A. Median of an Array

```
Input file:     standard input
Output file:    standard output
Time limit:     1 second
Memory limit:   256 megabytes
```

You are given an array a of n integers.

The *median* of an array $q_1, q_2, \ldots, q_k$ is the number $p_{\lceil \frac{k}{2} \rceil}$, where p is the array q sorted in non-decreasing order. For example, the median of the array $[9, 5, 1, 2, 6]$ is 5, as in the sorted array $[1, 2, 5, 6, 9]$, the number at index $\lceil \frac{5}{2} \rceil = 3$ is 5, and the median of the array $[9, 2, 8, 3]$ is 3, as in the sorted array $[2, 3, 8, 9]$, the number at index $\lceil \frac{4}{2} \rceil = 2$ is 3.

You are allowed to choose an integer i ($1 \le i \le n$) and increase a_i by 1 in one operation.

Your task is to find the minimum number of operations required to increase the median of the array.

Note that the array a may not necessarily contain distinct numbers.

Input

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

The first line of each test case contains a single integer n ($1 \le n \le 10^5$) — the length of the array a.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the array a.

It is guaranteed that the sum of the values of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output a single integer — the minimum number of operations required to increase the median of the array.

Standard Input	Standard Output

8 3 2 2 8 4 7 3 3 1 1 1000000000 5 5 5 5 4 5 6 2 1 2 3 1 4 2 1 2 2 1 1 4 5 5 5 5	1 2 1 3 2 1 2 3

Note

In the first test case, you can apply one operation to the first number and obtain the array $[3, 2, 8]$, the median of this array is 3, as it is the number at index $\lceil \frac{3}{2} \rceil = 2$ in the non-decreasing sorted array $[2, 3, 8]$. The median of the original array $[2, 2, 8]$ is 2, as it is the number at index $\lceil \frac{3}{2} \rceil = 2$ in the non-decreasing sorted array $[2, 2, 8]$. Thus, the median increased ($3 > 2$) in just one operation.

In the fourth test case, you can apply one operation to each of the numbers at indices $1, 2, 3$ and obtain the array $[6, 6, 6, 4, 5]$, the median of this array is 6, as it is the number at index $\lceil \frac{5}{2} \rceil = 3$ in the non-decreasing sorted array $[4, 5, 6, 6, 6]$. The median of the original array $[5, 5, 5, 4, 5]$ is 5, as it is the number at index $\lceil \frac{5}{2} \rceil = 2$ in the non-decreasing sorted array $[4, 5, 5, 5, 5]$. Thus, the median increased ($6 > 5$) in three operations. It can be shown that this is the minimum possible number of operations.

In the fifth test case, you can apply one operation to each of the numbers at indices $1, 3$ and obtain the array $[3, 1, 3, 3, 1, 4]$, the median of this array is 3, as it is the number at index $\lceil \frac{6}{2} \rceil = 3$ in the non-decreasing sorted array $[1, 1, 3, 3, 3, 4]$. The median of the original array $[2, 1, 2, 3, 1, 4]$ is 2, as it is the number at index $\lceil \frac{6}{2} \rceil = 3$ in the non-decreasing sorted array $[1, 1, 2, 2, 3, 4]$. Thus, the median increased ($3 > 2$) in two operations. It can be shown that this is the minimum possible number of operations.

,

B. Maximum Sum

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

You have an array a of n integers.

You perform exactly k operations on it. In one operation, you select any contiguous subarray of the array a (possibly empty) and insert the sum of this subarray anywhere in the array.

Your task is to find the maximum possible sum of the array after k such operations.

As this number can be very large, output the answer modulo $10^9 + 7$.

Reminder: the remainder of a number x modulo p is the smallest non-negative y such that there exists an integer q and $x = p \cdot q + y$.

Input

Each test consists of several test cases. The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. Then follows the description of the test cases.

The first line of each test case contains two integers n and k ($1 \le n, k \le 2 \cdot 10^5$) — the length of the array a and the number of operations, respectively.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$) — the array a itself.

It is guaranteed that the sum of the values of n and k for all test cases does not exceed $2 \cdot 10^5$.

Output

For each test, output a single integer — the maximum sum of the array that can be obtained after k operations modulo $10^9 + 7$.

