CSCE 110: Programming I

Lab #2 (100 points)

Due: Sunday, September 11th by 11:59pm

Directions: The following lab will help you explore how to write simple Python programs that you save to a file and run within the WingWare IDE 101 environment. Please label your Python programs `q<num>.py`, where `num` is the question number. For example, your solution to the first question, will be stored in the file `q1.py`.

Take your time and make sure you understand everything in this lab before getting started. Also, make sure your programs match the output EXACTLY as given for each question. This is important as one of the keys to being a good programmer is attention to details.

1 Lab Questions

1. Write a program (called q1.py) that modifies the program from the lecture slides that prints the greeting such that the last name is printed first, followed by the first and middle names. Below are examples of output when the program is executed.

   Note: To print an empty line, use `print()`.

   Example #1. At the prompt "Enter first name", the user enters Maria (line 1). The user then enters Elena at the middle name prompt (line 2). When the user is asked for a last name, they enter Perez (line 3). The greeting is printed on line 5.

```
1 Enter first name: Maria
2 Enter middle name: Elena
3 Enter last name: Perez
4
5 Howdy, Perez, Maria Elena!
```

Example #2.

```
Enter first name: Sam
Enter middle name: Paul
Enter last name: Smith

Howdy, Smith, Sam Paul!
```

2. Write a program (called q2.py) that asks for the number of people at a birthday party, where a 32 slice pizza will be served. Your program will output the number of pizza slices each person gets. There are two ways to handle divide the pizza equally. The first way ignores the extra pieces and give everyone the same amount. The second way cuts up the extra pieces so that everyone gets the same amount. Your program must output both options.

   Example #1. At the prompt, the user specifies that there are 10 guests. Then, the two options are shown for splitting the pizza between the 10 guests.

```
Number of guests: 10

Option 1: 3 slices each, 2 left over
Option 2: 3.2 slices each
```

   Example #2. Similar to Example #1 but for 8 guests.

```
Number of guests: 8

Option 1: 4 slices each, 0 left over
Option 2: 4.0 slices each
```

   Example #3. Similar to Example #1 but for 19 guests.

```
Number of guests: 19

Option 1: 1 slices each, 13 left over
Option 2: 1.6842105263157894 slices each
```

3. Write a program (called q3.py) that computes the distance between two points. Suppose the first point is represented by (x_1, y_1) and the second point by (x_2, y_2), then the distance between them is $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

   Note: You can use `a ** 0.5` to compute $\sqrt{a}$.

   Example #1. Here, the user computes the distance between the two points (1.5, -3.4) and (4, 5). Lines 1 and 2 ask the user to enter the values for the first point. Lines 3 and 4 ask for the second point. The distance between the points is shown on line 6.

```
1 Enter x1: 1.5
2 Enter y1: -3.4
3 Enter x2: 4.0
4 Enter y2: 5.0
5
6 Distance between two points:  8.764131445842194
```

   Example #2. Here, the user computes the distance between the points (1.0, 32.4) and (98765, 4.283).

```
1 Enter x1: 1.0
2 Enter y1: 32.4
3 Enter x2: 98765
4 Enter y2: 4.283
5
6 Distance between two points:  98764.00400229676
```

4. Write a program (called q4.py) that displays the following pattern. Your program will consist of print statements containing the characters 'F', 'U', 'N', and ' ' (space). There is no user input for this program.

   Here, the output shows that line 1 consists of seven F's followed by five spaces, a single 'U' followed by five spaces, one 'U' followed by five spaces, two 'N's followed by four spaces, and finally two 'N's. The remaining lines follow similarly although the number of spaces between characters varies. However, the number of spaces can be figured out once the output of line 1 is established.

```
1 FFFFFFF     U     U     NN    NN
2 FF          U     U     NNN   NN
3 FFFFFFF     U     U     NN N  NN
4 FF           U   U      NN  N NN
5 FF             UUU       NN   NNN
```

2 Submitting Your Assignment

Once you have completed your programs, submit each of them (q1.py, q2.py, q3.py, and q4.py) to eCampus. You only get one chance to submit your programs. However, if you make a mistake and it's before the deadline, please contact the instructor or TA so that you can resubmit your programs.