University of Wyoming
COSC 1030
Computer Science I

Program 03
Looping structures

1 Purpose

This program expands on the discussions of looping structures. You will create three (3) different loops in the program and test each one. Before you start writing C++ code, write pseudocode, a plan for the programming.

Before you ask, there is no particular reason for what each loop type is doing. The point is just to make you familiar with the different types and maybe give you an idea of when you should use which type (and when you should not) and maybe even if you need a loop at all.

2 Procedure

1. Your task is to write a C++ program which makes three different computations, each controlled by some loop. You will prompt the user for loop intervals something like you did for the last program.

   (a) Your program will first prompt the user for **one** integer. It will then compute the sum of the values between 1 and the user's input value. That is it will compute

   $$\sum_{i=1}^{n} i$$

   - This phase of the program **shall** use a <u>for</u> loop for the computation.
   - If the user's input is less than 1, you will continue to prompt the user for a value greater than or equal to one, until the user complies.
   - When the computation is done, output the result in a single line like
     `The sum of integers from 1 to 5 is:  15`

   (b) Next, you will prompt the user for a **pair** of integers and output the length of the interval *between* the numbers. Example: if the input is "5 5" the length is 0, if the input is "5 6" the length is 0, if the input is "1 9" the length is 7. Notice that the end points are NOT counted in the length.

   - This time you **shall** use a <u>while</u> loop to determine the interval.

- There are **no** incorrect values here, any two integers will do.
- Do not forget to test when the smaller value is negative (-1 and 14), the "larger" value is negative (1 and -14 or -14 and 1), and when both values are negative. This is to see if you have to change around the values so that your loop actually ends.
- When the computation is done, output the result on another line like
  `The interval between 1 and 5 is:  3`
- Yes, I know there are easier ways to do this than using a while loop, but this is for practice with loops.

(c) Last you will compute the product n^i. Prompt the user for **two** more integers and make the computation. YOU WILL NOT USE ANY SPECIAL FUNCTIONS TO MAKE THIS COMPUTATION!!!!!

- Now you **shall** use a do-while loop.
- The first value input, n can be any **integer**. The second value, i, **must** be a non-negative integer because we use it for the exponent. If the i input is negative, you will continue to prompt the user for a new **pair** of values, n and i, where the i is non-negative. Note: If the input values are large, the result will "overflow" and what is printed out will really be undefined. That is the way things happen sometimes, do not worry about it. Certainly do NOT try to detect this issue in the program.
- When the computation is done, output the product on a third line like
  `The product of 3 ∧ 4 is:  81`

2. Upload your solution to the WyoCourses site, Program 03 assignment.

3. Your solution should consist of precisely three files (NOTE: no .exe file!!!!).

   Please use the comment heading format as defined in Lab01 for all your solution files.

   (a) A single text file, **Prog03Pcode.txt** which contains the psuedocode that you developed as a plan for this program.

   (b) A single C++ source code file, **Prog03.cpp**

   (c) A single text file, **Prog03Test.txt** which contains the demonstration dialog of your programs behavior.

   Insert comments in this file regarding the validity of your results (for example, demonstrate that the user inputs are handled as prescribed above).

   An example of the testing for this programming assignment might be as follows:

```
// Prog03Test.txt
// Kim Buckner
// COSC 1030
// Program 03
// Date

// First run with all values in range

buckner ~...prog03/solution> Prog03

Please input an integer greater than or equal to 1: 9
The sum of integers from 1 to 9 is:45

Please input two integers: 1 9
The interval between 1 and 9 is: 7

Please input two integers, the second must be non-negative: 2 10
The product of 2 ^ 10 is: 1024

// Second run with values out of range

buckner ~...prog03/solution> Prog03

Please input an integer greater than or equal to 1: -34
Please input an integer greater than or equal to 1: -98
Please input an integer greater than or equal to 1: 0
Please input an integer greater than or equal to 1: 23
The sum of integers from 1 to 23 is: 276

Please input two integers: 9 -9
The interval between -9 and 9 is: 17

Please input two integers, the second must be non-negative: 2 -8
Please input two integers, the second must be non-negative: 2 0
The product of 2 ^ 0 is: 1

buckner ~...prog03/solution>
```