2022 Spring -- CSCI 1300L
Lab 07: List operations: Fibonacci sequences

Introduction

In this lab, you will use the list structure in Python. Lists are used to store multiple values without the need to create multiple independent variables. This lab covers the construction of the list, list operations and/or list methods.

Lab objectives

After completing this lab, you will be able to:
(1) create list expressions by using index values,
(2) use loop statements to process a list, and
(3) use list operations and apply appropriate list methods for a task.

Assignment

In this assignment, you will write a Python program to generate a list of Fibonacci numbers: Fibonacci sequence: 1, 1, 2, 3, 5, 8, … For the detailed information about Fibonacci numbers, see: https://en.wikipedia.org/wiki/Fibonacci_number . Moreover, you will make the program to accept user inputs to generate any sequence of numbers with the Fibonacci calculation rule.

This sequence starts with two numbers: 1 and 1. Each subsequent number is the sum of the two previous numbers. The list that you create in the program should treat the first number 1 as index #0, the second number 1 as index #1, the next number 2 as index #2, etc. The list should be generated by the computer program with the first two input values. Take an example as follows:

```
Enter the first number:
1
Enter the second number:
1
Enter how many numbers in this sequence:
10
The Fibonacci sequence is:
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
```

The red texts are user inputs. After the program accepts all three inputs, it will print the list of all elements that shows the Fibonacci sequence.

Sometimes, the user may input the first two values other than 1, 1. The generated sequence should also follow the same Fibonacci calculation rule (i.e. the sum of any two previous values is the value for the third number). Take another example as follows:

```
Enter the first number:
3
Enter the second number:
1
Enter how many numbers in this sequence:
5
The Fibonacci sequence is:
[3, 1, 4, 5, 9]
```

There are many different ways to generate a Fibonacci sequence in programming. One way is to use recursion (if you have heard of this term). This is not required for this programming practice, but you may try if you can independently write the code. Our suggested way to complete the task of sequence generation is to use loops. At each round of a loop, the program calculates the sum of then-previous two values, and it adds the newly calculated number into the existing list. Therefore, the list will grow as long as the loop continues.

For simplicity, your program does not need to handle non-integer inputs. Furthermore, there is no requirement for your program to process a very long sequence, e.g. with 1000+ numbers. Other taught programming practices including the use of functions may be applied, but it is still okay not to implement them. There is no strict requirement regarding the specific format of input or output. However, we must make sure that:

(1) the calculation and the generated sequence are correct for any valid user inputs, and

(2) the program can process the third user input successfully, EITHER to accept the value (if greater than or equal to 2) and output the sequence, OR to reject that value by printing an error message such as “the number cannot be smaller than 2” and terminate the program.

(3) The assignment is completed with the creation and use of a list. If no list structure is used in this assignment, no credit will be given.

Submission instruction

After you have completed the assignment, upload and submit the Python source code file *Lab07.py* to eLC. Always double check that your submission was successful on eLC.

Grading

A score between 0 and 5 will be assigned.

1. The program can successfully process an end user’s inputs (all three). (1 point)
2. The program can handle an invalid integer for the number of elements in the sequence, e.g. user inputs 0. (1 point)

3. The program can generate the original Fibonacci sequence correctly (i.e. first two numbers are 1 and 1, respectively). (1 point)
4. The program can generate any “customized Fibonacci sequence” correctly. For example, first two numbers are 0 and -7, respectively). (1 point)
5. Only a single source code file is submitted and no other file is submitted (0.5) and the entire Python program can be executed without any additional error (0.5). (1 point)

Special notice regarding the submission:

Late submission penalty. Points will be deducted from the original grade. If your submission is after the posted deadline…

(1) within 24 hours: -2
(2) between 24 hours and 48 hours: -3
(3) between 48 hours and 72 hours: -4
(4) after 72 hours: assignment will not be accepted.