Standard Input	Standard Output
12 2 2 -4 -7 3 3 2 2 8 1 7 7 5 1 4 -2 8 -12 9 7 4 8 14 -9 6 0 -1 3 7 100 5 3 -8 12 -5 -9 3 6 1000 -1000000000 -1000000000 -1000000000 -1000000000 -1000000000 -1000000000	999999996 96 896 17 351 716455332 42 2 0 897909241 0 416571966

2 1 1000000000 8 5 4 0 0 0 0 0 6 10 48973 757292 58277 -38574 27475 999984 7 1 -1000 1000 -1000 1000 -1000 1000 -1000 10 10050 408293874 -3498597 7374783 295774930 -48574034 26623784 498754833 -294875830 283045804 85938045	

Note

In the first test case, it is advantageous to take an empty subarray of the array twice and insert the sum of the empty subarray (zero) anywhere, then the sum of the resulting array will be $(-4) + (-7) + 0 + 0 = -11$, modulo $10^9 + 7$ this is $999\,999\,996$.

In the second test case, it is advantageous to take the sum of the entire array three times and place it anywhere in the array, then one of the possible sequences of actions: [$2, 2, 8$] $\rightarrow$ [$2, 2, 8, 12$] $\rightarrow$ [$2, 2, 8, 12, 24$] $\rightarrow$ [$2, 2, 8, 12, 24, 48$], the sum of the final array is $2 + 2 + 8 + 12 + 24 + 48 = 96$.

In the fourth test case, it is advantageous to take a subarray of the array consisting of the first three numbers (i.e. consisting of the numbers $4, -2$ and 8) and insert its sum at the beginning of the array, thereby obtaining the array [$10, 4, -2, 8, -12, 9$], the sum of this array is 17.

In the seventh test case, it will always be advantageous for us to take an empty subarray of the array. In this case, the sum of the resulting array will not differ from the sum of the original. The answer will be the sum of the original array, taken modulo — 42, because $(-6 \cdot (10^9 + 7) + 42 = -6\,000\,000\,000)$.

C. Tree Cutting

```
Input file:     standard input
Output file:    standard output
Time limit:     3 seconds
Memory limit:   512 megabytes
```

You are given a tree with n vertices.

Your task is to find the maximum number x such that it is possible to remove exactly k edges from this tree in such a way that the size of each remaining connected component[†] is at least x.

[†] Two vertices v and u are in the same connected component if there exists a sequence of numbers $t_1, t_2, \ldots, t_k$ of arbitrary length k, such that $t_1 = v$, $t_k = u$, and for each i from 1 to $k-1$, vertices t_i and t_{i+1} are connected by an edge.

Input

Each test consists of several sets of input data. The first line contains a single integer t ($1 \le t \le 10^4$) — the number of sets of input data. This is followed by a description of the sets of input data.

The first line of each set of input data contains two integers n and k ($1 \le k < n \le 10^5$) — the number of vertices in the tree and the number of edges to be removed.

Each of the next $n-1$ lines of each set of input data contains two integers v and u ($1 \le v, u \le n$) — the next edge of the tree.

It is guaranteed that the sum of the values of n for all sets of input data does not exceed 10^5.

Output

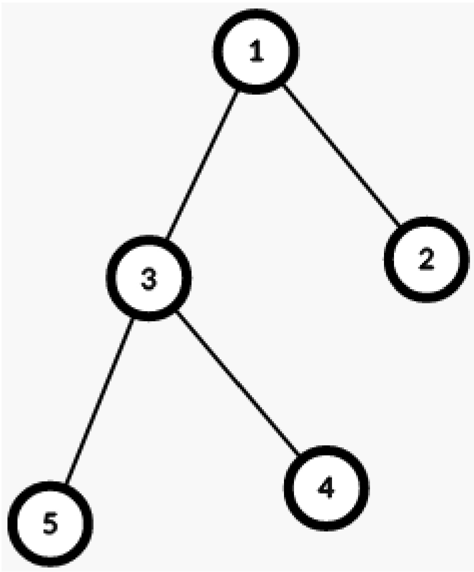
For each set of input data, output a single line containing the maximum number x such that it is possible to remove exactly k edges from the tree in such a way that the size of each remaining connected component is at least x.

