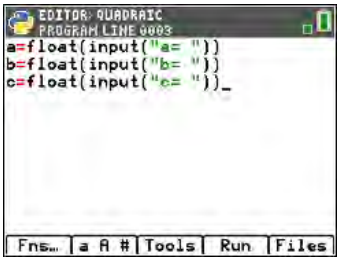
1.) Get coefficient a from the user. Store this value as a float.



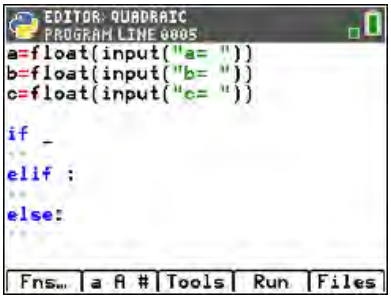
a	[alpha] [a]
=	[2nd] [Math]
float(	Use Fns -> Type
input(	Use fns -> I/O
"	[alpha] [+]

Notice the colors. = is printed in red. Red is used in this context to denote an operation.
"a= " is printed in green. Green is used to show highlight string expressions.

2.) Get coefficient b from the user. Store this value as a float. (Same process as A)
Get coefficient c from the user. Store this value as a float. (Same process as B)



3.) Use an if statement to determine the type of roots.
Python uses tabs to keep statements together for an if statement.



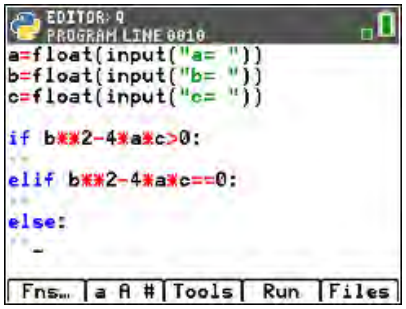
Fns -> Ctrl -> if..elif..else

Notice the template has faint gray diamonds. These represent a tab indent. The indents are used to keep each part of the selection statement separate.

Python uses elif as the syntax for else if.

Notice the color. Blue highlights the selection statement.

4.) Complete the selection statement. $b^2 - 4ac > 0$ and $b^2 - 4ac = 0$.
In python, the ^ is used for bitwise operations. Python uses ** for exponents.
4ac must be written as 4*a*c.
= means store in python
== means comparison equal. Use == for the second comparison.

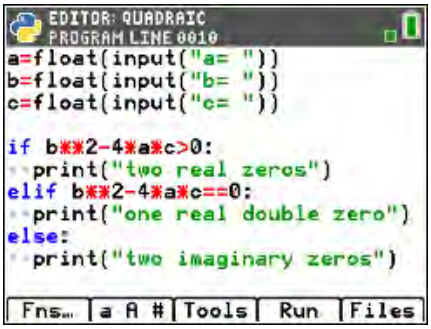


> fns->Ops or 2nd [Math]

== fns->Ops or 2nd [Math]

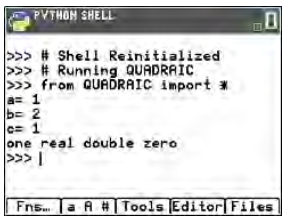
Notice the red color for **, > and ==. Red denotes an operation.

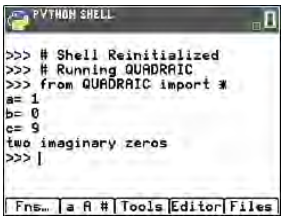
5.) Use a print statement to display the type of zeros.



print(fns->Ops

6.) Run the program [trace]





Quadratic Formula 2.0

This version will calculate the zeros.

1.) Replicate the original quadratic program. Replicating (duplicating) the program will allow us to keep both the 1.0 and the 2.0 version.

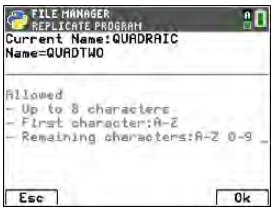
Go back to the File Manager.

