## OUTPUT

The user wants to see all hot dogs sold, separated by the type of hot dog. He would also like us to calculate the percentage of each.

Each time the program loops, we will need to output something like:

Traditional Dog:

Veggie Dog:

Curry Dog:

Beside each label, we will give the user the option to input a number. This will either add to the tally or leave it the same.

For example, here would be the output where the customer has already added 36 traditional hot dogs, 15 veggie dogs, and 7 curry dogs. They are also trying to add a traditional dog and curry dog, but not adding any veggie dogs.

Traditional Dog: 1

Veggie Dog: 0

Curry Dog: 1

Total sold:

Traditional Dogs: 37 (61.6%)

Veggie Dogs: 15 (25%)

Curry Dogs: 8 (13.3%)

## INPUT

We will need to indefinitely loop the program, letting the user tally a hot dog every time one is sold.

For example, we could store user input in variables such as traditional\_hot\_dog\_counter, veggie\_hot\_dog\_counter, and curry\_hot\_dog\_counter.

When the user enters a number, it will be stored in the counter variables, which we can add to the total later.

## PROCESS

This is fairly simple math. Each time the program loops, the user will add to the “counter” variables that we will add to the total.

For instance, if the user input a 1 beside the veggie dog line, it would add 1 to the veggie\_hot\_dog\_counter variable.

We can then declare a variable such as veggie\_hot\_dog\_total, and with each loop, add veggie\_hot\_dog\_counter to it.

We will also need to handle the percentage calculations. For this, we will need to take the total of, for example, curry dogs, and divide by the total number of all types of hot dogs, and then multiply by 100.

curry\_dog\_total / all\_hot\_dogs\_sold = some decimal which we will multiply by 100 to get a whole number percentage.

## PSEUDO-CODE

Create a while loop that will allow the user to exit by pressing the “q” key, otherwise this will be an infinite loop. We can do this by using something like “while not q”

Prompt the user for the amount of each hot dog sold and store it in a variable such as traditional\_dogs\_counter, curry\_dogs\_counter, and veggie\_dogs\_counter.

Prompt the user for input: “How many curry dogs were sold?” and store the answer in variable curry\_dogs\_counter

Prompt the user for input: “How many traditional dogs were sold?” and store the answer in variable traditional\_dogs\_sold

Prompt the user for input: “How many veggie dogs were sold?” and store the answer in variable curry\_dogs\_sold

Create a validation check for each input above, using something like isalpha False to check that the user is only inputting numbers.

Create variables named traditional\_dogs\_total, curry\_dogs\_total, and veggie\_dogs\_total.

Add the counter variable to the total variable every loop.

Then, we can output the total amount of each hot dog sold, along with the percentage of each.

To calculate the percentage, we will need to add all totals together to get a total number of all hot dogs sold, and store in a variable such as all\_hot\_dogs\_sold. We can then divide, for example, curry\_dogs\_sold by all\_hot\_dogs\_sold and multiply by 100, and that will be the percent of curry dogs sold.

# DESK-CHECK

**How many veggie dogs were sold?**

10: Adds 10 to the counter

10000: Adds 10000 to the counter

-1: “Input must be a whole, positive number.”

25000.25: “Input must be a whole, positive number.”

-900: “Input must be a whole, positive number.”

0: Adds nothing to the counter.

Eight: “Input must be a whole, positive number.”

**How many traditional dogs were sold?**

10: Adds 10 to the counter

10000: Adds 10000 to the counter

-1: “Input must be a whole, positive number.”

25000.25: “Input must be a whole, positive number.”

-900: “Input must be a whole, positive number.”

0: Adds nothing to the counter.

Eight: “Input must be a whole, positive number.”

**How many curry dogs were sold?**

10: Adds 10 to the counter

10000: Adds 10000 to the counter

-1: “Input must be a whole, positive number.”

25000.25: “Input must be a whole, positive number.”

-900: “Input must be a whole, positive number.”

0: Adds nothing to the counter.

Eight: “Input must be a whole, positive number.”