TU856/1 & TU858/1 Programming Assignment #2

Due Date: Sunday, March 20th, 2022 (11.59pm)

Requirements:

Develop a program that will play the Lotto game. The program should allow a user to enter their 6 selected numbers and give them a set of options, each performing a specific requirement. You must store the 6 numbers in a 1-Dimensional array.

There are a number of requirements that your program must meet. Your program must be modularised (i.e., use functions) and each task should be dealt in a separate function. The program should display a simple menu to the user and each option in the menu will be implemented by calling a separate function. You must use pointer notation to access array elements – DO NOT use subscript notation.

The requirements are as follows (each implemented in a separate function):

1. Enter any 6 numbers (1–42 inclusive) from the keyboard. Perform any necessary validation (error-checking) required (e.g., all numbers must be different, range 1–42, etc.).
2. Display the contents of the 1-D array containing your lotto numbers that you entered.
3. Sort the contents of the array in increasing order (i.e., 1st element = smallest number, 2nd element = 2nd smallest number, etc.). You may use any sorting algorithm of your choice.
4. Compare your chosen lotto numbers in the 1-D array with the following winning numbers:

   1,3,5,7,9,11 (Winning numbers)

Depending on how many of your chosen numbers match the above winning numbers, your program should display one of the following messages:

Match 6	Jackpot
Match 5	New car
Match 4	Weekend away
Match 3	Cinema pass

5. Display the frequency of the numbers entered each time the user enters a new set of numbers (without exiting the program) on the screen. For example, it might display:

   number 1 has been selected x times
   number 7 has been selected x times
   number 28 has been selected x times
   etc.,

6. Exit program

Extra:

- After a function has completed, your program should return to the main menu and allow the user to select another option.
- The user should only be allowed to select options 2, 3, 4 & 5 only if they have correctly entered their 6 valid chosen numbers, i.e., option 1. Display appropriate error messages to handle any errors.

Submission details:

1. Submit your program (.c source code file only) using the assignment listed in the Programming module in Brightspace. This must be submitted on or before Sunday, March 20th, 2022 (11.59pm).
2. Marks will be awarded for well-written code (comments, indentation, whitespace, good use of brackets, etc.,).

Note: You are required to demo your program in the lab session within 3 weeks following submission. Failure to demo your program will result in a zero mark being recorded.

Late submissions: Late submissions will incur a progressive penalty of 10% each day late, i.e., 1 day late loses 10%, 2 days late loses 20%, etc.,

No submissions accepted after 1 week and a zero-mark awarded.

NB - This is an individual assignment and **NOT** a group one. Do your own work and do not plagiarise your code. Anti-plagiarism software will be used to randomly check submissions. Any submitted code suspected of having been plagiarised will be brought to the attention of the module examiners for specialised checks under the TU Dublin general assessment regulations.

See marking scheme (rubric) below.

Marking scheme (Rubric):

Table 1 shows the marks allocated for this assignment.

Functionality (Requirements). Code should meet the highest professional standards	Menu option 1 (Enter 6 numbers)	5%	
	Menu option 2 (Display 6 numbers entered)	5%	
	Menu option 3 (Sort the array in increasing order)	15%	
	Menu option 4 (Compare numbers entered with winning numbers)	15%	
	Menu option 5 (Display frequency of numbers entered in current program run)	20%	
	Menu option 6 (Program ends gracefully)	5%	
Error checking (Validation) Validate all program input, etc.,		15%	
		Sub Total:	**80%**
Commenting	Program Description, Author, Date	5%	
	Good comments throughout code body	10%	
Indentation	Correct and consistent indentation throughout code body	5%	
		Sub Total:	**20%**

Table 1: Marking scheme (Rubric)