

## Integer Quadratic Zeros

In this lesson, you will learn how to find the x-intercepts for equations in the form  $y = (x - x_1)(x - x_2)$ .

You will explore how factored equations in the form  $y = (x - x_1)(x - x_2)$  can be rewritten in a standard form  $y = x^2 + bx + c$ .

You will create a game that will let you practice finding integer x-intercepts for equations of the form  $y = x^2 + bx + c$ .

### Objectives:

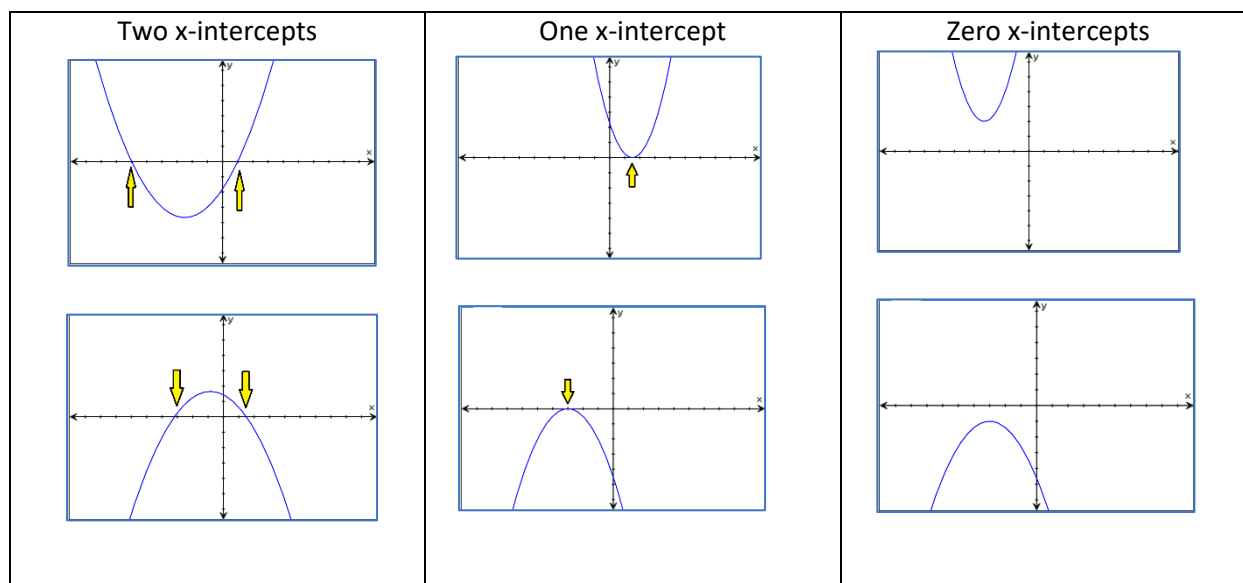
#### Programming Objectives:

- Use the input function and a variable to collect and store data from a user
- Use the randint() function to generate random integers.
- Use a while loop to repeat code
- Use if..elif..else statements to make decisions.

#### Math Objectives:

- Explore how x-intercepts are related to factored quadratic equations
- Explore how to factor equation in standard form
- Factor quadratic equations with integer solutions

The standard form of a quadratic equation is  $y = ax^2 + bx + c$  where a, b, and c are real numbers. Quadratic equations can cross the x-axis twice, once or not at all. The value of a, b and c determine the number of x-intercepts.



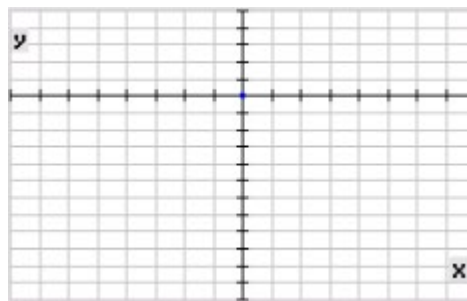
If the graph of the quadratic equation crosses the x-axis, the x values for the x-intercepts can be any integer such as 3, -5 and 0 or any other real number such as 1.3,  $\frac{-5}{7}$ , or  $\sqrt{3}$ . This activity will focus on quadratic equations with integer x-intercepts.

