**CSCE 110: Programming I**

**Lab 6**

**General Instructions:**

* The 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 L6q<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 that 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 detail.
* Grading is based on correctness and clarity.
* **Copying work from another source and submitting it as your own is plagiarism and a violation of the code of honor. The minimum penalty for plagiarism is a grade of zero and will be reported to the Aggie honor system office.**

**Lab Questions**

1. Write a Python program stored in a file L6q1.py for a grocery store to keep track of the fruits and prices in the store. The manager 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 store to allow her to add, edit, delete, or search an item.

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

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

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

Note 4: The program should print the up-to-date stock at the end of every option selected from the menu **(The name of the fruit is a string and prices are a list of float numbers).**

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.

**Sample Output #1:**

1. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Main Menu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. 1. Add fruits
3. 2. Edit fruit
4. 3. Delete fruit
5. 4. Search fruits
6. 5. Quit
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
8. Choose from menu: 1
9. Enter fruit follow by prices: Apple, 0.45 0.5 0.67 1.2
10. Current Stock: {'Apple': [0.45, 0.5, 0.67, 1.2]}
11. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Main Menu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
12. 1. Add fruits
13. 2. Edit fruit
14. 3. Delete fruit
15. 4. Search fruits
16. 5. Quit
17. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
18. Choose from menu: 1
19. Enter fruit follow by prices: Orange, 0.2 0.7 0.95
20. Current Stock: {'Apple': [0.45, 0.5, 0.67, 1.2], 'Orange': [0.2, 0.7, 0.95]}
21. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Main Menu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
22. 1. Add fruits
23. 2. Edit fruit
24. 3. Delete fruit
25. 4. Search fruits
26. 5. Quit
27. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
28. Choose from menu: 2
29. Enter the fruit name: Apple
30. Enter the fruit prices 0.3 0.57 0.9
31. Current Stock: {'Apple': [0.3, 0.57, 0.9], 'Orange': [0.2, 0.7, 0.95]}
32. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Main Menu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
33. 1. Add fruits
34. 2. Edit fruit
35. 3. Delete fruit
36. 4. Search fruits
37. 5. Quit
38. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
39. Choose from menu: 3
40. If you want to remove all items enter all otherwise enter the fruit name: Banana
41. Banana is not on the list.
42. Current Stock: {'Apple': [0.3, 0.57, 0.9], 'Orange': [0.2, 0.7, 0.95]}
43. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Main Menu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
44. 1. Add fruits
45. 2. Edit fruit
46. 3. Delete fruit
47. 4. Search fruits
48. 5. Quit
49. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
50. Choose from menu: 3
51. If you want to remove all items enter all otherwise enter the fruit name: Orange
52. Current Stock: {'Apple': [0.3, 0.57, 0.9]}
53. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Main Menu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
54. 1. Add fruits
55. 2. Edit fruit
56. 3. Delete fruit
57. 4. Search fruits
58. 5. Quit
59. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
60. Choose from menu: 4
61. Enter the fruit name: Orange
62. Orange is not on the list.
63. Current Stock: {'Apple': [0.3, 0.57, 0.9]}
64. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Main Menu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
65. 1. Add fruits
66. 2. Edit fruit
67. 3. Delete fruit
68. 4. Search fruits
69. 5. Quit
70. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
71. Choose from menu: 4
72. Enter the fruit name: Apple
73. Apple have 3 different prices of: 0.3 0.57 0.9
74. Current Stock: {'Apple': [0.3, 0.57, 0.9]}
75. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Main Menu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
76. 1. Add fruits
77. 2. Edit fruit
78. 3. Delete fruit
79. 4. Search fruits
80. 5. Quit
81. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
82. Choose from menu: 5
83. Current Stock: {'Apple': [0.3, 0.57, 0.9]}
84. A pattern in a DNA sequence is defined using A, T, G, and, C repeatedly. Scientists often need to find the most recurring pattern of a specific length in the DNA sequence. Write a Python program stored in a file L6q2.py that takes a DNA sequence and the pattern length as inputs. The program should then print the most frequent pattern(s) of the requested length. Use a Python dictionary to store the patterns and their count.

**Sample output #1**

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

**Sample Output #2:**

1. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. Enter DNA sequence: ATGCAATGCTAAT
3. Enter pattern length: 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 (L6q1.py, L6q2.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/2021 # Section: Student section number  # E-mail: student\_email@tamu.edu  # Description: # e.g. This program asks for ...** |

Submit your files on [gradescope.com](https://www.gradescope.com/)