

CSE 4202-Structured Programming-II Lab Summer 2022-23

Lab Quiz-2

Date: April 16, 2024 (Tuesday)

Target Group: 1B

Topic: Lab Quiz-2

Instructions:

- Task naming format: fullID_T01LQ2_1B.c
- If you find any issue in the problem descriptions/test cases, comment in the google classroom.
- Please comment if you find any tricky test cases that were not included and others might forget to handle.
- Modified portions will be marked in **BLUE** color.

Distribution:

Group	Lab Task
1B	1, 2, 3

Notes:

Use only C.

Task 1 — Subarray Sum

Given an unsorted array A that contains only **positive integers**, find a continuous sub-array that adds to a given number S and return the left and right indices (0-based indexing) of that subarray.

In the case of multiple subarrays, print the subarray indices that come first on moving from left to right. In the case no such subarray exists, print -1.

Input

First line containing one integer S , the target sum, followed by some positive integers in the next line. This line always ends with -1, so you should keep taking array inputs until you find -1. These positive integers are the elements of the array A , and there can be an arbitrary number of such elements. Since the size of the array is unspecified, you need to use a **Dynamically Growing Array**.

Output

Starting and ending indices of the target subarray if found, -1 otherwise.

Sample Input	Sample Output
10 1 7 5 3 2 11 9 0 -1	2 4
11 9 1 3 5 5 -1	-1
2 1 2 3 4 -1	1 1
8 9 2 5 6 1 3 4 4 7 10 2 6 5 3 200 2 -1	4 6

Note: If you don't want to use -1 as the sentinel value, then you can write EOF in the terminal or console using **Ctrl+Z**.

You'll see this appear as a result — ^Z.

Task 2 — Biggie Smalls

Given a string S consisting of only uppercase and lowercase letters, the task is to sort the uppercase and lowercase letters separately such that if the i^{th} place in the original string had an uppercase character, then it should not have a lowercase letter after being sorted and vice versa.

Input

A single string containing only upper and lowercase letters. The length of the string will not be more than 1024 characters.

Output

The characters within the string sorted, maintaining the positions of uppercase and lowercase letters.

Sample Input	Sample Output
defRTSersUXI	deeIRSfrsTUX
srbDKi	birdKs
NMSFKLMNkmkfasmfFMSKLFMNAadasfADFAS	AAADFFFFaaadffkKKLLMMMMNkmmssNNSSS

Task 3 — Don't Be Positive, Be Negative

Your task is to invert a certain portion of a given grayscale image. For this task, you'll be working with the **100dollars.tif** 8-bit grayscale image file that was previously provided in the Google Classroom. If the file isn't already downloaded in your PC, then use the link provided below.

(Link:

<https://drive.google.com/file/d/1RPesQOFH6OhfcgBVd53lj7WmAl5cZq3J/view?usp=sharing>).

Recall that the dimensions of this image is 500×1192 , which means it has a height of 500 pixels and a width of 1192 pixels. You'll be given the coordinates of a seed point $S \equiv (x, y)$ and the dimensions $m \times n$ of the subimage which you have to invert by keeping the seed point S at the **utmost top-left** of the subimage. The negative of a particular k -bit pixel with the intensity level value p is,

$$p_{neg} = 2^k - 1 - p$$

Input




The first line contains an integer T denoting the number of operations, followed by T seed points and subimage dimensions in the following format:

- $x \ y \ m \ n$

It is guaranteed that the seed points $S_i \equiv (x_i, y_i)$ are within the confines of the image boundary. However, the subimage region may trespass the boundaries of the image.

Output

For each seed point, invert the region covered by the subimage of the corresponding dimensions.

Sample Input	Sample Output
1 69 420 250 500	
4 100 100 450 450 450 1000 1000 1000 100 911 250 250 0 0 69 69	
2 125 350 350 350 40 490 350 350	
4 0 0 500 298 0 298 500 298 0 596 500 298 0 894 500 298	