PROBLEM 1: Array Fundamentals

Que.1 Assume you are given the array lower bound (lb), upper bound(ub), base address(ba), words size(ws) and memory location(loc). Write a program to find the following parameters.

(a) Number of elements in the array

(b) Total memory consumed to store the numbers.

(c) Memory location of a[i] (lb<=i<=ub)

The first argument of your command line argument will specify which problem needs to be executed. The number of arguments will change depending on the subproblem, hence ***ensure the desired sanity checks in your code***. If the program fails the sanity check, it should just print **"error",** nothing less or more. Please note that "error" is different from "ERROR" and "Error" in C.

Test Case1: ./a.out 1 -15 64 459 2 10 *(First argument is the question number)*

(Here, problem number = 1, lb=-15, ub= 64, ba= 459, ws= 2,loc = 10)

Output:

80

160

509

Note: Please note the minus sign in the lower bound.

Test Case2: ./a.out 1 39 200 100 4 80

Output:

162

648

264

Test Case 3: ./a.out 1 10 20 2140 2 5 4

Output:

Error

Test Case 4: ./a.out 10 20 2140 2 5 4

Output:

error

Que.2 Write a C program to find the address of location a[I][J] of a 2-d matrix which is stored in a matrix by row major order. Following values are taken as inputs from the user.

- **(a) Number of rows (R)**
- **(b) Number of column (C)**
- **(c) Base address (BA)**
- **(d) Word Size (W)**
- **(e) I**
- **(f) J**

Test Case 1: ./a.out 2 10 20 2140 2 5 4 *(The first argument is the question number)*

Output: 2348

Test Case 2: ./a.out 2 20 30 1000 2 7 4

Output: 1428

Que.3 Write a C program to find the address of location a[i][j] in a 2-d matrix which is stored in a matrix by column major order. Following values are taken as input from the user.

- **(a) Number of rows (R)**
- **(b) Number of column (C)**
- **(c) Base address (BA)**
- **(d) Word Size (W)**
- **(e) I**
- **(f) J**

Test Case 1: .a/.out 3 10 20 2140 2 5 4 *(The first argument is the question number)*

Output: 2230

Test Case 2: ./a.out 3 20 30 1000 2 7 4

Output: 1174

==

SUBMISSION INSTRUCTIONS

Name your C file as [your_roll_number]_[your_name]_p1.c. The question number will be decided by the first command line argument.

Place the file in a folder named [your_roll_number]_[your_name]_lab2. Zip the folder with the name [your_roll_number]_[your_name]_lab2 and upload it to the Google classroom. Please note that the file name convention is slightly different from the last time. Kindly use your complete roll number as assigned by the institute.

Ensure proper sanity checks in all your codes.

==

For Further Practice (Need not submit)

Que.4 Read two sets of numbers and store them into two arrays say A and B. Find the sets C and D such that C=A∪B and D=A∩B, where ∪ and ∩ denote the union and intersection operations on sets. Stores the results into two arrays say C and D. Print the arrays C and D.

Input: A= [2,5,7,9]

B= [3,5,6,8]

Output:

C= [2,3,5,6,7,8,9]

D= [5]

Input: A= [3,14, 7,9]

B= [7, 22, 3]

Output:

C= [3,7,9,14,22]

D=[3, 7]

Que5 : Given a 1-D array, write a program to find the kth maximum element in this array. The input to the program will be

1. Size of array
2. Array elements
3. k

Test Case 1: .a/.out 10 4 6 1 3 8 9 23 46 90 100 2

Output: 90

Test Case 2: .a/.out 10 4 6 1 3 8 9 23 46 90 100 5

Output: 10

Test Case 3: .a/.out 10 4 6 1 3 8 9 23 46 90 100 -1

Output: error

Test Case 4: .a/.out 7 4 6 1 3 8 9 23 46 90 100 2

Output: error