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).

- Before you leave the lab room make sure to checkout with the TA by showing your student ID.

- 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 Questions

1. Write a Python program stored in a file q1.py for a fruit seller to keep track of the fruits and prices in her shop.

   The seller can enter a fruit’s name, followed by its prices in one line. A comma should separate the fruit and the fruit’s prices. Each fruit in this shop should have at least three prices based on the quality of the fruit.

   Your program should print a main menu for the seller to allow her to add, edit, delete, or search an item.

   (a) Note 1: The delete option has two features: delete all the items, or delete one item at a time.

   (b) Note 2: If one item is not in stock and the seller is trying to edit, delete, or search it, your program should return a message to the seller with the name of the fruit.

   (c) Note 3: Your program should keep running until the user quits.

   The program should print the up-to-date stock at the end of every option selected from the menu.

Example:

   (a) The program prints the main menu.

   (b) The user chooses a number from the menu.

   (c) A comma separates the fruit and the prices.

   The sample output showing the behavior of the program is on the next page.
*************Main Menu*************

1. Add fruits
2. Edit fruit
3. Delete fruit
4. Search fruits
5. Quit

Choose from menu: 1
Enter fruit follow by prices: Apple, 0.12 0.23 0.78 1.2
Current Stock: {'Apple': [0.12, 0.23, 0.78, 1.2]}

*************Main Menu*************

1. Add fruits
2. Edit fruit
3. Delete fruit
4. Search fruits
5. Quit

Choose from menu: 1
Enter fruit follow by prices: Orange, 0.45 0.5 0.78
Current Stock: {'Apple': [0.12, 0.23, 0.78, 1.2], 'Orange': [0.45, 0.5, 0.78]}

*************Main Menu*************

1. Add fruits
2. Edit fruit
3. Delete fruit
4. Search fruits
5. Quit

Choose from menu: 2
Enter the fruit name: Apple
Enter the fruit prices: 1.2 2.5 3.1
Current Stock: {'Apple': [1.2, 2.5, 3.1], 'Orange': [0.45, 0.5, 0.78]}

*************Main Menu*************

1. Add fruits
2. Edit fruit
3. Delete fruit
4. Search fruits
5. Quit

Choose from menu: 3
If you want to remove all items enter all otherwise enter the fruit name: Orange
Current Stock: {'Apple': [1.2, 2.5, 3.1]}
1. Add fruits
2. Edit fruit
3. Delete fruit
4. Search fruits
5. Quit

Choose from menu: 4
Enter the fruit name: Banana
Banana is not in the stock.
Current Stock: {'Apple': [1.2, 2.5, 3.1]}

Choose from menu: 4
Enter the fruit name: Apple
Apple has 3 prices: 1.2 2.5 3.1
Current Stock: {'Apple': [1.2, 2.5, 3.1]}

Choose from menu: 5
Current Stock: {'Apple': [1.2, 2.5, 3.1]}
2. **Pattern in DNA sequence.** DNA sequence is represented by using A, T, G and, C repeatedly. It is often required to find the most recurring pattern of specific length in the DNA sequence. Write a Python program stored in a file q2.py that takes a DNA sequence and a pattern length as inputs and prints the most frequent pattern(s) of specified length. Use a Python dictionary to store the patterns and their count.

**Example #1.** The user enters DNA sequence (line 1) and pattern length (line 2). The program then prints the most frequent pattern (line 4 and onward).

```
1 Enter DNA sequence: GTACCCGTACTTAG
2 Enter pattern length: 4
3
4 Most frequent pattern of length 4:
5     GTAC
```

**Example #2.** The user enters DNA sequence (line 1) and pattern length (line 2). The program then prints the most frequent pattern (line 4 and onward).

```
1 Enter DNA sequence: ATGCAATGCTAAT
2 Enter pattern length: 3
3
4 Most frequent pattern of length 3:
5     ATG
6     TGC
7     AAT
```
Submitting Your Assignment

Once you have completed your programs, submit each of them (q1.py, q2.py) electronically. You may resubmit your files as many times as you need until the due date. Only the most recent submission is graded.

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

# 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 ...

Submit your files on: gradescope.com