COMP ENG 2SH4

DURATION OF EXAMINATION: 2.5 Hours Instructor: MOHAMED HASSAN
MCMASTER UNIVERSITY Final Exam Dec 2021

SPECIAL INSTRUCTIONS:

- First things first: Update the README file with your information.
- **Closed-Book Exam:** No reference material of any kind is allowed. Only McMaster standard calculator (Casio FX-991 MS or MS Plus) is allowed.
- Exam is three questions, Q1 has three parts (a)–(c), Q2 has two parts (a) and (b), while Q3 has four parts (a)–(d).
- Answer questions in their corresponding source files as instructed in each question.
- Make sure to commit/push at least after each part of the exam and/or every 10 minutes to avoid any problems.
- Students approved by SAS for extra-time, please follow the guidelines given to you in the SAS letter.

1. (20 Marks) **Question1.**
   This question with its parts should be answered in file: Q1.c.
   In this question we are interested in calculating 1+2+...+n, where
   Pre-condition: n is 1 or greater
   Post-condition: this functions returns the value of 1+2+3+...+n
   Examples: sum(1) is 1, sum(2) is 3, sum(3) is 6, ...

   (a) [7 Marks] Complete the function sum_while in Q1.c using a single while-loop.

   (b) [7 Marks] complete the function sum_for implementation in Q1.c using a single for-loop.

   (c) [6 Marks] complete the function sum_do_while implementation in Q1.c using a single do-while-loop.

   Note that you are given an incomplete function signature; you need to complete the signature as well as the function body to deploy the intended functionality as explained above.

2. (30 Marks) **Question2.**
This question with its parts should be answered in file: Q2.cpp.

In this question, we are looking to identify substrings of Size Three with Distinct Characters. A string is good if there are no repeated characters. Given a string s, return the number of good substrings of length three in s. Note that if there are multiple occurrences of the same substring, every occurrence should be counted. A substring is a contiguous sequence of characters in a string.
Example 1:
Input: s = "xyzzaz"
Output: 1
Explanation: There are 4 substrings of size 3: "xyz", "yzz", "zza", and "zaz".
The only good substring of length 3 is "xyz".

Example 2:
Input: s = "aababcabc"
Output: 4
Explanation: There are 7 substrings of size 3: "aab", "aba", "bab", "abc", "bca", "cab", and "abc".
The good substrings are "abc", "bca", "cab", and "abc".

(a) [25 marks] Complete the function countGoodSubstrings in Q2.cpp and correct the signature as well to perform this task.

(b) [5 marks] Complete the main() function in Q2.cpp to test the countGoodSubstrings function.

3. (50 Marks) **Question 3.**
This question with its parts should be answered in files: Q3.c, ScoreBank.h, ScoreBank.cpp, Curve1.h, Curve1.cpp, Curve2.h, and Curve2.cpp.

In this question, you are going to help your instructors perform grade curving. Basically, you are asked to write a program that reads in test scores (part (a)) and applies two different curves to them (parts (b) and (c)). For all parts, feel free to add any other methods or data members to the classes other than the ones mentioned if it will help you implement any of the required features. The program should contain the following:

(Q3-a)[14 marks] a base class ScoreBank with two private(or protected as needed) data members:

- an integer pointer for the scores list
- a float for the average.

The class should also contain the following methods:

- method EnterScores which:
  - asks the user how many test scores are needed,
  - allocates enough memory, and reads in the scores.
- a method CalcAverage which:
  - stores the average of the entered scores in the float data member.
- a method printScores which:
  - prints the list of test scores to the screen as well as the average.
- Make sure also to complete the given constructor and destructor. In the given constructor, you should free the memory allocated in the class.

All this part must be answered in files ScoreBank.h and ScoreBank.cpp

(Q3-b)[14 marks] In this part, we will explore the first curving method. Derive from ScoreBank a class Curve1 which contains:

- a method Curve which:
  - sets the average score to 75 if it is less than 75. Find out how far away from 75 the actual average is and then add this value to each test score.
- Overload the printScores method to print:
  - the original scores and
  - the curved scores as well as
  - the original and new average.
- Make sure also to complete the given constructor and destructor. In the given constructor, you should free the memory allocated in the class.

All this part must be answered in files Curve1.h and Curve1.cpp

(Q3-c)[14 marks] In this part, we will explore the second curving method. Derive from ScoreBank a class Curve2 which contains:

- a method Curve which:
  - sets the high score to 100 and scales the rest of the scores accordingly. The scale should be 100 divided by the highest score; then, multiply each score in the list with the scale.
- Overload the printScores method to print:
  - the original scores and
  - the curved scores as well as
  - the original and new average.

All this part must be answered in files Curve2.h and Curve2.cpp

(Q3-d)[8 marks] Write a main function to test the functionality of these classes. You should at least: create objects of type Curve1 and Curve2 such that the user inputs a list of test scores into each object and then you print the information of the lists using the printScores functions from each object. This should be written inside the main() function in file Q3.cpp

THE END