**Explorations:**

1. Graph the equation  $y = (x + 4)(x - 2)$  on your calculator.

Copy the graph on the grid to the right.

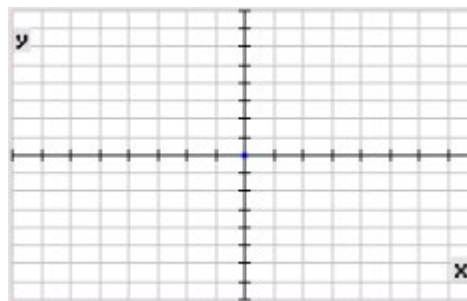
Where are the x-intercepts? \_\_\_\_\_



2. Graph the equation  $y = (x - 1)(x - 3)$  on your calculator.

Copy the graph on the grid to the right.

Where are the x-intercepts? \_\_\_\_\_



3. Look at your x-intercepts and the equations for the previous two problems. Do you notice similarities between the x-intercepts and the equations?

4. Let  $y = (x + 5)(x - 1)$ . Use your results from above to predict where the x-intercepts will be on the graph.

5. Why does this relationship exist between the factored form of an equation and the x-intercepts? The answer lies with the number 0.

List two numbers that multiply to zero. \_\_\_\_ and \_\_\_\_

List another pair of numbers that multiply to zero. \_\_\_\_ and \_\_\_\_

List another pair of numbers that multiply to zero. \_\_\_\_ and \_\_\_\_

Look at your answers above.

*Did you list zero in each pair?*

6. If two numbers multiply to zero, at least one of those two numbers must be a zero.

You will use this fact to find the x-intercepts for quadratics.



The x-intercepts have a y-value of 0.

Therefore, the x-intercepts occur at  $0 = (x + 5)(x - 1)$ .

The value of  $(x + 5)$  times  $(x - 1)$  must equal 0.

Using this fact, either  $x + 5 = 0$  or  $x - 1 = 0$ .

Solve each equation for x.     $x = \underline{\hspace{2cm}}$                        $x = \underline{\hspace{2cm}}$

Compare these values with your predictions from #4.

7. If the x-intercepts are at  $x = -2$  and  $x = 7$ , what is the factored form for the quadratic equation?

$$y = ( \quad ) ( \quad )$$

Graph your equation and verify it crosses the x-axis at  $x = -2$  and  $x = 7$ .

How do you find the x-intercept for a quadratic if it is not in factored form?

8. You know that if a graph has x-intercepts at  $x = 6$  and  $x = -3$ , you can write it as

$$y = (x - 6)(x + 3)$$

Use distribution to write the equation in standard form.

standard form  $y = \underline{\hspace{4cm}}$

Graph your standard form equation on your calculator.

Does it have x-intercepts at  $x = 6$  and  $x = -3$ ?

Compare the standard form with the factored form.

How does the coefficient for x in the standard form relate to the x-intercepts?

How does the constant in the standard form compare to the x-intercepts?



**After completing the activity above, you know:**

Given two real x-intercepts,  $x_1$  and  $x_2$ , you can write the factored form of a quadratic as

$$y = (x - x_1)(x - x_2)$$

You can distribute the values in this equation to get the standard form  $y = x^2 + bx + c$ .

- How do the x-intercepts relate to the constant b?
- How do the x-intercepts relate to the constant c?

**You will use this to create a quadratic factoring game.**

The game will generate a quadratic equation with integer x-intercepts. It will display the quadratic equation and ask for the x-intercepts.

**The user will:**

\*factor the given equation

$$y = (x - 8)(x + 6)$$

\*Find and enter the x-intercepts.

$$x = 8 \text{ and } x = -6$$

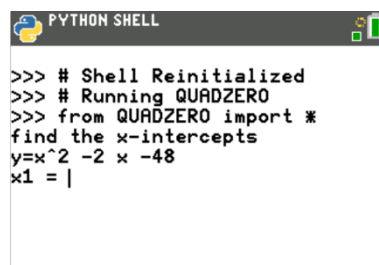
If the user is correct, say correct.