Standard Input	Standard Output
6	2
5 1	1
1 2	3
1 3	1
3 4	1
3 5	2
2 1	
1 2	
6 1	
1 2	
2 3	
3 4	
4 5	
5 6	
3 1	
1 2	
1 3	

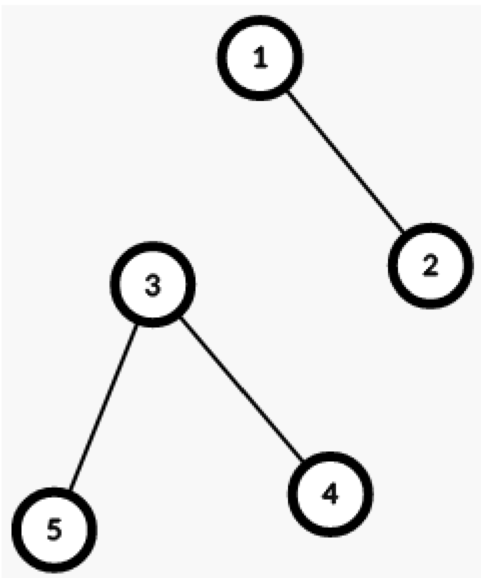
8 2 1 2 1 3 2 4 2 5 3 6 3 7 3 8 6 2 1 2 2 3 1 4 4 5 5 6	

Note

The tree in the first set of input data:



After removing the edge $1 — 3$, the tree will look as follows:



The tree has split into two connected components. The first component consists of two vertices: 1 and 2. The second connected component consists of three vertices: $3, 4$ and 5. In both connected components, there are at least two vertices. It can be shown that the answer 3 is not achievable, so the answer is 2.

,

D. Birthday Gift

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

Yarik's birthday is coming soon, and Mark decided to give him an array a of length n.

Mark knows that Yarik loves bitwise operations very much, and he also has a favorite number x, so Mark wants to find the maximum number k such that it is possible to select pairs of numbers $[l_1, r_1]$, $[l_2, r_2]$, $\ldots$ $[l_k, r_k]$, such that:

- $l_1 = 1$.
- $r_k = n$.
- $l_i \le r_i$ for all i from 1 to k.
- $r_i + 1 = l_{i+1}$ for all i from 1 to $k-1$.
- $(a_{l_1} \oplus a_{l_1+1} \oplus \ldots \oplus a_{r_1}) | (a_{l_2} \oplus a_{l_2+1} \oplus \ldots \oplus a_{r_2}) | \ldots | (a_{l_k} \oplus a_{l_k+1} \oplus \ldots \oplus a_{r_k}) \le x$, where $\oplus$ denotes the operation of bitwise XOR, and $|$ denotes the operation of bitwise OR.

If such k does not exist, then output -1.

Input

Each test consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. The following lines contain the descriptions of the test cases.

The first line of each test case contains two integers n and x ($1 \le n \le 10^5, 0 \le x < 2^{30}$) — the length of the array a and the number x respectively.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i < 2^{30}$) — the array a itself.

It is guaranteed that the sum of the values of n across all test cases does not exceed 10^5.

Output

For each test case, output a single integer on a separate line — the maximum suitable number k, and -1 if such k does not exist.

Standard Input	Standard Output
8 3 1 1 2 3 2 2 1 1 2 2 1 3 2 3 0 0 3 2 0 0 1 4 2 1 3 3 7	2 2 1 2 3 -1 1 2

2 2 2 3 5 0 0 1 2 2 1	

Note

In the first test case, you can take k equal to 2 and choose two segments $[1, 1]$ and $[2, 3]$, $(1)|(2 \oplus 3) = 1$. It can be shown that 2 is the maximum possible answer.

In the second test case, the segments $[1, 1]$ and $[2, 2]$ are suitable, $(1)|(1) = 1$. It is not possible to make more segments.

In the third test case, it is not possible to choose 2 segments, as $(1)|(3) = 3 > 2$, so the optimal answer is 1.

E. Girl Permutation

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

Some permutation of length n is guessed.

You are given the indices of its prefix maximums and suffix maximums.

Recall that a permutation of length k is an array of size k such that each integer from 1 to k occurs exactly once.

Prefix maximums are the elements that are the maximum on the prefix ending at that element. More formally, the element a_i is a prefix maximum if $a_i > a_j$ for every $j < i$.

Similarly, suffix maximums are defined, the element a_i is a suffix maximum if $a_i > a_j$ for every $j > i$.

You need to output the number of different permutations that could have been guessed.

As this number can be very large, output the answer modulo $10^9 + 7$.

Input

Each test consists of several test cases. The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. Then follows the description of the test cases.

