Assignment 3 – User-Defined Functions/Recursion/ Arrays

Deadline: Monday July 24 at 23:59 PM EDT
Type: Group Assignment
Weight: 5%

Submission instructions:

- Create a cpp file for each question
- Compress the files using zip or other tools
- Submit the zip file on Moodle
- Please do not submit exe files
- All submissions must be done through Moodle

Marking Scheme:

- Program correctness (90%)
- Program clarity (output format, comments, completeness, readability) (10%)

Q1. (25 marks): Write a C++ program that uses an array to store the grades of N students (N is entered by the user), and outputs the information below. Your program should be structured using functions. Each functionality should be implemented in a separate function.

1- 4 highest grades
2- 4 lowest grades
3- Average grade
4- Median grade
5- Number of A grades (grades equal or more than 90)
6- Number of B grades (grades between 80 (inclusive) and 90)
7- Number of C grades (grades between 65 (inclusive) and 80)
8- Number of D grades (grades between 50 (inclusive) and 65)
9- Number of F grades (grades less than 50)

Q2. (25 marks) A Palindrome is a string that is spelled the same way forward and backward. Examples of palindromes include “radar” and “able was I ere I saw elba”. Write a **recursive function testPalindrome** that returns true if a string is palindrome, and false otherwise. Note that like an array, the square brackets ([]) operator (index) can be used to iterate through the characters in a string.

Q3. (50 marks): We want to create a program that controls the movements of a robot in an array of size Initially, the robot is at position 0. The controller supports the following commands:

- right (n): the robot moves n cells right
- left (n): the object moves n cells left
- reboot: the object comes back to cell 0

Consider the following example:

The new array after executing the command right (5) will look like this:

The new array after executing the command left (3) will look like this:

The program takes commands from a user as shown in the following example:

Controller Menu:

1. Right
2. Left
3. Display
4. Reboot
5. Show Array
6. Exit

Command "Display" display the position of the robot in the array. Command "Show Grid" displays the array. Command "Exit" terminates the program.

- Implement the controller's commands using functions. Add the necessarily checks to ensure that the robot does not go out of bounds. Your program should be structured using functions. For example, each command should be implemented in a separate function. (30 marks)
- We want to add the following commands (20 marks):
    - cancel (n): The program cancels the last n operations and returns the robot to the initial position
    - replay (n): The program replays the last n operations that have been cancelled.