If you exited python, select [prgm] -> Python

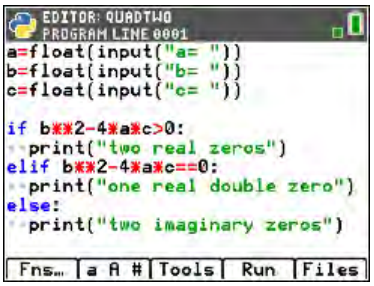
Make sure the arrow is pointing to the Quadratic program

2. Choose Replicate

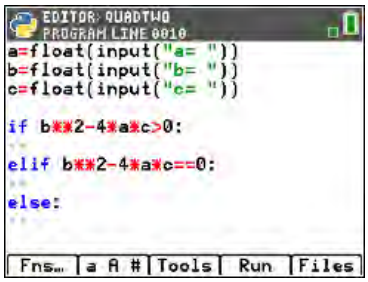
3. Rename the program: QUADTWO



4. Make sure the program looks correct

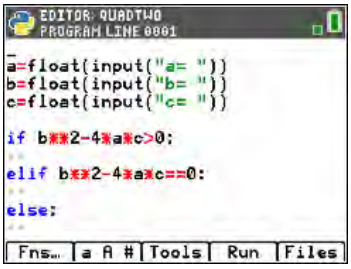


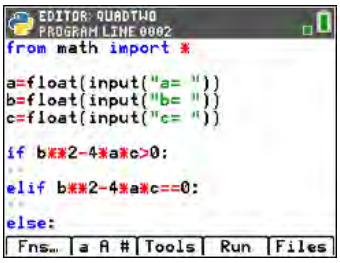
Delete the print statement



If the cursor is on the row with the print(statement, press [clear]

5. To calculate the zeros, you'll need the square root function. Python requires the math library to execute the square root function. Go to the first line, hover over the a. Press [enter] to get a new line. Then get the math library from Fns -> Module.





Fns -> Module

from math import *

6. b**2 – 4ac > 0 has two zeros. Calculate the zeros.

```
EDITOR: QUADTWO
PROGRAM LINE 0012
b=float(input("b= "))
c=float(input("c= "))

if b**2-4*a*c>0:
  r1=(-b+sqrt(b**2-4*a*c))/(2*a)

  r2=(-b-sqrt(b**2-4*a*c))/(2*a)

elif b**2-4*a*c==0:

else:
Fns… | a A # | Tools | Run | Files
```

= Fns -> Ops or 2nd [math]

sqrt(Fns->Module->Math

After typing the r1 statement, press [enter] to generate the line for r2. Notice this line is also tabbed in two diamonds. This alignment keeps these statements together in the one if.

7. b**2 – 4ac = 0 has one double zero. Calculate the zeros.

```
EDITOR: QUADTWO
PROGRAM LINE 0011

if b**2-4*a*c>0:
  r1=(-b+sqrt(b**2-4*a*c))/(2*a)

  r2=(-b-sqrt(b**2-4*a*c))/(2*a)

elif b**2-4*a*c==0:
  r1=(-b)/(2*a)
  r2=r1
else:
Fns… | a A # | Tools | Run | Files
```

8. b**2 – 4ac < 0 has two imaginary numbers. Calculate the zeros

```
EDITOR: QUADTWO
PROGRAM LINE 0009
  r2=_-b-sqrt(b**2-4*a*c))/(2*a)

elif b**2-4*a*c==0:
  r1=(-b)/(2*a)
  r2=r1
else:
  r1=complex(-b,sqrt(-b**2+4*a*c
     ))/(2*a)
  r2=complex(-b,-sqrt(-b**2+4*a*
     c))/(2*a)
Fns… | a A # | Tools | Run | Files
```

Complex(2nd [0] – catalog function

The portion under the square root needs to be positive. You can switch the signs as shown in the picture, or take the absolute value of the quantity before the square root function.

9. Print the roots

```
EDITOR: QUADTWO
PROGRAM LINE 0019
  r1=(-b)/(2*a)
  r2=r1
else:
  r1=complex(-b,sqrt(-b**2+4*a*c
     ))/(2*a)
  r2=complex(-b,-sqrt(-b**2+4*a*
     c))/(2*a)

print("root ",r1)
print("root ",r2)
Fns… | a A # | Tools | Run | Files
```

The string “root” will be displayed as well as the root variable.

The comma in the print statement allows strings and variables to be printed in the same statement. The comma will add a space between the string and the variable when printed.

