

CS2710 - Programming and Data Structures Lab

Lab Session - 2

Aug 11, 2021

Instructions

- This lab is **graded**. You are expected to solve ALL the three problems on repl.it using C++.
 - The questions are based on your training in programming in CS1111. So no additional inputs are needed, except the changes needed to switch from C to C++.
 - You are expected to solve each problem on your own. If you need assistance, ask your TA, not your classmate.
 - For all the problems the maximum size of the test case is 10000.
 - For Q1, there are 6 public test cases. You get 0.5 mark for each test case. For Q2, TAs will evaluate 2 private test cases for 0.5 marks each and also verify that your code uses the "efficient idea". The 1 mark for Q2 is reserved for the efficient idea. Finally Q3 will have 6 public test cases worth 3 marks and 4 private test cases worth 2 marks. Note that any correct program is fine for Q3.
-

Problems to be solved in lab

1. [3] In a parade ground, army soldiers of Regiment-A are standing in a line in **increasing** order of their height (from left to right). The soldiers of Regiment-B are standing in **decreasing** order of their height again from left to right. For some reason these two regiments merge without disturbing the relative orders amongst themselves. That is, the soldiers in Regiment-B stand to the right of the last soldier in Regiment-A. The Lieutenant arrives and wants the tallest soldier to lead the parade. You need to write program to help the Lieutenant in finding the position (or index) of a soldier with maximum height.

Input format: First line of the input contains a single integer N, denoting the number of the soldier. Next line contains N space separated integers like 10 21 34 23 53. The sequence of integers a_0, a_1, \dots, a_{N-1} satisfies the property that there exists an index $0 \leq p \leq N - 1$ such that $a_0 < a_1 < a_2 < \dots < a_p$ and $a_p > a_{p+1} > \dots > a_{n-1}$.

Example :

Input :

7

100 200 340 780 600 550 250

Output :

3

Explanation :

Soldier with maximum height (780) is standing at position 3.

Note:

Indexing starts with index zero i.e. 0.

Your code should consider the possibility of Regiment-A or Regiment-B having no soldiers at all.

2. (2 points) In this question the input, output and goal are exactly the same as Question 1. The test cases are also exactly the same. However, you are expected to write code which is "better than" the one in the previous program. **Hint:** Think of binary search that you have studied. Can you use a similar idea?
3. (5 points) **Tug of war** is a sport that pits two teams against each other in a test of strength: teams pull on opposite ends of a rope, with the goal being to bring the rope a certain distance in one direction against the force of the opposing team's pull.

Mr. Krishnamurti, a sports faculty, asked students of Class 10 to stand in a straight line. In order to conduct a fair game Mr. Krishnamurti has to divide students in such a way that the sum of weight of students on the left side is equal to sum of weight of students on the right side. Since the game requires a referee Mr. Krishnamurti has decided to select a student as Referee who is standing at a point of equilibrium i.e. sum of weight of student in left side is equal to sum of weight of student in right side. You need to help Mr. Krishnamurti in selecting a student as Referee.

Write a program to return the position (or index) of the Referee if one exists. If no such split is possible, return -1.

Note:

The students cannot be reordered.

Indexing starts with index zero i.e. 0.

Weight of students can be negative.

Input format : First line of the input contains a single integer N, denoting the number of students. Next line contains N space separated integers like 10 21 34 23 53.

Example 1:**Input :**

7
6 2 5 2 4 3 6

Output :

3

Explanation :

Weight of students are given as : Wt []= 6, 2, 5, 2, 4, 3, 6

As, $Wt[0] + Wt[1] + Wt[2] = Wt[4] + Wt[4] + Wt[6]$

Referee position is : 3

Example 2:**Input :**

7
-7 12 4 6 -4 3 0

Output :

2

Explanation :

Weight of students are given as : Wt []= -7, 12, 4, 6, -4, 3, 0

As, $Wt[0] + Wt[1] = Wt[3] + Wt[4] + Wt[5] + Wt[6]$

Referee position is : 2

Example 3:**Input :**

3
1 2 3

Output :

-1

Explanation : No split is possible.

This part is for students to try and play around with their codes. This is not needed for the lab. You can use the following code for timing parts of your code. Note that you are not expected to output the time as a part of the output.

```
#include<iostream>
#include <chrono>
using namespace std::chrono;
using namespace std;

int main() {
    auto start = high_resolution_clock::now();

    for (int i=0; i<100000; i++) {
        // do nothing.
    }
    auto stop = high_resolution_clock::now();
    auto duration = duration_cast<microseconds>(stop - start);
    cout << "Time taken by above code: "
         << duration.count() << " microseconds" << endl;
}
```