Welcome to Code Fest 6.0

"First, solve the problem. Then, write the code."

– John Johnson

<CODE FEST 6.0/>

OPEN HOUSE UNIVERSITY COMPETITION

Organized by FCIT - University of the Punjab

Programming Competition

FCIT

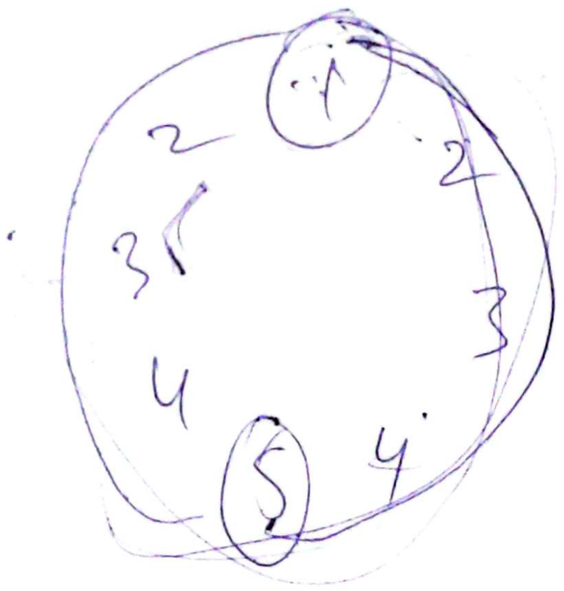
University of the Punjab

$a_{i-1} \le a_i \le a_{i+1}$

$a_{i-1} < a_i > a_{i+1}$

$a_i > \max(a_{i-1}, a_{i+1})$, violation :

$[1, 2, 3, 4, 5]$



A - Muneeb's Road Trip Riddle

Muneeb lives at point 0 on a long road that goes up to point n, starting from his house. In his neighborhood, people get around using special hopping balloons. These balloons jump a certain number of points depending on how many coins you spend on them.

For example, if a balloon costs 10 coins, it jumps 10 points each time. So, if Muneeb buys this balloon, it will carry him from point 0 to 10, then to 20, and so on, all the way to the end of the road. Plus, these balloons can hop backward and forward, too, like from point 20 to 10 if needed.

Now, Muneeb wants to visit his k relatives, each living at a different point labeled as $r_1, r_2, r_3, \ldots, r_k$. Where r_i denotes the point where ith relative lives. But here's the twist: he can only buy one balloon from the shop because of a rule. Since he's in a hurry and wants to make the best choice, he's asking you for help. Can you tell him how many coins he should spend on the balloon to reach all his relatives in the minimum number of jumps?

Input:

The first line contains the number of test cases t $(1 \le t \le 100)$. The description of the test cases follows.

The first line of each test case contains two integers $\boxed{n,\ k}$ $(1 \le n \le 10^{17}, \boxed{1 \le k \le 2 * 10^5})$ — the length of road and the number of relatives.

The second line of each test case contains k integers $r_1, r_2, r_3, \ldots, r_k$ $(1 \le r_i \le 10^{17})$ — the point where i^{th} relative lives.

Output:

For each test case, output a single line containing an integer: the amount of coins Muneeb should spend to shop the balloon.

Sample 1:

Input	Output

2
10 4
6 4 2 8
7 3
3 2 6
	2
	1

Road

no. of relative

max → nops's ave.

jo 1 Cey sub 1 ~ divisible no.

Explanation:

In test case 1, Muneeb shops a balloon using 2 coins. Muneeb can visit all of his relatives in 4 jumps [0→2→4→6→8] which can be shown as minimum.

In test case 2, Muneeb shops a balloon using 1 coin and visits all his relatives in 6 jumps which are the minimum number of jumps. Muneeb takes following jumps: [0→1→2→3→4→5→6]

1 3 5 7 9

5 7 0 2 3 4 5 2 9

B - Pass The Ball

On a sunny day in the park, n friends gathered for a picnic. They ordered the food and decided to play a game to stave off boredom.

Here's how the game works: All n friends who are playing gather in a line, with the ball initially in the hands of the first player. The first player passes the ball to the second, the second to the third, and so on until it reaches the last player. Since there's no one standing after the last player, they pass the ball back to the second last player. This pattern continues, with the ball moving back and forth among all the players. Each transfer of the ball from one player to another, whether forward or backward, counts as one move.

-3

-1

3 2 0 1 2 3

2 3

For example, let's say there are 5 friends. At the start, friend 1 has the ball. In the first move, friend 1 passes the ball to friend 2. In the second move, friend 2 passes the ball to friend 3, and so on, until it reaches friend 5. Then, friend 5 passes it back to friend 4, and the backward pass begins: [5→4→3→2→1]. Then, the cycle repeats with the forward pass [1→2→3→4→5].

Given the number of friends and the current move, you need to determine which friend currently has the ball and to whom they are passing it.

2 3 4 5

Input Format

The first line contains the t $(1 \le t \le 100)$ number of test cases.
Each test case contains number of the friends n $(2 \le n \le 2 * 10^5)$ and the m $(1 \le m \le 10^{12})$ move.

(1, 2, 3, 4, 5)

Output Format

The friend number who passed the ball at the given move and the friend number who received the ball space-separated.

