PRACTICE EXERCISES - LAB 4 (Subroutines)

1. We want to write a simple Python program that prints out the lyrics to the familiar tune *Old McDonald Had A Farm*:

```
Old MacDonald had a farm, E I E I O,
And on his farm he had a cow, E I E I O.
With a moo moo here and a moo moo there,
Here a moo, there a moo, ev'rywhere a moo moo.
Old MacDonald had a farm, E I E I O.

Old MacDonald had a farm, E I E I O,
And on his farm he had a pig, E I E I O.
With a oink oink here and a oink oink there,
Here a oink, there a oink, ev'rywhere a oink oink.
Old MacDonald had a farm, E I E I O.

Old MacDonald had a farm, E I E I O,
And on his farm he had a goose, E I E I O.
With a honk honk here and a honk honk there,
Here a honk, there a honk, ev'rywhere a honk honk.
Old MacDonald had a farm, E I E I O.
```

Here is our program so far:

```
def main():
  printLyrics("cow", "moo")
  print ""
  printLyrics("pig", "oink")
  print ""
  printLyrics("goose", "honk")

def printLyrics(animal, sound):
  # You complete this function

main()
```

Complete the program be writing the `printLyrics` function. This function has two parameters, `animal` and `sound`, where the first parameter is the animal's name (e.g. cow) and the second parameter is the animal's associated sound (e.g. moo).

2. Write a Python program that prints out a table of the Celsius temperatures from 0 to 50 in steps of 5, along with their corresponding Fahrenheit temperatures (as integers). Use a function that converts a temperature given in a parameter from Celsius to Fahrenheit and return its result.

3. Write a program that contains a function that has one parameter, n, representing an integer greater than 0. The function should return n! (n factorial). Then write a main function that calls this function with the values 1 through 20, one at a time, printing the returned results. This is what your output should look like:

```
1      1
2      2
3      6
4      24
5      120
6      720
7      5040
8      40320
9      362880
10     3628800
11     39916800
12     479001600
13     6227020800
14     87178291200
15     1307674368000
16     20922789888000
17     355687428096000
18     6402373705728000
19     121645100408832000
20     2432902008176640000
```

4. Modify the Python program we wrote for Bubble Sort so it asks the user for the number of data values in the array and then reads them in one at a time from the user of the program using the `getData` function discussed in lab.