

# Iris and Adjacent Products

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            **3 seconds**  
Memory limit:         **256 megabytes**

Iris has just learned multiplication in her Maths lessons. However, since her brain is unable to withstand too complex calculations, she could not multiply two integers with the product greater than  $k$  together. Otherwise, her brain may explode!

Her teacher sets a difficult task every day as her daily summer holiday homework. Now she is given an array  $a$  consisting of  $n$  elements, and she needs to calculate the product of each two adjacent elements (that is,  $a_1 \cdot a_2$ ,  $a_2 \cdot a_3$ , and so on). Iris wants her brain to work safely, and in order to do that, she would like to modify the array  $a$  in such a way that  $a_i \cdot a_{i+1} \leq k$  holds for every  $1 \leq i < n$ . There are two types of operations she can perform:

1. She can rearrange the elements of the array  $a$  in an arbitrary way.
2. She can select an arbitrary element of the array  $a$  and change its value to an arbitrary integer from 1 to  $k$ .

Iris wants to minimize the number of operations of **type 2** that she uses.

However, that's completely not the end of the summer holiday! Summer holiday lasts for  $q$  days, and on the  $i$ -th day, Iris is asked to solve the Math homework for the subarray  $b_{l_i}, b_{l_i+1}, \dots, b_{r_i}$ . Help Iris and tell her the minimum number of type 2 operations she needs to perform for each day. Note that the operations are **independent** for each day, i.e. the array  $b$  is not changed.

## Input

Each test consists of multiple test cases. The first line contains a single integer  $t$  ( $1 \leq t \leq 5 \cdot 10^4$ ) — the number of test cases. The description of the test cases follows.

The first line of each test case contains three integers  $n$ ,  $q$  and  $k$  ( $2 \leq n \leq 10^5$ ,  $1 \leq q \leq 10^5$ ,  $1 \leq k \leq 10^6$ ) — the length of array  $b$ , the number of days, and the upper bound for the multiplication calculation.

The second line of each test case contains  $n$  integers  $b_1, b_2, \dots, b_n$  ( $1 \leq b_i \leq k$ ) — the elements of the array  $b$ .

Then  $q$  lines follow, the  $i$ -th of them contains two integers  $l_i$  and  $r_i$  ( $1 \leq l_i < r_i \leq n$ ) — the boundaries of the subarray on the  $i$ -th day.

It is guaranteed that the sum of  $n$  over all test cases does not exceed  $10^5$ , and the sum of  $q$  over all test cases does not exceed  $10^5$ .

## Output

For each test, output a single line containing  $q$  integers — the minimum number of operations of type 2 needed for each day.

## Example

standard input	standard output
5	0
3 1 1	0 1
1 1 1	1 1 2 2
1 3	1 1 1 1 0
3 2 10	3 3 4 3 2 2 1 1 2 1
1 10 9	
1 3	
2 3	
5 4 2	
2 2 2 2 2	
1 2	
2 4	
2 5	
1 5	
6 5 10	
3 2 5 10 10 1	
1 4	
3 6	
1 6	
2 5	
5 6	
10 10 10	
10 9 8 7 6 5 4 3 2 1	
1 10	
1 9	
1 8	
1 7	
2 10	
3 10	
4 10	
5 10	
3 9	
6 8	

## Note

In the first test case, as Iris can always multiply 1 and 1 together, no operations are needed, so the answer is 0.

In the second test case, the first day's homework is [1, 10, 9]. Iris can rearrange its elements to get [9, 1, 10], so no operations of type 2 are needed. The second day's homework is [10, 9], and she can change one element to get the array [1, 9], so one operation of type 2 is needed.