2 3

Sample 1:

Input	output

2
3 2
5 7
	2 3
	3 2

7 - 5

cur + 1

even → 2nd

odd → 1st

[1, 2, 3, 4, 5, 4, 3, 2, 1]

(1, 2, 3, 4, 5)

Explanation:

Test Case 1: The first 2 moves look like this [1→2→3]. In move 2, the ball was passed from friend 2 to friend 3, so the answer is 2 3.

Test Case 2 : The first 7 moves look like this [1→2→3→4→5→4→3→2]. In move 7, the ball was passed from friend 3 to friend 2, so the answer is 3 2

C - Farhan and Water Bottles

Farhan has N empty bottles where each bottle has a capacity of X liters.
There is a water tank in "Farhan's Land" having K liters of water. Farhan wants to fill the empty bottles using the water in the tank.

Assuming that Farhan does not spill any water while filling the bottles, find out the maximum number of bottles Farhan can fill completely.

Input Format

No. of bottles.

First line will contain T, number of test cases. Then the test cases follow.
Each test case contains of a single line of input, three integers $N, X,$and K. K [illegible]

capacity X.

Output Format

For each test case, output in a single line answer, the maximum number of bottles Farhan can fill completely.

Constraints

$1 <= T <= 100$

$1 <= N, X <= 10^5$

$0 <= K <= 10^5$

for in 5 :

8 - 2

Sample 1:

Input	Output

2
5 2 8
10 5 4
	4
	0

Explanation:

Test Case 1: The amount of water in the tank is 8 liters. The capacity of each bottle is 2 liters. Hence, 4 water bottles can be filled completely.

Test Case 2: The amount of water in the tank is 4 liters. The capacity of each bottle is 5 liters. Hence, no water bottle can be filled completely.

D - Hidden Array

Zubair unexpectedly saw a present from one of his previous birthdays. It is array of n numbers from 1 to 200. Array is old and some numbers are hard to read. Zubair remembers that for all elements at least one of its neighbors Is not less than it, more formally:

$a_1 \le a_2$

$a_n \le a_{n-1}$

$a_i \le max(a_{i-1}, a_{i+1})$ for all i from 2 to $n - 1$.

Zubair does not remember the array and asks to find the number of ways to restore it. Restored elements also should be integers from 1 to 200. Since the number of ways can be big, print it modulo 998244353.

Input

First line of input contains one integer N $(2 \le N \le 10^5)$ — size of the array.

Second line of input contains N integers a_i — elements of array.

Either $a_i = -1$ or $1 \le a_i \le 200$. $a_i = -1$ means that i-th element can't be read.

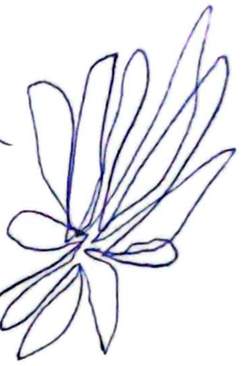
Output

Print number of ways to restore the array modulo 998244353.

8

K 4

6



Sample 1:

3

1 -1 2 1

Sample 2:

2

-1 -1 200

Note

In the first example, only possible value of a_2 is 2.

In the second example, $a_1 = a_2$ so there are 200 different values because all restored elements should be integers between 1 and 200.

-3

2 9

4

1

2^n - 2

E1 - Product Game (easy version)

The only difference b/w the easy and hard version is constraint on the elements of the array. In the hard version array can have element 0 but in easy version array doesn't have 0.

Given an array of integers and an integer k, return the number of contiguous subarrays where the product of all the elements in the subarray is **strictly less than** k.

Input

First line of input contains one integer T $(1 \leq T \leq 100)$ — number of test cases.

Second line of input contains n (size of array) and k $(1 \leq k \leq 10^6)$.

Third line contains n integers $a_1, a_2, \ldots, a_n$. $(1 \leq a_i \leq 10^5)$.

Output

Print number of subarrays having product strictly less than k.

Input	Output

2
4 10
3 3 2 1
2 1
2 2
	8
	0

cit 1.
2
3 3
3 3
3 2

Explanation:

1- In first test case these contiguous subarrays has product less than 10
(3), (3), (2), (1), (3, 3), (3, 2), (2, 1), (3, 2, 1)

2- In second case there is no subarray having product less than 1

3 3 2 10

15

3 9 18 | 18 6 6

9
1 9 4

3 3 1 4 2

E2 - Product Game (hard version)

The only difference b/w the easy and hard version is constraint on the elements of the array. In hard version array can have element 0 but in easy version array doesn't have 0.

Given an array of integers and an integer k, return the number of *contiguous subarrays* where the product of all the elements in the *subarray* is ***strictly less than*** k.

Input

First line of input contains one integer T $(1 \leq T \leq 100)$ — number of test cases.

Second line of input contains n (size of array) and k $(1 \leq k \leq 10^6)$.

Third line contains n integers $a_1, a_2, ..., a_n$. $(0 \leq a_i \leq 10^5)$.

Output

Print number of subarrays having product strictly less than k.

Input	Output

2
4 10
3 0 2 1
2 1
2 2
	10
	0

Explanation:

1- In first test case these contiguous sub arrays has product less than 10
(3), (0), (2), (1), (3, 0), (0, 2), (2, 1), (3, 0, 2), (0, 2, 1), (3, 0, 2, 1)

2- In second case there is no sub array having product less than 1.