If the user is incorrect, tell the user the correct values for the zeros.

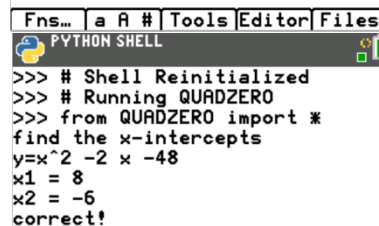
$$y = (x - 3)(x - 4)$$

$$x = 3 \text{ and } x = 4$$

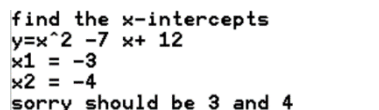
The user should have entered positive 3 and positive 4.



```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running QUADZERO
>>> from QUADZERO import *
find the x-intercepts
y=x^2 -2 x -48
x1 = |
```



```
Fns... | a A # | Tools | Editor | Files
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running QUADZERO
>>> from QUADZERO import *
find the x-intercepts
y=x^2 -2 x -48
x1 = 8
x2 = -6
correct!
```



```
find the x-intercepts
y=x^2 -7 x+ 12
x1 = -3
x2 = -4
sorry should be 3 and 4
```

11. Create a new Python project.

[prgm]

[zoom] The zoom button opens the [New] menu

Name the project “QUADZERO”.

You will need to generate random values for the x-intercepts. Therefore, choose the type “Random Simulation”. This will include one of the libraries you need

12. You should have the image to the right. If you didn't select “Random Simulation” from the menu, you'll have a blank screen.

The line with the hashtag, #, is a comments

(Did you know another name for the # symbol is octothorpe?)

Comments allow the programmer to label sections of the code, making it easier to read. Comments are optional, they do not change the way the program executes.

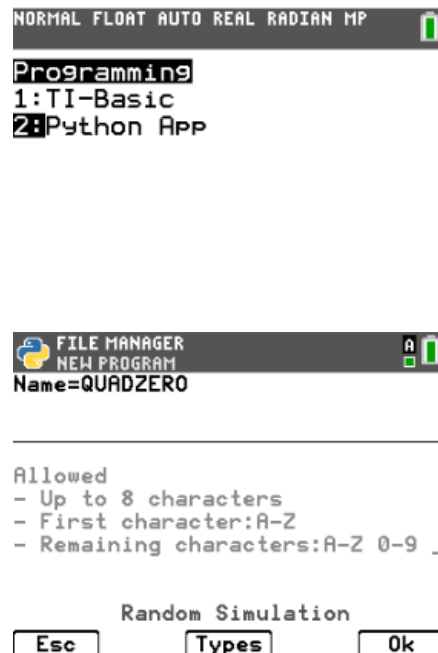
The line that start with the word “from” imports a library. Without this library, you won't be able to generate random integers for the random problems.

If you didn't select Random Simulations, you can import this library by using the following steps.

**Fns> Module > Random> from random import \***

## INTEGER QUADRATIC ZEROS

### TEACHER NOTES



# Math Explorations with Python

## TI-84 PLUS CE PYTHON TECHNOLOGY

13. In order to play the game, you will need to generate two random numbers for x-intercepts. The line
- $$x1 = \text{randint}(-10,10)$$
- will generate one random integer from -10 to 10 and store it in the variable x1.

You don't need to type all of the code using the keyboard.

The randint() function can be found using

**Fns** → **Module** → **Random** → **randint(min,max)**

Can you add a line of code to generate another integer value between -10 and 10 and store that value in x2?

Hint: The [Tools] menu has an option to copy and paste. If you choose to copy and paste, make sure to change the second variable to an x2.

14. Frequently checking your code to see if it works properly is a good programming practice. Add a print statement to print the values of x1 and x2 to the screen.

**Fns** > **I/O** > **print()**

After adding the line print(x1,x2) execute your code in the Python shell.

Activate the [Run] menu by pressing the [trace] button

Does your code generate two random integers between the values of -10 and 10?

In the example on the right, the numbers -4 and -5 were randomly generated. Run the code a few more times to generate more random number pairs. (Press the [trace] button from this screen to get back to the editor.)

