# Output

User wants the program to loop infinitely until they press 0 to exit

Output will look something like:

“A 14 inch pizza can be cut into 6, 8, or 10 slices. If cut into 6 slices, each slice will be 25.66 inches. If cut into 8 slices, each slice will be 19.24 inches. If cut into 10 slices, each slice will be 15.39 inches.”

User wants decimal values rounded to 2 decimal places

# Input

Need to get the diameter of the pizza from the user and store in diameter variable

Will handle input validation through the use of while loops and if/else statements

Ensure that input is only numerical values within an acceptable range (8” and 24”)

# Process

There is quite a bit of math involved here

First, we obtain the value of the diameter from the user, and store in diameter variable

Assign the value of PI as a constant equal to roughly 3.14

Next, we assign the value of the radius variable by doing diameter / 2

We then find the value of radius\_squared by multiplying radius \* radius

Finally, assign pizza\_area the value equal to radius\_squred \* PI

We will also need to find the value of each slice according to the way the pizza was cut. It will probably be best to assign variables such as slices\_6, slices\_8 and so on. From there, all we will need to do is, for example, slices\_6 = pizza\_area / 6. We will also need to round to 2 digits using the round function.

# Pseudo Code

Declare a constant for the value of PI

Start a while loop so that the user is continuously prompted for new pizzas until they choose to exit the program

Add the ability for user to exit the program by pressing 0. This should be as simple as

if x == “0”

exit ()

Prompt the user for the diameter of their pizza and store input in “diameter” variable

Check to make sure diameter is numeric before we begin processing

Check to make sure diameter is within acceptable range (not smaller than 8” or larger than 24”)

If pizza is a numerical value within acceptable range, we can begin processing. We will need to write an elif statement for every range of pizza size

8” to <12” – 6 slices

12” to <14” – 6 or 8 slices

14” to <16” – 6, 8 or 10 slices

16” to < 20” – 6, 8, 10, or 12 slices

20” to 24” – 6, 8, 10, 12, or 16 slices

From there, we can begin our processing. For example, if we were calculating an 8 inch pizza, we would use slices\_6 = round((pizza\_area / 6), 2) to find the size of each slice, and round it to 2 decimal places.

We can then output a message like “An 8 inch pizza can be cut into 6 slices. Each slice will be {slices\_6} inches.”

# Desk Check

Input: 10

Result: A 10 inch pizza can be cut into 6 slices. Each slice would be 13.09 inches.

Input: 10000

Result: Pizza is too big!

Input: -1

Result: Pizza is too small!

Input: 25000.25

Result: Pizza is too big!

Input: -900

Result: Pizza is too small!

Input: 0.00

Result: Pizza is too small!

Input: 0

Result: Exits the program

Input: Eight

Result: Please enter a number.