CSE221 Assignment 02 Spring 2025

A. Two Sum Trouble

1 second, 256 megabytes

Your little brother, Bob, loves playing with integers. One day, his teacher gave him a sorted list of ***N*** integers in **non-decreasing** order. Now, your brother wants to play a game with you.

Bob will give you an integer ***S***. You have to find if it is possible to find two values from the list (at distinct positions) whose sum is equal to ***S***.

Since you are feeling very tired, you decide to write a program that can quickly answer Bob's query.

Input

The first line contains two integers ***N*** $(1 \leq N \leq 10^6)$ and ***S*** $(1 \leq S \leq 10^9)$, denoting the length of the list, and the target Sum.

In the next line, there will be N integers $a_1, a_2, a_3 \ldots a_n$ $(1 \leq a_i \leq 10^9)$ in non-decreasing order, separated by spaces.

Output

Print two distinct 1-based indices ***i*** and ***j*** such that $a_i + a_j = S$ where $i < j$. If no such pair exists, then print ***-1***. If multiple solutions exist, you may print any one of the valid answers.

input
4 10 1 3 5 7
output
2 4

input
6 18 1 5 8 9 9 10
output
3 6

input
4 7 2 4 6 8
output
-1

input
4 10 1 5 6 8
output
-1

In the second sample input, 4 5 is also a valid output.

B. A Beautiful Sorted List

1 second, 1024 megabytes

Alice and Bob are two friends. Alice has a list of length ***N*** in **non-decreasing** order, and Bob has a list of length ***M***, also in **non-decreasing** order.

Now, they want to combine their lists into a single non-decreasing list of length **N+M**. However, they are not very good at algorithms, so they asked for your help.

Since you are a computer science student, your task is to write an efficient algorithm to merge the two given lists into one non-decreasing list. **Solve the problem in O(N+M)**.

Input

The first line contains an integer ***N*** $(1 \leq N \leq 10^6)$, denoting the length of Alice's list.

The second line contains N space-separated integers representing Alice's list.

The third line contains an integer ***M*** $(1 \leq M \leq 10^6)$, denoting the length of Bob's list.

The fourth line contains M space-separated integers representing Bob's list.

All the numbers given in the input will fit within a **32-bit signed integer**. It is guaranteed that the given lists will be in **non-decreasing** order.

Output

You have to make a sorted list in **non-decreasing** order from the given lists and show the output.

input
4 1 3 5 7 4 2 2 4 8
output
1 2 2 3 4 5 7 8

input
3 2 10 12 6 3 4 6 7 8 9
output
2 3 4 6 7 8 9 10 12

input
5 1 2 3 4 5 2 10 12
output
1 2 3 4 5 10 12

input
4 1 2 12 13 3 10 15 18
output
1 2 10 12 13 15 18

input
8 1 2 3 8 8 10 12 14 9 1 1 4 5 6 8 13 15 16
output
1 1 1 2 3 4 5 6 8 8 8 10 12 13 14 15 16

C. Longest Subarray Sum

1 second, 256 megabytes

You are given an array of ***N*** integers and an integer ***K***. Your task is to find the length of the longest contiguous subarray whose sum is less than or equal to ***K***.

Input

The first line contains two integers ***N*** $(1 \leq N \leq 10^5)$ and ***K*** $(1 \leq K \leq 10^9)$ — the size of the array and the maximum allowed sum.

The second line contains N space-separated integers $a_1, a_2, a_3 \ldots a_n$ $(1 \leq a_i \leq 10^6)$ — the elements of the array.

Output

Print a single integer — the length of the longest contiguous subarray whose sum is less than or equal to ***K***.

input
5 4 4 1 2 1 5
output
3

input
5 5 1 1 1 1 1
output
5

input
3 1 2 3 4
output
0

input
10 12 1 2 6 4 3 2 3 1 4 2
output
5

In the first example, possible subarrays with sum less than or equal to 4 are $[4], [1], [2], [1], [1, 2], [2, 1], [1, 2, 1]$. Among them, the longest size is 3.

In the second example, sum of the entire array is 5. Hence, we can take the whole array.

In the third example, no subarray has sum less than or equal to 1. Hence, the answer is 0.

D. Can you Iterate the Binary String?

0.5 seconds, 256 megabytes

You are given ***T*** test cases. Each test case contains a binary string ***S*** that follows a specific pattern:

- There will be zero or more 0s in the prefix of ***S***.
- There will be zero or more 1s in the suffix of ***S***.

For each string, find the index of the first occurrence of the character 1 in the **1-based indexing**. Find the output of each query in $O(log|S|)$.

Input

The first line contains an integer ***T*** $(1 \leq T \leq 10^4)$ — the number of test cases. Each of the next T lines contains a binary string ***S*** $(1 \leq |S| \leq 4 \times 10^3)$, where |S| represents the length of the string.

Output

For each test case, print a single integer:

- The first occurrence of the character 1 in the string S in the 1-based indexing.
- If there is no 1 in the string, print -1.

input
15 0000011111111 00000111111111 00000 0000 1111 111 0 1 01 01111 000000000000001 0000000000000001 00000000000001111111111111111111111 000000011111111111111111 0000000111111111111111111111111111
output
6 6 -1 -1 1 1 -1 1 2 2 15 16 14 8 8

E. Count the Numbers

1 second, 256 megabytes

You are given a sorted array a of n elements, and some queries. In each query, you are given a pair $[x, y]$ and you have to count how many numbers a_i are there such that $x \leq a_i \leq y$. For example, if the array is $[10, 20, 20, 45, 79]$ and you are given a query $[20, 50]$, then answer will be 3 because there are in total 3 numbers that's value is between 20 and 50.

Input

The first line of the input contains $n(1 \le n \le 10^5)$ and $q(1 \le q \le 10^5)$ denoting the array size and the number of queries respectively. The next line will contain the array elements separated by space where $1 \le a_i \le 10^9$ where $i = 0, 1, 2, \ldots, n-1$. Each of the next q lines will contain a pair $[x, y]$ where $1 \le x \le y \le 10^9$. See the sample input format for better understanding.

Note1: It is guaranteed that the given array is sorted in **non-decreasing** order.

Note2: It is also guaranteed that the queries are valid. Which means, for each query $[x, y]$, $x \le y$.

Output

For each query $[x, y]$, output a single integer P denoting the number of elements in the array a such that $x \le a_i \le y$.

input
5 3 10 20 20 45 79 20 50 5 45 1 100

output
3 4 5