## INTEGER QUADRATIC ZEROS

### TEACHER NOTES



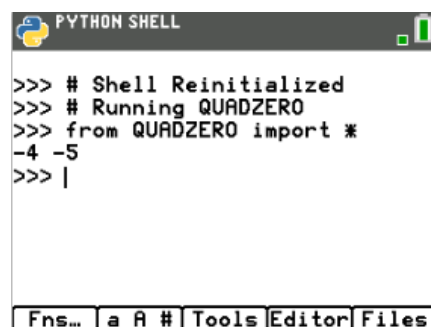
```
EDITOR: QUADZERO
PROGRAM LINE 0004
# Random Simulation
from random import *
x1=randint(-10,10)
```

Fns... a A # Tools Run Files



```
EDITOR: QUADZERO
PROGRAM LINE 0006
# Random Simulation
from random import *
x1=randint(-10,10)
x2=randint(-10,10)
print(x1,x2)
```

Fns... a A # Tools Run Files



```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running QUADZERO
>>> from QUADZERO import *
-4 -5
>>> |
```

Fns... a A # Tools Editor Files

### Teacher Tip:

The randint function generates integers from -10 to 10. Therefore, periodically one or both x-intercepts will be zero.

15. Now that you have verified the code runs, remove the print(x1,x2) line from your code.
16. Recall the quadratic equation can be written in factored form and standard form.

Factored Form

$$y = (x - x1)(x - x2)$$

Standard Form

$$y = ax^2 + bx + c$$

How is the coefficient “b” related to the factored form?

How is the coefficient “c” related to the factored form?

17. Recall, using distribution you get

$$y = (x - x_1)(x - x_2)$$

$$y = x^2 - x_1x - x_2x + x_1x_2$$

$$y = x^2 + (-x_1 - x_2)x + x_1x_2$$

Therefore,  $b = -x_1 - x_2$  and  $c = x_1x_2$

Code the two lines:

```
b = -x1 - x2
```

```
c = x1 * x2
```

18. Use the print statement to print the equation.

Recall in step 4, you found print by using the following steps.

Fns → I/O → print()

You will separate string such “x^2” and variables such as b with a comma.

Add the following lines:

```
print("find the x-intercept")
```

```
print("y=x^2 +",b,"x +",c)
```

\*Hint: The ^ key will add \*\* to the screen. In Python, \*\* is the syntax for exponents. To get the ^ symbol, press the [window] button which will open the [a A #] menu.

\*Notice string inside quotes such as “find the zeros for” and “y=x^2” appear in green. Math operations such as =, \*, -, + appear in red.

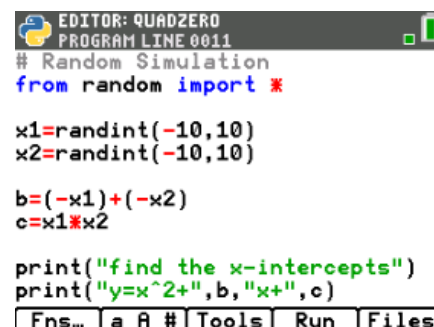


```

EDITOR: QUADZERO
PROGRAM LINE 0009
# Random Simulation
from random import *

x1=randint(-10,10)
x2=randint(-10,10)

b=(-x1)+(-x2)
c=x1*x2
-
    
```



```

EDITOR: QUADZERO
PROGRAM LINE 0011
# Random Simulation
from random import *

x1=randint(-10,10)
x2=randint(-10,10)

b=(-x1)+(-x2)
c=x1*x2

print("find the x-intercepts")
print("y=x^2+",b,"x+",c)
    
```





# Math Explorations with Python

## TI-84 PLUS CE PYTHON TECHNOLOGY

### INTEGER QUADRATIC ZEROS

### TEACHER NOTES

19. Run your program a few times.

Notice if the b or c coefficient is negative, it displays a plus (+) followed by a negative (-).

