CS215: Introduction to Program Design, Abstraction and Problem Solving
(Fall, 2022)
Lab Assignment 4
(20 points)

Today's Date: Tuesday, September 13

Demonstration Due Date: the end of Lab5 class
Submission Due Date: Friday, September 23

The purpose of this lab assignment is

- to continue practicing conditional statements
- to practice loop statements
- to validate the user input

Problem Statement

Write a program that repeatedly reads a sequence of integers from the user as inputs, until the user clicks "Q" or "q" to quit the problem, and then prints

- The corresponding item in the cumulative total sequence. For example, if the user input sequence is **`1 7 9 -4 5`**, its cumulative total sequence is **`1 8 17 13 18`** . When the user first types **`1`**, its corresponding item in the cumulative total sequence is **`1`**; then the user types **`7`**, its corresponding item is **`8`** (it is equal to **`1+7`**); the user then types **`9`**, its corresponding item is **`17`** (it is equal to **`1+7+9`**); and so on
- The smallest and largest among the sequence of user inputs
- Counting how many prime numbers among the sequence of user inputs (A **prime number** (or a **prime**) is a natural number greater than 1 that is not a product of two smaller natural numbers.)

Your program cannot assume the user will always input the valid data, which the program is expecting, instead your program should be able to validate the user input as we discussed in the lecture. For example, the user may type a floating-point number such as **`17.89`** when your program is waiting for an integer. In this case, the input stream will extract **`17`** and store it into your integer variable as the valid input (Note that this will not make the `cin.fail()` function into true), but the rest part **`.89`** will stay in the input stream and block the future input, hence you need to extract the rest part from the input stream so that you can get more user input. The user may type characters such as **`four`** when your program is waiting for an integer. In this case, your program should allow the user to try to input again till a valid data has been input or the user types letter Q or q to quit the program. Think about designing your own testing cases for your program.

The following are some examples of running your program:

Sample output 1:

```
Please enter a series of numbers, then type (Q or q) to
process: Q↵
No input number!
```

```
Have a great day!
```

Sample output 2:

```
Please enter a series of numbers, then type (Q or q) to
process: 2↵
The corresponding element for the cumulative total sequence
is : 2
Please enter a series of numbers, then type (Q or q) to
process: q↵

Largest: 2
Smallest: 2
How many Prime numbers? 1
Have a great day!
```

Sample output 3:

```
Please enter a series of numbers, then type (Q or q) to
process: eight seven six↵
Invalid input, please try again...
Please enter a series of numbers, then type (Q or q) to
process: five↵
Invalid input, please try again...
Please enter a series of numbers, then type (Q or q) to
process: -5 is ok? or not? ↵
The corresponding element for the cumulative total sequence
is : -5
Please enter a series of numbers, then type (Q or q) to
process: 88.3↵
The corresponding element for the cumulative total sequence
is : 83
Please enter a series of numbers, then type (Q or q) to
process: 25↵
The corresponding element for the cumulative total sequence
is : 108
Please enter a series of numbers, then type (Q or q) to
process: 18.75 ok? ↵
The corresponding element for the cumulative total sequence
is : 126
Please enter a series of numbers, then type (Q or q) to
process: Q↵

Largest: 88
Smallest: -5
How many Prime numbers? 0
Have a great day!
```

Sample output 4:

```
Please enter a series of numbers, then type (Q or q) to
process: you mean numbers, right?↲
Invalid input, please try again...
Please enter a series of numbers, then type (Q or q) to
process: eight↲
Invalid input, please try again...
Please enter a series of numbers, then type (Q or q) to
process: 8↲
The corresponding element for the cumulative total sequence
is : 8
Please enter a series of numbers, then type (Q or q) to
process: 7seven↲
The corresponding element for the cumulative total sequence
is : 15
Please enter a series of numbers, then type (Q or q) to
process: Did I input seven?↲
Invalid input, please try again...
Please enter a series of numbers, then type (Q or q) to
process: -33↲
The corresponding element for the cumulative total sequence
is : -18
Please enter a series of numbers, then type (Q or q) to
process: 13.75↲
The corresponding element for the cumulative total sequence
is : -5
Please enter a series of numbers, then type (Q or q) to
process: is 19 ok?↲
Invalid input, please try again...
Please enter a series of numbers, then type (Q or q) to
process: 19 should be ok!↲
The corresponding element for the cumulative total sequence
is : 14
Please enter a series of numbers, then type (Q or q) to
process: 215 is my lucky number:)↲
The corresponding element for the cumulative total sequence
is : 229
Please enter a series of numbers, then type (Q or q) to
process: q↲

Largest: 215
Smallest: -33
How many Prime numbers? 3
Have a great day!
```

Note that the blue part represents the user input, and "↲" represents the return key from the user input.

Demonstration and Submission

1. Each Lab assignment needs to demonstrate to your TA to be graded. You can demonstrate Lab4 during Lab4 class (with possible bonus 3 points) or no later than the end of Lab5 class (this is **the demonstration deadline** for Lab4).
If you finish Lab4 assignment during Lab4 class, you may demonstrate your program to your TA and answer your TA's questions, you can get up to 3 extra points for this lab assignment. (Note you can also demonstrate your program to your TA during Lab5 class. However, any demonstration later than the end of the Lab4 class cannot get bonus 3 points.)
If you need extra time, you can continue working on Lab4 assignment after the Lab class, and try to finish it before the next Lab class. Then demonstrate your Lab4 during Lab5 class.
If you do not demonstrate your code, even if you submit it in Canvas, you will receive a grade of 0!! *The TA may ask you to make some corrections. If so, make the corrections and demonstrate again…repeat until you have 100%!*
2. After the successful demonstration, submit the code in Canvas. Open the link to Course Canvas page (https://www.uky.edu/canvas), and log in to your account using your LinkBlue ID and password. Please submit your **source code in a .cpp file** through link "**Lab 4**".
Even if you successfully demonstrated it to the TA, if you do not submit in Canvas by **the submission deadline*****, you will receive a grade of 0!***

Grading (20 points + Bonus 3 points)

1. Attend the lab session or have a documented excused absence. (5 points)
2. Demonstrate your program to your TA and submit it in Canvas. (15 points)
   - Include comments as specified in the lecture notes. (2 points)
   - Handle input validation for different cases (3 points)
   - Generate the correct largest and smallest among input sequence. (3 points)
   - Generate the correct count of prime numbers among input sequence. (3 points)
   - Generate the correct corresponding items for cumulative total sequence (4 points)

Demonstrate your program to your TA and answer TA's questions during Lab class when the same Lab assignment is given. (Bonus 3 points)