10. Test the program

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running QUADTWO
>>> from QUADTWO import *
a= 2
b= 5
c= -12
root  1.5
root  -4.0
>>> |
Fns… | a A # | Tools | Editor | Files
```

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running QUADTWO
>>> from QUADTWO import *
a= 1
b= 2
c= 1
root  -1.0
root  -1.0
>>> |
Fns… | a A # | Tools | Editor | Files
```

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running QUADTWO
>>> from QUADTWO import *
a= 1
b= -2
c= 10
root  (1+3j)
root  (1-3j)
>>> |
Fns… | a A # | Tools | Editor | Files
```

Quadratic Formula 3.0

This version will calculate the zeros using a definition for easy repetitive use.

1.) Replicate the second quadratic program. Go back to the File Manager.

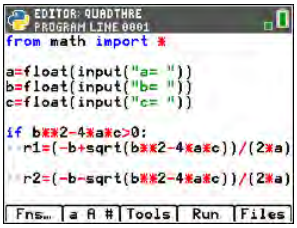
If you exited python, select [prgm] -> Python

Make sure the arrow is pointing to the Quadratic program

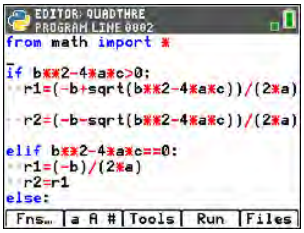
2. Choose Replicate

3. Rename the program: QUADTHRE

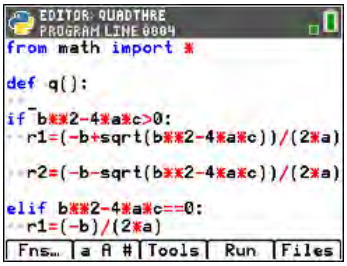
4. Make sure the program looks correct



5. Delete the a, b, c input lines.



6. Insert a definition between the math import and the if statement



Fns-> select def

Type q in front of the (): This names the defined function q.
Notice the line below def q(): has two diamonds indicating everything in the function needs to be indented one space.

7. Our function takes 3 inputs a, b, c. Type a,b,c inside q():

```
EDITOR: QUADTHRE
PROGRAM LINE 0003
from math import *

def q(a,b,c):

if b**2-4*a*c>0:
  r1=(-b+sqrt(b**2-4*a*c))/(2*a)

  r2=(-b-sqrt(b**2-4*a*c))/(2*a)

elif b**2-4*a*c==0:
  r1=(-b)/(2*a)
Fns... | a A # | Tools | Run | Files
```

8. Everything in the definition must be indented. Therefore, the if and everything inside the if must be indented one level. The print will also need to be indented.

```
EDITOR: QUADTHRE
PROGRAM LINE 0003
from math import *

def q(a,b,c):_
  if b**2-4*a*c>0:
    r1=(-b+sqrt(b**2-4*a*c))/(2*
      a)
    r2=(-b-sqrt(b**2-4*a*c))/(2*
      a)
  elif b**2-4*a*c==0:
    r1=(-b)/(2*a)
    r2=r1
```

Remove the extra blank line under the def if you like.

To indent each line, put the cursor on the line
Tools->Indent

```
  else:
    r2=complex(-b,-sqrt(-b**2+4*
      a*c))/(2*a)

  print("root ",r1)
  print("root ",r2)
_
Fns... | a A # | Tools | Run | Files
```

9. Add directions to tell the user what to do when the program is running.

```
EDITOR: QUADTHRE
PROGRAM LINE 0016
  elif b**2-4*a*c==0:
    r1=(-b)/(2*a)
    r2=r1
  else:
    r2=complex(-b,-sqrt(-b**2+4*
      a*c))/(2*a)

  print("root ",r1)
  print("root ",r2)

print("type q(a,b,c)")_
Fns... | a A # | Tools | Run | Files
```

10. Execute the program. You can enter the function multiple times.

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running QUADTHRE
>>> from QUADTHRE import *
type q(a,b,c)
>>> |
Fns... | a A # | Tools | Editor | Files
```

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running QUADTHRE
>>> from QUADTHRE import *
type q(a,b,c)
>>> q(2,-5,-12)
root  4.0
root  -1.5
>>> |
Fns... | a A # | Tools | Editor | Files
```

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running QUADTHRE
>>> from QUADTHRE import *
type q(a,b,c)
>>> q(2,-5,-12)
root  4.0
root  -1.5
>>> q(1,2,1)
root  -1.0
root  -1.0
>>> |
Fns... | a A # | Tools | Editor | Files
```

```
PYTHON SHELL
type q(a,b,c)
>>> q(2,5,-12)
root  1.5
root  -4.0
>>> q(1,2,1)
root  -1.0
root  -1.0
>>> q(1,0,9)
root  3j
root  -3j
>>> |
Fns... | a A # | Tools | Editor | Files
```

Quadratic Formula Investigation Program

This version will let the user investigate the relationship between the average of the roots.
It will use the code from QUADTWO to calculate roots. The investigation will add a loop and a list to the code. This will allow the user to enter multiple values for a,b,c then explore the relationship between the averages of the roots and the coefficients.

