CSE108 – Computer Programming Lab.

Lab 10

Dynamic memory allocation

Due at 10:00

Hand in: A student with number 20180000001 should hand in a zip file named 20180000001.zip for this lab.

Part 1. (25 pts) Write a C **function** that generates and returns an array of integers. The size of the array is obtained from the user **in the main function** and passed to the function as a parameter. The function dynamically allocates memory for the array based on the given size. Next, the function prompts the user to input integer values for each element of the array. **In the main function**, the minimum value among all the elements in the array is printed.

Example:

Input:

Enter the size of the array: 5

Enter the elements of the array: 8 4 2 8 10

Output:

Min of the array elements: 2

Part 2. (25 pts)

Write a C program that utilizes the function from part 1 to generate an array of integers. As before, the size of the array should be obtained from the user in the main function. Afterward, the program should generate and allocate memory for a second array of the same size using **calloc** within the **main function**. The second array should store the cumulative sum of the elements from the first array. Finally, in the main function, print the elements of both arrays.

Example:

Input:

Enter the number of integers: 5

Enter 5 integers: 1 2 3 4 5

Output:

First array: 1 2 3 4 5

Second array (cumulative sum): 1 3 6 10 15

Part 3. (50 pts) Define a struct called "Student" with the following members: "ID" (integer), "age" (integer), and "gpa" (float). Write a C function that dynamically allocates memory for an array of Student structures. The size of the array should be 10.000, and the function should allocate memory accordingly. The content of the items (ID, age, and GPA) in the array should be filled with randomly generated values for each student. Finally, the function calculates and returns the average GPA.

Call this function in the main function 10,000 times and print the average of the average GPAs.

Example:

Input:

No input.

Output:

Average GPA of 10.000 x 10.000 students: 2.2

Note: Don't forget to deallocate the dynamically allocated memory at the end of the function to avoid memory leaks.

General Rules:

1. You will have two hours to provide a solution to the given problem set. You are not permitted to ask any questions. If there is a significant error in the assigned tasks, it will be addressed later.
2. You will be able to hand in your solutions via Teams in the next two hours. The submission will be closed exactly at 10am.
3. There will be an interview session immediately after the submission deadline. Starting at 10am, you will be randomly invited to attend a meeting by a TA to demonstrate your solution and answer any questions asked by the TA.
4. You must be available until 1pm to respond to the demo invitation whenever you receive it. You will have 3 minutes after you are called via Teams. If you do not answer/appear in 3 minutes, you will miss you interview.
5. If you miss your interview or are unable to give satisfactory answers to the questions, you will receive a zero for that lab even if you have submitted your solution.
6. If you have not submitted a solution in time, you will not be invited for the interview and receive zero for that lab.
7. Due to time constraints, some students may not be invited to an interview. In that case, their solutions will be graded offline.
8. Unless you aren’t declared for a specific prototype, you may use arbitrary but proper function and variable names that evoke its functionality.
9. The solution must be developed on given version of OS and must be compiled with GCC compiler, any problem which rises due to using another OS or compiler won’t be tolerated.
10. Note that if any part of your program is not working as expected, then you can get zero from the related part, even it is working partially.
11. Zip your solution file before uploading it to MS Teams. The zip file must contain the C file with your solution and screenshots of the valid outputs of the program.