The example on the right shows:

$$x^2 + 1x + -2 = 0$$

How could you alter the code so it would display only the minus sign if the operation was add a negative?

$$x^2 + 1x - 1 = 0$$

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running QUADZERO
>>> from QUADZERO import *
>>> find the x-intercepts
>>> y=x^2+ 1 x+ -2
>>> |
  
```

Fn... a A # Tools Editor Files

20. There are 4 different cases to consider when cleaning up the display code:

If b and c are both positive:

display doesn't need to change

If b and c are both negative:

change to + - displays to a -

If only c is negative:

change the second + - display to a -

If only b is negative:

change the first + - display to a -

Try fixing the + - print display by writing an if statement in the form:

if

elif

elif

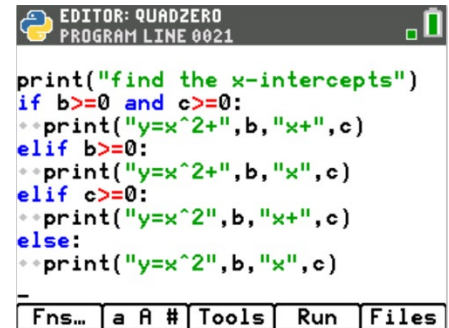
else

**\*if Fns> Ctl**

21. Did you write something similar to:

```

if b >= 0 and c >= 0:
    print("y=x^2 +",b,"x +",c)
elif b >= 0:
    print("y=x^2 +",b,"x ",c)
elif c >= 0:
    print("y=x^2 ",b,"x +",c)
else:
    print("y=x^2 ",b,"x ",c)
    
```



```

EDITOR: QUADZERO
PROGRAM LINE 0021

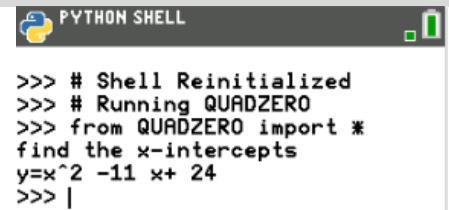
print("find the x-intercepts")
if b>=0 and c>=0:
    print("y=x^2+",b,"x+",c)
elif b>=0:
    print("y=x^2+",b,"x",c)
elif c>=0:
    print("y=x^2",b,"x+",c)
else:
    print("y=x^2",b,"x",c)
    
```

**Teacher Tip:** In Python, indentation is key to if elif statements. Notice there are two diamonds that represent a space for each statement beneath an if, elif or else statement. If students deviate from the two space format the program will not compile.

22. Execute your program at least 5 times.

Verify the statements printed do not contain + -.

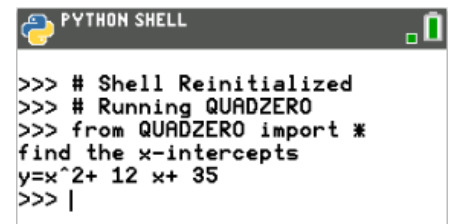
The screen to the right shows three random examples.



```

PYTHON SHELL

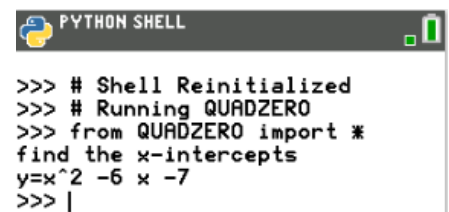
>>> # Shell Reinitialized
>>> # Running QUADZERO
>>> from QUADZERO import *
find the x-intercepts
y=x^2 -11 x+ 24
>>> |
    
```



```

PYTHON SHELL

>>> # Shell Reinitialized
>>> # Running QUADZERO
>>> from QUADZERO import *
find the x-intercepts
y=x^2+ 12 x+ 35
>>> |
    
```



```

PYTHON SHELL

>>> # Shell Reinitialized
>>> # Running QUADZERO
>>> from QUADZERO import *
find the x-intercepts
y=x^2 -6 x -7
>>> |
    
```

23. Notice the first equation in the problem above displays

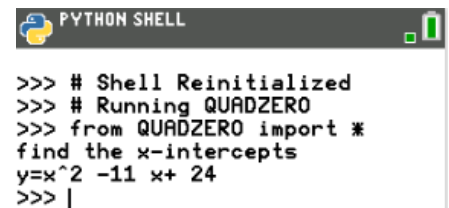
$$x^2 - 11x + 24 = 0$$

Which condition was executed i,ii, iii or iv?

- i) if b >= 0 and c >= 0:
 

