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*** $(1 \le N \le 10^6)$ and ***S*** $(1 \le S \le 10^9)$, denoting the length of the list, and the target Sum.

In the next line, there will be N integers $a_1, a_2, a_3 \dots a_n$ $(1 \le a_i \le 10^9)$ 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. Two Sum Revisited

1 second, 256 megabytes

You are given two integer arrays A and B of sizes N and M, respectively, and an integer K. Both arrays are sorted in non-decreasing order. Your task is to find any pair of indices (i, j) such that:

- i is a valid index in array A $(1 \le i \le N)$
- j is a valid index in array B $(1 \le j \le M)$
- the sum A[i] + B[j] is closest to K (i.e., it minimizes |A[i] + B[j] − K|).

Input

The first line contains Three integers N, M and K $(1 \le N, M \le 2 \times 10^5, -10^9 \le K \le 10^9)$.

The second line contains N integers — the elements of array A $(-10^9 \le A_i \le 10^9)$.

The third line contains M integers — the elements of array B $(-10^9 \le B_j \le 10^9)$.

Output

Print two space-separated integers i and j the 1-based indices of the chosen pair.

If there are multiple answers, output any.

input
```
4 4 0
-5 -2 -1 5
-5 0 1 1
```
output
```
3 4
```

input
```
6 6 1
-5 -3 3 4 4 5
-2 0 2 2 3 5
```
output
```
3 1
```

input
```
1 1 8
-2
-8
```
output
```
1 1
```

input
```
2 2 -4
-7 4
-5 4
```
output
```
1 2
```

C. Triple The Trouble

1 second, 256 megabytes

You are given an array of n integers, and your task is to find three values (at distinct positions) whose sum is x.

Input

The first input line has two integers n $(1 \le n \le 5000)$ and x $(1 \le x \le 10^9)$, the array size and the target sum. The second line has n integers $a_1, a_2, \dots, a_n$ $(1 \le a_i \le 10^9)$, the array values.

Output

Print three integers: the positions of the values. If there are several solutions, you may print any of them. If there are no solutions, print -1.

input
```
7 3
2 1 1 2 2 1 1
```
output
```
2 3 6
```

input
3 5 1 3 2
output
-1

D. 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*** $(1 \le N \le 10^6)$, 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*** $(1 \le M \le 10^6)$, 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

E. 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*** $(1 \le N \le 10^5)$ and ***K*** $(1 \le K \le 10^9)$ — the size of the array and the maximum allowed sum.

The second line contains N space-separated integers $a_1, a_2, a_3 \dots a_n$ $(1 \le a_i \le 10^6)$ — 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.

F. Longest K-Distinct Subarray

1 second, 256 megabytes

You are given an array of integers of length N and an integer K. Your task is to find the length of the longest contiguous subarray that contains at most K distinct elements.

Input

The input consists of:

The first line contains two integers N and K — the size of the array and the maximum number of distinct elements allowed $(1 \le N \le 2 \times 10^5, 1 \le K \le N)$.

The second line contains N space-separated integers $A_1, A_2, A_2 \ldots A_n$ — the elements of the array $(1 \le A_i \le N)$.

Output

Print a single integer — the length of the longest contiguous subarray that contains at most K distinct elements.

input
```
4 1
2 1 2 4
```
output
```
1
```

input
```
6 2
6 6 5 6 1 2
```
output
```
4
```

input
```
1 1
1
```
output
```
1
```

input
```
2 2
1 2
```
output
```
2
```

G. 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 $x \le a_i \le y$. 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
```

H. Searching is Fun

1 second, 256 megabytes

You are given two positive integers: k and x. Consider the sequence of all positive integers that are not divisible by x. Your task is to find the k-th number in this sequence.

For example, if $x = 3$, and $k = 7$, then all numbers that are not divisible by 3 are: $1, 2, 4, 5, 7, 8, 10, 11, 13 \ldots$. The 7-th number among them is 10.

Input

The first line contains a single integer T $(1 \le T \le 10^5)$ — the number of test cases.

Each of the next T lines contains two integers k and x $(1 \le k \le 10^9, 1 < x \le 10^9)$— the position in the sequence and the divisor to avoid.

Output

For each test case, output a single integer — the k-th positive integer that is not divisible by x.

input
```
6
7 3
100 5
49 13
36 2
1 2
11 11
```
output
```
10
124
53
71
1
12
```