The first line of each test case contains three integers n, m_1 and m_2 ($1 \le m_1, m_2 \le n \le 2 \cdot 10^5$) — the length of the permutation, the number of prefix maximums, and the number of suffix maximums, respectively.

The second line of each test case contains m_1 integers $p_1 < p_2 < \ldots < p_{m_1}$ ($1 \le p_i \le n$) — the indices of the prefix maximums in increasing order.

The third line of each test case contains m_2 integers $s_1 < s_2 < \ldots < s_{m_2}$ ($1 \le s_i \le n$) — the indices of the suffix maximums in increasing order.

It is guaranteed that the sum of the values of n for all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output a single integer on a separate line — the number of suitable permutations modulo $10^9 + 7$.

Standard Input	Standard Output
6 1 1 1 1 1 4 2 3 1 2 2 3 4 3 3 1 1 2 3 3	1 3 1 0 317580808 10

5 3 4 1 2 3 2 3 4 5 20 5 4 1 2 3 4 12 12 13 18 20 6 2 3 1 3 3 4 6	

Note

The following permutations are suitable for the second set of input data:

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

The following permutations are suitable for the sixth set of input data:

- $[2, 1, 6, 5, 3, 4]$
- $[3, 1, 6, 5, 2, 4]$
- $[3, 2, 6, 5, 1, 4]$
- $[4, 1, 6, 5, 2, 3]$
- $[4, 2, 6, 5, 1, 3]$
- $[4, 3, 6, 5, 1, 2]$
- $[5, 1, 6, 4, 2, 3]$
- $[5, 2, 6, 4, 1, 3]$
- $[5, 3, 6, 4, 1, 2]$
- $[5, 4, 6, 3, 1, 2]$

,

F. Nobody is needed

```
Input file:     standard input
Output file:    standard output
Time limit:     6 seconds
Memory limit:   512 megabytes
```

Oleg received a permutation a of length n as a birthday present.

Oleg's friend Nechipor asks Oleg q questions, each question is characterized by two numbers l and r, in response to the question Oleg must say the number of sets of indices $(t_1, t_2, \ldots, t_k)$ of any length $k \geq 1$ such that:

- $l \leq t_i \leq r$ for each i from 1 to k.
- $t_i < t_{i+1}$ for each i from 1 to $k-1$.
- $a_{t_{i+1}}$ is divisible by a_{t_i} for each i from 1 to $k-1$.

Help Oleg and answer all of Nechipor's questions.

Input

Each test consists of several sets of input data. The first line contains a single integer t ($1 \leq t \leq 10^4$) — the number of sets of input data. The description of the sets of input data follows.

The first line of each set of input data contains two integers n and q ($1 \leq n, q \leq 10^6$) — the length of the permutation and the number of Nechipor's questions.

The second line of each set of input data contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq n$) — the permutation a itself.

In each of the next q lines of each set of input data, there are two integers l and r ($1 \leq l \leq r \leq n$) — the next question of Nechipor.

It is guaranteed that the sum of the values of n and the sum of the values of q over all test cases does not exceed 10^6.

Output

For each set of input data, output the answers to all of Nechipor's questions.

Standard Input	Standard Output
4 8 8 2 1 6 3 5 4 8 7 1 8 2 8 1 7 1 6 1 3 5 8 4 4 2 3 1 1 1	20 15 18 12 5 5 1 3 1 2 3 2 27

1 1 3 3 3 2 1 1 2 1 3 2 3 8 1 1 2 3 4 5 6 7 8 1 8	

Note

All suitable arrays in the first set of input data: (1), (2), (3), (4), (5), (6), (7), (8), $(1, 3)$, $(1, 6)$, $(1, 7)$, $(1, 6, 7)$, $(2, 3)$, $(2, 4)$, $(2, 5)$, $(2, 6)$, $(2, 7)$, $(2, 8)$, $(2, 6, 7)$, $(6, 7)$.

All suitable arrays in the fourth set of input data: (1), (2), (3), (4), (5), (6), (7), (8), $(1, 2)$, $(1, 3)$, $(1, 4)$, $(1, 5)$, $(1, 6)$, $(1, 7)$, $(1, 8)$, $(1, 2, 4)$, $(1, 2, 6)$, $(1, 2, 8)$, $(1, 2, 4, 8)$, $(1, 3, 6)$, $(1, 4, 8)$, $(2, 4)$, $(2, 6)$, $(2, 8)$, $(2, 4, 8)$, $(3, 6)$, $(4, 8)$.