1.) Replicate the second quadratic program. Go back to the File Manager.

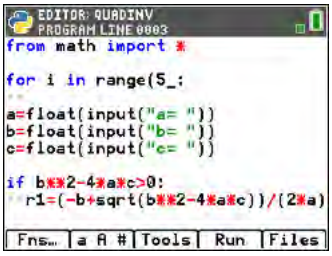
If you exited python, select [prgm] -> Python

Make sure the arrow is pointing to the Quadratic program

2. Choose Replicate

3. Rename the program: QUADINV

4. Add a loop to repeat the collection of a,b,c from the user.

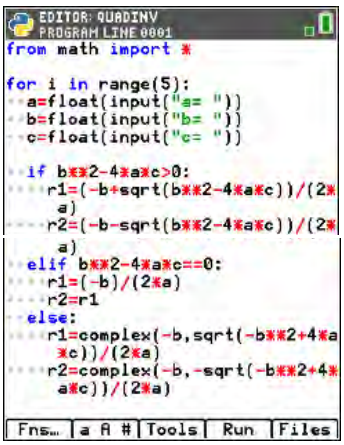


Above the input lines for a,b,c
Fns->Ctl-> for i range

Add a 5 inside the (). This will make the loop execute 5 times.

Notice the 2 diamonds below the for statement. All the repeated code needs to be indented.

5.Indent all the lines under the for loop by one level.



Remove the extra line between the for and the a input.

To indent a line:
Move the cursor to the line
Tools->Indent line

Remove the print statements.

Run the program. Nothing should show. Make sure there aren't any errors.

6. Insert 4 lists to keep track of all the values for a, b, c and the mean of the roots.

```
EDITOR: QUADINV
PROGRAM LINE 0006
from math import *

al=[]
bl=[]
cl=[]
m=[]
for i in range(5):
  a=float(input("a= "))
  b=float(input("b= "))
  c=float(input("c= "))
Fns… | a A # | Tools | Run | Files
```

= fns->Ops or 2nd [math]

[] fns->List or 2nd [stat]

7. After the if statement, but still part of the loop, append(add) the values to the list.

```
EDITOR: QUADINV
PROGRAM LINE 0007
for i in range(5):
  a=float(input("a= "))
  b=float(input("b= "))
  c=float(input("c= "))

  if b**2-4*a*c>0:
    r1=(-b+sqrt(b**2-4*a*c))/(2*
      a)
    r2=(-b-sqrt(b**2-4*a*c))/(2*
      a)
  elif b**2-4*a*c==0:
```

```
    r1=(-b)/(2*a)
    r2=r1
  else:
    r1=complex(-b,sqrt(-b**2+4*a
      *c))/(2*a)
    r2=complex(-b,-sqrt(-b**2+4*
      a*c))/(2*a)
  al.append(a)
  bl.append(b)
  cl.append(c)
  m.append((r1+r2)/2)
Fns… | a A # | Tools | Run | Files
```

.append(Fns->List or 2nd [stat]

Notice, the tab on the append statements. There are only two diamonds because it is part of the for not part of the if.

8. Display the values for a, b, c in a table format. To make it a table, we'll use \t for a tab.
The table will be displayed after all the data is collected and calculated. Therefore, it will not be part of the current loop.

```
EDITOR: QUADINV
PROGRAM LINE 0026
      *c))/(2*a)
    r2=complex(-b,-sqrt(-b**2+4*
      a*c))/(2*a)
  al.append(a)
  bl.append(b)
  cl.append(c)
  m.append((r1+r2)/2)

print("")
Fns… | a A # | Tools | Run | Files
```