```
print("y=x^2 +",b,"x +",c)
```
- ii) elif b >= 0:
 

```
print("y=x^2 +",b,"x",c)
```



```

PYTHON SHELL

>>> # Shell Reinitialized
>>> # Running QUADZERO
>>> from QUADZERO import *
find the x-intercepts
y=x^2 -11 x+ 24
>>> |
    
```



```

iii) elif c >= 0:
    print("x^2",b,"x +",c,"=0 ")
iv) else:
    print("x^2",b,"x",c,"= 0")

```

24. Notice the second equation in #12 displayed

$$x^2 + 12x + 35 = 0$$

Which condition was executed i,ii, iii or iv?

```

i) if b >= 0 and c >= 0:
    print("y=x^2 +",b,"x +",c)
ii) elif b >= 0:
    print("y=x^2 +",b,"x",c)
iii) elif c >= 0:
    print("x^2",b,"x +",c,"=0 ")
iv) else:
    print("x^2",b,"x",c,"= 0")

```

25. Your program generates two x-intercepts, then calculates and displays the standard quadratic equation. Now you need to ask the user to enter the two x-intercepts.

The command input() will let you ask a question. By default, the information from an input() statement is stored as string. That means the computer thinks the information entered is a list of characters. To store information as integers, you must use int(input()) to convert the input to an integer.

int() can be found using      Fns → Type  
input() can be found using    Fns → I/O

Add the lines:

```

z1 = int(input("x1 = "))
z2 = int(input("x2 = "))

```

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running QUADZERO
>>> from QUADZERO import *
>>> find the x-intercepts
y=x^2+ 12 x+ 35
>>> |

```

```

EDITOR: QUADZERO
PROGRAM LINE 0014
* print("y=x^2+",b,"x+",c)
elif b>=0:
* print("y=x^2+",b,"x",c)
elif c>=0:
* print("y=x^2",b,"x+",c)
else:
* print("y=x^2",b,"x",c)

z1=int(input("x1 = "))
z2=int(input("x2 = "))

```

Fns... | a A # | Tools | Run | Files



# Math Explorations with Python

## TI-84 PLUS CE PYTHON TECHNOLOGY

26. Recall  $x_1$  and  $x_2$  have the real  $x$ -intercepts. The variables  $z_1$  and  $z_2$  have the user's entered  $x$ -intercepts. Now you need to check to see if they are equivalent. The order the user types the  $x$ -intercepts does not matter. Therefore, the user is correct:

if  $x_1$  equals  $z_1$  and  $x_2$  equals  $z_2$  or  $x_1$  equals  $z_2$  and  $x_2$  equals  $z_1$

In python, the equals sign by itself is used to store a value. To check if two values are equivalent, use two equal signs ==.

Add the following lines of code:

```
if (x1 == z1 and x2 == z2) or (x1 == z2 and x2 == z1):
    print("Correct!")
else:
    print("Sorry should be",x1,"and",x2)
```

27. Test your program. **Run**

Sample Test:

$$y = x^2 - 9x + 14$$

$$(x - 7)(x - 2) = 0$$

$$x = 7 \quad x = 2$$

28. Your program will now let you practice finding the  $x$ -intercepts for a quadratic equation. You could stop coding here. However, each time you want to generate a question you have to press [trace] to make the program Run

You can add a loop to your program to repeat the code. There are multiple ways to write a loop. You will use

## INTEGER QUADRATIC ZEROS

### TEACHER NOTES

```
EDITOR: QUADZERO
PROGRAM LINE 0027
print("y=x^2",b,"x",c)

z1=int(input("x1 = "))
z2=int(input("x2 = "))

if (x1==z1 and x2==z2) or (x1==z2 and x2==z1):
    print("correct!")
else:
    print("sorry should be",x1,"and",x2)
```

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running QUADZERO
>>> from QUADZERO import *
find the x-intercepts
y=x^2 -9 x+ 14
x1 = |
```

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running QUADZERO
>>> from QUADZERO import *
find the x-intercepts
y=x^2 -9 x+ 14
x1 = 7
x2 = 2
correct!
>>> |
```

```
EDITOR: QUADZERO
PROGRAM LINE 0004
# Random Simulation
from random import *

for i in range():
    x1=randint(-10,10)
    x2=randint(-10,10)

    b=(-x1)+(-x2)
    c=x1*x2
```

```
for i in range(size):
```

Which will let you determine the number of times the code repeats.

