CSC 2720 Data Structure
Lab 1

Due: 01/14/2024 @ 11:59PM

Requirements:

(Failure to follow the requirements will result in a score of Zero.)

1. You may use whatever IDEs / editors you like, but you must submit your responses on iCollege as **.py files**.
2. Your submissions will work exactly as required.
3. Make sure you submit a file that compiles.
4. Your submission will show an output. Should you recive a Zero for no output shown do not bother to email with "but the logic is perfect".
5. Your program's output must exactly match the specs (design, style) given here for each problem to pass the test cases.
6. Design refers to how well your code is written (i.e., is it clear, efficient, and elegant), while Style refers to the readability of your code (correct indentation, good variable names). Add comments to have necessary explanations for your program.
7. Add a "heading" at the very beginning of your .java files as follow:

   Your Name
   CSc 2720 Lab #N
   Lab time: put your lab time here
   Due time: put the due date here

Exercise 1

(20points) Write a python program called ***PersonalizedHelloWorld*** that prompts a user for their name and then displays "Hello, [name here]!" The flow should look like the following:

```
What is your name?
Please enter your name: Firstname Lastname
Hello, Firstname Lastname!
```

If the user does not enter anything but presses Enter anyways, you should re-prompt for the user's name. This flow should look like the following (note that there should be a space after any "?" or ":"):

```
What is your name?
Please enter your name:
Please enter your name:
Please enter your name: Firstname Lastname
Hello, Firstname Lastname!
```

Exercise 2

(80 points) Write a python program called ***GuessMyNumber*** that prompts the user for an integer n, tells the user to think of a number between 0 and n − 1, then makes guesses as to what the number is. After each guess, the program must ask the user if the number is lower, higher, or correct. You must implement the divide-and-conquer algorithm from class. In particular, you should round up when the middle of your range is in between two integers. (For example, if your range is 0 to 31, you should guess 16 and not 15, but if your range is 0 to 30 you should certainly guess 15). The flow should look like the following:

```
Enter n: 32
Welcome to Guess My Number!
Please think of a number between 0 and 31.
Is your number: 16?
Please enter C for correct, H for too high, or L for too low.
Enter your response (H/L/C): H
Is your number: 8?
Please enter C for correct, H for too high, or L for too low.
Enter your response (H/L/C): L
Is your number: 12?
Please enter C for correct, H for too high, or L for too low.
Enter your response (H/L/C): C
Thank you for playing Guess My Number!
```

As part of your implementation, you should check that n is not 0 or negative. (You need not worry about the case where the user enters a non-integer). You should also check that the user is entering one of the letters H, L, or C each time your program makes a guess. This flow should look like the following:

```
Enter n: -1
Enter a positive integer for n: 32
Welcome to Guess My Number!
Please think of a number between 0 and 31.
Is your number: 16?
Please enter C for correct, H for too high, or L for too low.
Enter your response (H/L/C): asdf
Enter your response (H/L/C): H
Is your number: 8?
...
```

(You can assume that the user will always give honest answers.)