General Instructions:

• **Lab is due online by 11:59 pm of the due date.** The assignment must be typed, not handwritten or scanned.

• 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`.

• 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.

• If you forgot to bring a computer, you can check out a temporary laptop from ZACH 383 for a few hours (depending on the availability).

• Grading will be based on correctness and clarity. **Copying work from another source and submitting it as your own is plagiarism.** The minimum penalty for plagiarism is a zero on this lab.
Lab Instructions

In this lab you will work with the print function, input function and basic math operations and get familiarized with the concept of variables. You can consult the following links to see examples: Print function and Input Function.

Lab Questions

1. Write a Python program (called q1.py) that asks the user for their first name, middle name, and last name. Afterward, the program greets them using Howdy. Below are examples of output when the program is executed.

   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.

   ```python
   Enter first name: Maria
   Enter middle name: Elena
   Enter last name: Perez
   Howdy, Maria Elena Perez!
   ```

   Example #2. Similar to Example #1 but with different input.

   ```python
   Enter first name: Sam
   Enter middle name: Paul
   Enter last name: Smith
   Howdy, Sam Paul Smith!
   ```

2. Write a program (called q2.py) that modifies the program from the previous question and 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.

   ```python
   Enter first name: Maria
   Enter middle name: Elena
   Enter last name: Perez
   Howdy, Maria Elena Perez!
   ```

   Example #2. Similar to Example #1 but with different input.

   ```python
   Enter first name: Sam
   Enter middle name: Paul
   Enter last name: Smith
   Howdy, Sam Paul Smith!
   ```
Enter first name: Maria
Enter middle name: Elena
Enter last name: Perez

Howdy, Perez Maria Elena!

Example #2.

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

Howdy, Smith Sam Paul!

3. Write a program (called q3.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.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Number of guests: 19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Option 1: 1 slices each, 13 left over</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Option 2: 1.6842105263157894 slices each</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. Write a program (called q4.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.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter x1: 1.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Enter y1: -3.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Enter x2: 4.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Enter y2: 5.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Distance between two points: 8.764131445842194</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter x1: 1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Enter y1: 32.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Enter x2: 98765</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Enter y2: 4.283</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Distance between two points: 98764.00400229676</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Write a program (called q5.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.
Submitting Your Assignment

Once you have completed your programs, submit each of them (q1.py, q2.py, q3.py, q4.py and q5.py) electronically.
You may resubmit your files as many times as necessary until the due date. Only the most recent submission will be graded.

You are required to include the following lines in the header of all your files:

```python
# File: filename.py
# Author: Student name
# Date: xx/xx/2019
# Section: Student section number
# E-mail: student_email@tamu.edu
# Description:
# e.g. This program asks for the coordinates of two points and computes the distance
# between the two points.
```

- If you are in section 501 or 502, submit your files on: mimir.io
- If you are in section 503, 504, 505 or 506, submit your files on: gradescope.com