|  |  |
| --- | --- |
| Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Student ID#: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| **SODV 1101**  **Programming Fundamentals**  Winter 2024  Assignment 2 | |
| **Instructor: Mahbub Murshed** | Date: 2023-03-21 |
| **Directions:**   * Written questions can be submitted via soft copy or a scanned document and submitted by d2l dropbox, **as a single doc file or single pdf file**. Answers should be written clearly and coherently. **Programming questions must be submitted in repl.it.** * Assignments must be submitted by the posted due date, or be subject to the course’s late policy. Work must be completed individually and in accordance with the academic integrity policy. * All codes shown in this assignment are compiled and tested except for questions that aim to find errors in a C++ program. * For all written questions, you may assume that the usual header files are included as follows:   **#include <iostream>**  **#include <cmath>**  **#include <string>**  **using namespace std;**  **Note: The assignment’s written questions not submitted as a single file will not be graded!** | |
| |  |  | | --- | --- | | **Written Questions** | | | **1** | / 6 | | **2** | / 14 | | **3** | / 10 | | **Programming Questions** | | | **Quality** | / 20 | | **Functionality** | / 50 | | **Total** | | | / 100 | | | |

**Assignment 2: Written Questions**

|  |  |  |
| --- | --- | --- |
| 1. | Give concise answers to the following questions |  |
| a) | What is an array? | / 3 |
| b) | What do the break and continue keywords do? | / 3 |
| 2. |  |  |
| a) | What is the exact output of the code below if we execute the function call q2(20)?   |  | | --- | | void q2(int n)  {  for (int i = 1; i < n; i \*= 2)  cout << i;  } | | / 4 |
| b) | The function in part (a) uses a for-loop. Rewrite the function using a while-loop without changing how the code works. | / 5 |
| c) | Rewrite the function from part (a) using a do-while loop. (Hint: make sure the code produces the same output for all cases). | / 5 |

|  |  |  |
| --- | --- | --- |
| 1. In the following function implementation, there are 2 syntax errors and 2 logic errors. Mark the errors in the code below with a circled number corresponding to the spaces provided. Briefly explain why it is an error **and** give a fix for it.  |  | | --- | | #include<iostream>  using namespace std;  int isAscending(int[] arr, int size) {  i = 1;  for (; i <= size; i++)  {  if (arr[i - 1] > arr[i]) {  return false;  }  else  return true;  }  } |   1) Syntax Error: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  2) Syntax Error: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  3) Logic Error: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  4) Logic Error: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | / 10 |

**Program 2(A): Array Merging**

**(Please submit in repl.it)**

**Learning Objective**

* Identify program structure and control the flow of execution.
* Write structured programming instructions so that the computer shows input/output behaviors
* Use different types of loops to write code with the required number of iterations.
* Demonstrate logical thinking by using programming syntax and strategies.
* Use arrays to store collections of data.

**Overview**

Write a program to merge two arrays

**Directions**

For example, consider two arrays A and B.

A = {10, 25, 35, 40, 55}

B = {15, 30, 5, 20, 45, 65}

The merged array should be another array C with elements {5, 10, 15, 20, 25, 30, 35, 40, 45, 55, 65}. The resultant merged array should contain all the elements of A and B in sorted order. This doesn’t mean that you can two copy two arrays to the third array and sort the resultant array. Initially, you can sort Array ‘A’ and Array ‘B’ with your own sorting method. Afterward, you will need to merge the contents of the two arrays into a single array by taking individual values from the two arrays one by one and inserting them into the resultant Array ‘C’ in the appropriate position, so that it becomes sorted right after the insert operation.

**Sample Input/output:**

How many elements are there in the first Array? 5

How many elements are there in the second Array? 6

Elements in ‘ A’: 10, 25, 35, 40, 55 (Randomly populated)

Elements in ‘ B’: 15, 30, 5, 20, 45, 65 (Randomly populated)

Sorted Elements in ‘ A’: 10, 25, 35, 40, 55 (Sorted by calling your own created method)

Sorted Elements in ‘ B’: 5, 15, 20, 30, 45, 65 (Sorted by calling your own created method)

Elements in ‘C’: 5, 10, 15, 20, 25, 30, 35, 40, 45, 55, 65 (After Merging A and B; and no sorting methods called)

**Program 2(B): Replace String   
(Please submit in repl.it)**

**Learning Objective**

* Identify program structure and control the flow of execution.
* Write structured programming instruction so that the computer shows input/output behaviors
* Use for loops to write code with a required number of iterations.
* Build programs that can construct or parse strings.
* Create formatted outputs by using strings
* Demonstrate logical thinking by using programming syntax and strategies.

**Overview**

Write a C++ program that takes a single line input string. Your program also collects a search string and a replace string. Finally, your program should search for all occurrences of the given search string in the single-line input and replaces them all with the replace string.

**Directions**

In the single-line input, there can be two or more occurrences of the same search string. The search string can be followed by a space character (' '), comma (','), or full stop character ('.').

Therefore, use a loop, inside the loop at every iteration replace one search string with the replace string. Repeat this process until all the search strings are replaced.

**Sample Input/output 01:**

Single Line Input String:

Alex is a good student. Programming is fun for Alex.

Search String:

Alex

Replace String:

Your Name

Your Name is a good student. Programming is fun for Your Name.

**Evaluation**

This assignment has 2 main components, assigned marks as follows:

|  |  |
| --- | --- |
| Task | Marks |
| Written Questions | 30 |
| Code Quality and Functionality | 70 |
| Total | 100 |

Marks for written questions are indicated beside each question.

The **following rubrics will be followed for assessing the program code:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Excellent (100%)** | **Competent (80%)** | **Satisfactory (50%)** | **Unsatisfactory (0%)** |
| * Code contains no compile-time errors. * Code contains no logical errors. * Code produces the exact correct output and does not crash for valid inputs. * Code is well organized, clear, and easy to read. * Code is consistent throughout and comments are used where needed. | * *Code contains no compile-time errors.* * *Code produces the exact correct output except for a few formatting issues / incorrect data type issues.* * *Code does not crash for valid inputs.* * *Code produces the correct output most of the time. However, for some valid inputs, it shows the incorrect output.* * *Minor logical errors are present.* | * *Code contains no compile-time errors.* * *Code is disorganized or could use some refactoring.* * *Code contains major logical errors and does not match the expected output.* | * *Code contains compile-time errors.* * *Code contains major logical errors.* * *Code is very difficult to read.* * *Code does not execute.* |