Type the heading: a A # [window]

```
EDITOR: QUADINV
PROGRAM LINE 0019
a\t\tb\t\tc\t\tm"_

  # ␣ ' : , ; . ! ? _ \ ~
  a b c d e f g h i j k l m
  n o p q r s t u v w x y z

  () [] {} * ** % //
  = == != < <= > >=
  and or not True False
  << >> & | ^ ~
Esc | ⊡◂▸A | Select | Paste
```

Type the values needed then paste
To format the header with the numbers, use two tabs, \t\t, for the heading.

```
EDITOR: QUADINV
PROGRAM LINE 0019
    r1=complex(-b,sqrt(-b**2+4*a
      *c))/(2*a)
    r2=complex(-b,-sqrt(-b**2+4*
      a*c))/(2*a)
  al.append(a)
  bl.append(b)
  cl.append(c)
  m.append((r1+r2)/2)

print("a\t\tb\t\tc\t\tm")
Fns… | a A # | Tools | Run | Files
```

Notice the \t for the tab appears in brown.

9. Use a loop to print the a,b,c,m list.

```
EDITOR: QUADINV
PROGRAM LINE 0020
    a*c))/(2*a)
  al.append(a)
  bl.append(b)
  cl.append(c)
  m.append((r1+r2)/2)

print("a\t\tb\t\tc\t\tm")
for i in range(5):
  print(str(al[i])+"\t"+str(bl[i
    ])+"\t"+str(cl[i])+"\t"+str
    (m[i]))
Fns… | a A # | Tools | Run | Files
```

for Fns->Ctr

str Fns->Type

\t a A # [window]

10. Execute the program to ensure everything lines up and runs without errors.

```
EDITOR: QUADINV
PROGRAM LINE 0001
_rom math import *

al=[]
bl=[]
cl=[]
m=[]
for i in range(5):
  a=float(input("a= "))
  b=float(input("b= "))
  c=float(input("c= "))
  if b**2-4*a*c>0:
    r1=(-b+sqrt(b**2-4*a*c))/(2*
      a)
    r2=(-b-sqrt(b**2-4*a*c))/(2*
      a)
  elif b**2-4*a*c==0:
    r1=(-b)/(2*a)
    r2=r1
  else:
    r1=complex(-b,sqrt(-b**2+4*a
      *c))/(2*a)
    r2=complex(-b,-sqrt(-b**2+4*
      a*c))/(2*a)
  al.append(a)
  bl.append(b)
  cl.append(c)
  m.append((r1+r2)/2)
print("a\t\tb\t\tc\t\tm")
for i in range(5):
  print(str(al[i])+"\t"+str(bl[i
    ])+"\t"+str(cl[i])+"\t"+str
    (m[i]))
```

```
PYTHON SHELL
c= 2
a= 2
b= 3
c= 4
a     b     c     m
1.0   2.0   3.0   (-1+0j)
1.0   2.0   1.0   -1.0
2.0   3.0   4.0   (-0.75+0j)
4.0   3.0   2.0   (-0.375+0j)
2.0   3.0   4.0   (-0.75+0j)
>>> |
Fns… | a A # | Tools | Editor | Files
```

Use the program to complete the Quadratic Investigation worksheet.

Quadratic Formula Investigation Worksheet

1.) Execute the quadratic formula investigation program QUADINV.

2.) Use your program to find the missing mean values for the table below.
Remember, your program uses the coefficients a, b and c in the equation $y = ax^2 + bx + c$.
The program calculates the two zeros for the function, then finds the mean (m).

a	b	c	m
1	5	6	
1	-4	-21	
1	-2	-24	
1	10	16	
1	-12	20	

3.) Make a conjecture: How can you use the coefficients a, b and c to calculate the mean of the zeros?

4.) Use your program to find the missing values for the table below.
If m is a decimal, convert it to a fraction.

a	b	c	m	m as a fraction
2	-5	-12		
3	5	-2		
-2	7	-5		
-3	5	-2		
1.5	-8	10		

5.) How are the values for a, b, c different in question 4 from those in question 2?

Does your conjecture in part 3 still hold for the values in part 4?

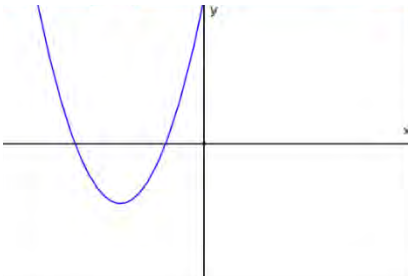
If your conjecture no longer works, modify your conjecture so it works for the values in #2 and #4.

