

A1 (28 marks)

Focus: intro to C

- Q1. Write a sequential C code that reads 4 integers from the user and then prints their average along with how many integers are above average. (8 marks)

Sample run1 (green text is user input):

Enter 4 integers separated by spaces: 1 2 3 10
There is 1 entry above the average (4.0)

Sample run2:

Enter 4 integers separated by spaces: 2 4 7 8
There are 2 entries above the average (5.3)

Marking Guide: +2 for reading integers, +2 for finding sum and avg, +4 for counting and printing results.

- Q2. Write a sequential C code that prompts the user to enter a single integer, n . Based on this input, your program should print a pyramid of asterisks (*) that has n rows. Each row should be centered with respect to the pyramid. Assume that user input will be positive. (8 marks)

Sample run1 (green text is user input):

Enter number of rows for pyramid: 4
Output:
*

Sample run2:

Enter number of rows for pyramid: 2
Output:
*

Marking Guide: +2 for reading integers, +2 for finding the loops, +4 for printing results.

- Q3. Write sequential C code to add 2 vectors A and B with $A[i] = i * 3$ and $B[i] = -i * 3$ as $C[i] = A[i] + B[i]$. Also find the sum of all elements in C . Eventually, your code should print the sum (which is 0) as well as the execution time.

Run your code four times with vector sizes of 1, 10, 50, and 200 million integers. Report the results from these four runs as a comment in your code. (12 marks)

Sample run 1: (successful)

Sum: 0
Execution time: 0.88 sec

Sample run 2: (unsuccessful memory allocation)

Not enough memory.

Notes:

- Lookup the function `clock()` from `time.h`.
- Create your arrays using the syntax: `int* R = (int*) malloc(n * sizeof(int))`, where n is the number of integers in the array R .
- Write code to print the error message if the operating system fails to allocate memory to your arrays, in which case `malloc` will return `NULL`.

Marking Guide: +2 for creating arrays, +2 for error checking, +2 for rest of code, +2 for time computation, +4 for reporting correct results as a comment

Submission Instructions

For this assignment, you need to do the following:

- 1- *Programming questions:* Create one C file for *each* programming question and write your answer inside that file. Your files should have the same name as the question number (e.g., Q1.c)
- 2- Also put the solution of each of the answers in a .txt file just in case that there is formatting issues (e.g., Q1.txt).
- 3- Non-programming questions:
 - a. If there are any discussion/essay questions related to a programming question, write your answers as comments at the end of your code for that question.
 - b. For all other non-programming questions (i.e., not related to any programming question), write your answers to all of them in *one* Word document file,
- 4- After solving all questions, compress all your files into one zip folder and give a name to the zipped file that matches your ID (e.g., 1234567.zip).
- 5- Submit the zipped file **to Canvas**.

Note that you can resubmit an assignment, but the new submission overwrites the old submission and receives a new timestamp.