You need to repeat everything except for the library import at the top of the code.

Move your cursor above the line that says:

```
x1 = randint(-10,10)
```

Add a for loop:

```
Fns> Ctl> for i in range(size):
```

29. You could repeat your code as many times as you like.

For now, have the code repeat 5 times.

Inside the range() function, add 5.

If you run the program now, you will have an error.

Notice under the for statement, two diamonds  $\diamond\diamond$  have appeared.

Python uses indentation to group all lines of code for a loop.

Currently, the only line that is part of the loop, is the empty line directly below the for statement. All your lines below the for statement should be repeated. Therefore, they will all need indented.

```
EDITOR: QUADZERO
PROGRAM LINE 0006
# Random Simulation
from random import *

for i in range(5):
  **
  x1=randint(-10,10)
  x2=randint(-10,10)

b=(-x1)+(-x2)
c=x1*x2
```

30. Move your cursor to the line

```
x1=randint(-10,10)
```

To indent the line:

```
Tools> Indent
```

Notice now, this line has two diamond infront indicating the indentation.

```
EDITOR: QUADZERO
PROGRAM LINE 0006
# Random Simulation
from random import *

for i in range(5):
  **
  *x1=randint(-10,10)
  x2=randint(-10,10)

b=(-x1)+(-x2)
c=x1*x2
```



31. Use the same procedure to indent all the lines below the for statement.

Notice the indentation of the if statements. The print lines under the if, elif, else lines are now indented 4 diamonds. Two diamonds for the loop and two diamonds for the if.

### INTEGER QUADRATIC ZEROS TEACHER NOTES

```

EDITOR: QUADZERO
PROGRAM LINE 0001
# Random Simulation
from random import *

for i in range(5):
    *
    *x1=randint(-10,10)
    *x2=randint(-10,10)

    *b=(-x1)+(-x2)
    *c=x1*x2

    *print("find the x-intercepts")

    *if b>=0 and c>=0:
    *   *print("y=x^2+",b,"x+",c)
    *elif b>=0:
    *   *print("y=x^2+",b,"x",c)
    *elif c>=0:
    *   *print("y=x^2",b,"x+",c)
    *else:
    *   *print("y=x^2",b,"x",c)

    *z1=int(input("x1 = "))
    *z2=int(input("x2 = "))

    *if (x1==z1 and x2==z2) or (x1==
    *   z2 and x2==z1):
    *   *print("correct!")
    *else:
    *   *print("sorry should be",x1,"
    *       and",x2)_

Fns... | a A # | Tools | Run | Files

```

32. You have completed the first coding project for factoring quadratic equations. Use your program to practice factoring. When you have mastered factoring these types of problems, move on to coding activity "Rational Quadratic Zeros" to add another level of factoring to your game.

#### Teacher Tip:

```

# Random Simulation
from random import *

for i in range(5):
    x1=randint(-10,10)
    x2=randint(-10,10)

    b=(-x1)+(-x2)
    c=x1*x2

    print("find the x-intercepts")

    if b>=0 and c>=0:
        print("y=x^2+",b,"x+",c)
    elif b>=0:
        print("y=x^2+",b,"x",c)
    elif c>=0:
        print("y=x^2",b,"x+",c)

```



```
else:
    print("y=x^2",b,"x",c)

z1=int(input("x1 = "))
z2=int(input("x2 = "))

if (x1==z1 and x2==z2) or (x1==z2 and x2==z1):
    print("correct!")
else:
    print("sorry should be",x1,"and",x2)
```