

Assignment 3 – User-Defined Functions/Arrays

Deadline: Friday Oct. 25 by 11:55pm

Type: Group Assignment

Weight: This assignment is worth 4% of your final grade

Submission: Must be on Moodle

Submission instructions:

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

Exercises:

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. Maximum grade
2. Minimum grade
3. Average grade
4. Median grade
5. Number of A grades (grades more than 80)
6. Number of B grades (grades between 70 and 80)
7. Number of C grades (grades between 55 and 70)
8. Number of D grades (grades between 40 and 55)
9. Number of F grades (grades less than 40)

Q2. (25 marks) Write a C++ program that takes a sentence as input (type string) and returns what follows. Structure your program using functions.

1. The number of characters
2. The number of words in the sentence
3. The number of proper words (any word that starts with a capital letter except the first word)
4. The number of words that contain two consecutive letters (e.g., "letter")

Q3. (50 marks) We want to create a program that controls the movements of a robot in an array of size 10. 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:

^									
0	1	2	3	4	5	6	7	8	9

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

					^				
0	1	2	3	4	5	6	7	8	9

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

		^							
0	1	2	3	4	5	6	7	8	9

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 necessary 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.
- **(Bonus)** Add a second robot to the grid. Repeat the above questions. The two robot should never collide, meaning, they should never end up on the same cell (25 marks).