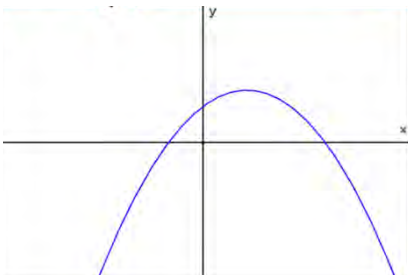
6.) Create 4 equations of your own. At least 2 equations need a values not equal to 1. Use your conjecture to calculate the value of m the mean for the zeros.

Equation	a	b	c	m

7.) Use your program to verify your m values found in problem 6.

8.) Circle the zeros (x-intercepts) on the graphs below.





9.) Put a dot on the x-axis where the **<u>mean of the two roots</u>** would occur.
Draw a vertical line through this point.

10.) What is the name of the vertical line you just drew?

11.) Use your conjecture to complete the following statement.

Given an equation of the form $y = ax^2+bx+c$ where $a \neq 0$, the mean of the x-intercepts (zeros) could be found using the formula _____. This value can be used to write the equation for the axis of symmetry x = _____. It finds the x value for the vertex.

Quadratic Formula Investigation Worksheet- Answer Key

1.) Execute the quadratic formula investigation program QUADINV.

2.) Use your program to find the missing mean values for the table below.
Remember, your program uses the coefficients a, b and c in the equation $y = ax^2 + bx + c$.
The program calculates the two zeros for the function, then finds the mean (m).

a	b	c	m
1	5	6	**-2.5**
1	-4	-21	**2**
1	-2	-24	**1**
1	10	16	**-5**
1	-12	20	**-6**

3.) Make a conjecture: How can you use the coefficients a, b and c to calculate the mean of the zeros?

Answers will vary. Students might say -b because all the a values are 1. This allows the students to find the -b portion on the formula, the next portion will allow them to adjust when a changes.

4.) Use your program to find the missing values for the table below.
If m is a decimal, convert it to a fraction.

a	b	c	m	m as a fraction
2	-5	-12	1.25	**5/4**
3	5	-2	-0.8333	**-5/6**
-2	7	-5	1.75	**7/4**
-3	5	-2	0.83333	**5/6**
1.5	-8	10	2.66667	**8/3**

5.) How are the values for a, b, c different in question 4 from those in question 2?
The a values are not equal to 1 in part 4 like they are in part 2.

Does your conjecture in part 3 still hold for the values in part 4? **No if they just did -b/2**

If your conjecture no longer works, modify your conjecture so it works for the values in #2 and #4.

-b/(2a)

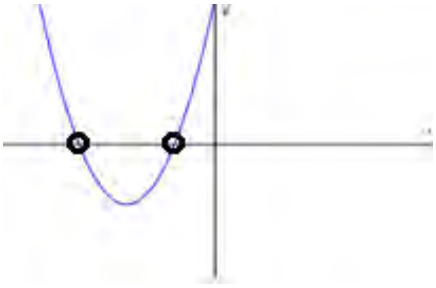
6.) Create 4 equations of your own. At least 2 equations need a values not equal to 1. Use your conjecture to calculate the value of m the mean for the zeros.

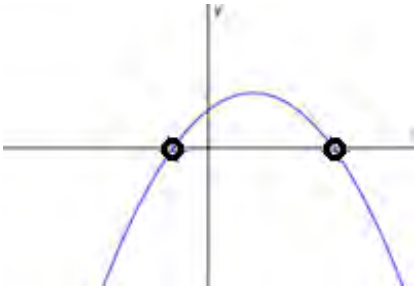
Equation	a	b	c	m

Answers will vary.

7.) Use your program to verify your m values found in problem 6.

8.) Circle the zeros (x-intercepts) on the graphs below.





9.) Put a dot on the x-axis where the **mean of the two roots** would occur.
Draw a vertical line through this point.

10.) What is the name of the vertical line you just drew?

Axis of symmetry

11.) Use your conjecture to complete the following statement.

Given an equation of the form $y = ax^2+bx+c$ where $a \neq 0$, the mean of the x-intercepts (zeros) could be found using the formula __**-b/(2a)**___. This value can be used to write the equation for the axis of symmetry x = ___**-b/(2a)**___. It finds the x value for the vertex.