**CSCE 110: Programming I**

**Lab 4**

**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 L4q<num>.py, where num is the question number. For example, your solution to the first question will be stored in the file L4q1.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 L4q1.py that :
2. Gets a number as an input from the user and stores it in a variable***.*** That number represents the length of a list. The program should then keep prompting until the user enters a valid number which is less than 10, and greater than 2.
3. Creates a list where the user enters each entry in the list (explicitly convert each entry to type int). An entry is considered valid only if it is strictly greater than 0. If the entry is not valid, then the program should re-run from the invalid entry index, until valid entries are entered by the user.
4. Gets a target number as an input from the user***,*** and prints any two distinct list entries (values and indices) whose sum is equal to the target number. If no such entries are found then it should print a message stating that fact.

**Sample Output #1:**

1. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Section A\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
2. **Enter the length: 2**
3. **Invalid list length, try again.**
4. **Enter the length: 0**
5. **Invalid list length, try again.**
6. **Enter the length: 5**
7. **The list contains 5 entries.**
8. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Section B\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
9. **Enter the entry at index 0: 4**
10. **Enter the entry at index 1: 5**
11. **Enter the entry at index 2: -1**
12. **Invalid entry at index 2.**
13. **Enter the entry at index 2: 6**
14. **Enter the entry at index 3: -8**
15. **Invalid entry at index 3.**
16. **Enter the entry at index 3: 5**
17. **Enter the entry at index 4: 3**
18. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Section C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
19. **Enter the target value: 10**
20. **The entries 4, 6 found at index 0, 2 sum to 10.**
21. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Sample Output #2:**

1. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Section A\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
2. **Enter the length: 3**
3. **The list contains 3 entries.**
4. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Section B\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
5. **Enter the entry at index 0: 2**
6. **Enter the entry at index 1: 4**
7. **Enter the entry at index 2: 5**
8. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Section C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
9. **Enter the target value: 8**
10. **No valid combination summing to 8 was found.**
11. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

1. *Fibonacci Sequence.* Fibonacci numbers are numbers in a sequence, where each number is the sum of the two previous numbers. Starting with 0 and 1, Fibonacci sequence goes like 1, 1, 2, 3, 5, 8, 13 etc. Write a Python program stored in a file L4q2.py that asks for a starting number and an ending number, and then prints all the Fibonacci numbers between them (and including the starting and ending numbers, if they are also Fibonacci numbers). Each Fibonacci number, must be separated by a space.  
     
   Example #1. The user enters start and end numbers (lines 2–3). The program then outputs the total number of Fibonacci numbers within range, along with the numbers.

**Sample output**

1. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
2. **Enter starting number: 5**
3. **Enter ending number: 500**
4. **The 10 Fibonacci numbers between 5 and 500 are:**
5. **5 8 13 21 34 55 89 144 233 377**
6. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Example #2. The user enters start and end numbers (lines 1–2). The program then outputs the total number of Fibonacci numbers within range, along with the numbers.

**Sample output**

1. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
2. **Enter starting number: 55**
3. **Enter ending number: 999**
4. **The 7 Fibonacci numbers between 55 and 999 are:**
5. **55 89 144 233 377 610 987**
6. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

1. Write a Python program stored in a file L4q3.py that:

* Gets single-digit numbers from the user on one line (digits are separated by white space) and adds them to a list. The first digit and the last digit should not be a zero. If the user provides an invalid entry, the program should prompt for a new value.
* Converts every entry in the list to an integer and prints the list. The digits in the list represent a non-negative integer. For example, if the list is [4,5,9,1], the entries represent the integer 4591. The most significant digit is at the zeroth index of the list.
* Adds the integer that the reversed value of the integer. For example, if the list is [4,5,9,1], the addition will be 4591 + 1954 = 6545 and the resulting list will be [6,5,4,5]

**Sample Output#1:**

1. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
2. **Enter single digits separated by one space: 0 5 1 1**
3. **Invalid list, try again.**
4. **Enter single digits separated by one space: 8 5 9 0**
5. **Invalid list, try again.**
6. **Enter single digits separated by one space: 0 58 1 14**
7. **Invalid list, try again.**
8. **Enter single digits separated by one space: 4 5 9 1**
9. **The initial list is: [4,5,9,1]**
10. **The reversed list is: [1,9,5,4]**
11. **The addition list is: [6,5,4,5]**
12. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Sample Output#2:**

1. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
2. **Enter single digits separated by one space : 9 9 9 9 2**
3. **The initial list is: [9,9,9,9,2]**
4. **The reversed list is: [2,9,9,9,9]**
5. **The addition list is: [1,2,9,9,9,1]**
6. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
7. Write a Python program stored in a file L4q4.py that:that takes a string variable as input from the user and prints the length of the entry on the next line. If the length of the entry is even, print all the characters that are in the even position. If the length of the entry is odd, print all the characters that are in the odd position.

Add an empty line.

On the following line, print the index of each character under the corresponding character.

**Sample output #1:** Print one full line of \* on line 1. At the prompt, the user enters “Texas A&M University” (line 2). The program outputs the length of input on line 3 (even). Line 4 is blank and on line 5 all the characters in the even positions are printed. The corresponding index of each character is printed below them on line 6 and one full line of \* is printed on line 7.

|  |
| --- |
| 1. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** 2. **Enter a string variable: Texas A&M University** 3. **Length of the string Texas A&M University is 20** 4. **T    x    s    A    M    U    i    e    s    t** 5. **0    2    4    6    8    10   12   14   16   18** 6. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** |

**Sample output #2:** Print one full line of \* on line 1. At the prompt, the user enters “Howdy! Good Luck.” (line 2). The program outputs the length of input on line 3 (odd). Line 4 is blank and on line 5 all the characters in even positions are printed. The corresponding index of each character is printed below them on line 6 and one full line of \* is printed on line 7.

|  |
| --- |
| 1. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** 2. **Enter a string variable: Howdy! Good Luck.** 3. **Length of the string Howdy! Good Luck. is 17** 4. **o    d    !    G    o          u     k** 5. **1    3    5    7    9    11    13    15** 6. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** |

**Submitting Your Assignment**

Once you have completed your programs, submit each of them (L4q1.py, L4q2.py, L4q3.py, L4q4.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/)