

# Processing and displaying user inputs part 3

Object Oriented Programming - Programming Coursework #3 (11 pts)

Due Date : May 28, 2020 5pm

## 1 Task

The objective of this coursework is to gain a hand-on experience on making decisions and looping. The students have to implement two programs. The first program (Program 1) takes from the keyboard, two (integer) numbers, say  $a$  and  $b$  and prints the sum of all integer numbers that are larger than or equal to  $a$  and smaller than or equal to  $b$  (3 pts). Note this program is the same as Program 1 of CW2 except that now,  $a$  and  $b$  can take any integer values. The second program (Program 2) takes a positive integer, say  $c$  in  $\{1, \dots, 100000\}$  and checks whether  $c$  is a prime number or not (3 pts): Program 2 will print out 1 when  $c$  is a prime number and 0 otherwise. The submitted codes have to satisfy the compilation and file name requirements (Sec. 3): If the codes do not meet these requirements, the students will receive 0 point. We will test Program 1 and Program 2 with five different user (keyboard) input combinations. If the submitted program generates correct answers for all five input instances, the students will receive full marks.

## 2 Program specifications

For both Program 1 and Program 2, the input will be provided by the users using the keyboard. After taking the respective inputs, Program 1 will print out (i.e., display in the monitor) an integer number, while Program 2 will print 1 (when the input is a prime number) or 0 (When the input is not a prime number). For program 1, if  $a$  is larger than  $b$ , the output should be zero. For program 2, please note that 1 is not a prime number while 2 is a prime number.

### 2.1 Sample inputs and outputs

Program 1 input: The user will type two numbers separated by a space and then type 'Return' ('Enter').

```
2 500
```

Program 1 output to be visualized in the monitor:

```
125249
```

Note that  $125249 = 2 + 3 + \dots + 499 + 500$ .

Program 1 input: The user will type two numbers separated by a space and then type 'Return' ('Enter').

```
-2 5
```

Program 1 output to be visualized in the monitor:

```
12
```

Note that  $12 = -2 - 1 + 0 + 1 + 2 + 3 + 4 + 5$ .

Program 1 input: The user will type two integers separated by a space and then type 'Return' ('Enter').

```
7 3
```

Program 1 output to be visualized in the monitor:

```
0
```

Note the output is zero since the first number (7) is larger than the second one (3).

Program 1 input: The user will type two numbers separated by a space and then type ‘Return’ (‘Enter’).

```
5 5
```

Program 1 output to be visualized in the monitor:

```
5
```

Program 2 input: The user will type an integer in  $\{1, \dots, 100000\}$  then type ‘Return’ (‘Enter’).

```
92567
```

Program 2 output to be visualized in the monitor:

```
1
```

Note 92567 is a prime number.

Program 2 input: The user will type an integer in  $\{1, \dots, 100000\}$  then type ‘Return’ (‘Enter’).

```
1
```

Program 2 output to be visualized in the monitor:

```
0
```

Program 2 input: The user will type an integer in  $\{1, \dots, 100000\}$  then type ‘Return’ (‘Enter’).

```
65536
```

Program 2 output to be visualized in the monitor:

```
0
```

For this CW, your programs do not have to handle exceptions caused by incorrect input data types. For instance, for Program 2 the user wouldn’t provide any non-integer or negative number, or any number larger than 100000:

```
0.009
```

### 3 What to hand in for assessment

Please submit the source codes for Program 1 and Program 2, and a brief report (a single report for all programs) via Black Board. Comments in the codes and the report should be written in English. The report should provide code comments explaining the algorithms. If the submission does not include a report, the maximum possible points will be kept at 40% of the full points. Please format the submission as a single zip file ‘StudentID\_Name.zip’ that includes the files of the source codes and the report,

e.g.,

20201111\_KwangInKim.zip

- Program1.cpp

- Program2.cpp

- Report.pdf

**The submitted source codes should contain ‘Program1.cpp’ and ‘Program2.cpp’ files each including the id and name of the submitting student.**

In “Program1.cpp”

```
//StudentId YourName
#include <iostream>
Your code here...
```

### 3.1 Compilation requirements and instruction

Each submitted code needs to be compilable by the GNU C++ compiler and in the Linux environment. Specifically, they should be **compilable and executable in UNI06 server**. Students can verify the correct compilation and execution of their codes by following the instructions provided in ‘howtocompile.pdf’ document accompanying the CW1 announcement.