

Problem A. Nicolina

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 1024 megabytes



Nicolina is actually sort of a district of Iași, so Little IR12660 still had time to appreciate the Ieșean skyline. However, he can't really figure out whether some buildings belong to Nicolina or Iași, so he naturally comes up with the following problem:

You are given a function $f(a)$ that represents the maximum skyline achievable by permuting the elements of an array a .

The skyline of an array a is defined as:

$$\text{skyline} = \max(\min(a_i, a_{i+1}, \dots, a_j) \cdot (j - i + 1)), \text{ where } 1 \leq i \leq j \leq n$$

Given an array a of size n , partition its elements into two non-empty subsequences X and Y such that: $f(X) + f(Y)$ is maximized.

A subsequence of a is a sequence that can be derived from a by deleting some (or no) elements without changing the order of the remaining elements.

Input

The first line contains an integer t ($1 \leq t \leq 10^4$) — the number of test cases.

The description of each test case is as follows:

- The first line of a test case contains an integer n ($2 \leq n \leq 2 \cdot 10^5$) — the size of the array.
- The second line of a test case contains n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — the elements of the array.

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

Output

For each test case, output a single integer — the maximum value of $f(X) + f(Y)$ that can be achieved by partitioning a into two non-empty subsequences X and Y .

Example

standard input	standard output
2	11
5	80
3 1 4 1 5	
4	
10 20 30 40	

Note

In the first testcase, we choose subsequences $X = [3, 4, 5]$, $Y = [1, 1]$ and achieve a cost of $9 + 2 = 11$.

Problem B. Buhăiești

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



GrandPaPa in Buhăiești

Tired of going by train with the godfather, Little IR12660 decided to go for the next best thing: going with the Grandpapa. He soon started regretting his decision after the Grandpapa started telling him stories of his youth. One of them was how he got to solve the following:

You are given integers n , m , and an array B of length n with values in the range $[0, m]$.

Determine the number of ways to construct an array A of length n , such that:

- $0 \leq A[i] \leq m$
- $\text{mex}(A[1 \dots i]) \leq \text{mex}(B[1 \dots i])$ for every $1 \leq i \leq n$.
- A is lexicographically smaller than B .

Here, mex (minimum excluded value) of a set of integers is defined as the smallest non-negative integer that is not present in the set. For example:

- $\text{mex}(\{0, 1, 3\}) = 2$
- $\text{mex}(\{1, 2, 3\}) = 0$

The notation $\text{mex}(A[1 \dots i])$ refers to the mex of the elements $A[1], A[2], \dots, A[i]$.

Little IR12660 didn't pay much attention to Grandpapa's solution, so now he has to make his own. But of course, he will ask you to solve it instead of trying on his own powers.

Input

The first line contains a single integer t ($1 \leq t \leq 10$) — the number of test cases.

Each test case consists of:

- A single line containing two integers n and m ($1 \leq n, m \leq 50$).
- A second line containing n integers — the array B ($0 \leq B[i] \leq m$).

It is guaranteed that the sum of n does not exceed 80 and the sum of m does not exceed 50.

Output

For each test case, output a single integer — the number of valid arrays A satisfying the conditions modulo $10^9 + 7$.

Example

standard input	standard output
8	5
3 2	11
2 1 2	0
4 3	1089
1 0 2 3	62
2 1	52856
1 1	0
5 4	3
3 0 1 2 4	
6 2	
1 2 1 0 2 2	
9 3	
2 1 0 3 2 1 3 2 0	
1 0	
0	
2 2	
2 2	

Note

For the first testcase, the possible arrays A are:

- $[1, 2, 2]$
- $[1, 1, 2]$
- $[1, 1, 1]$
- $[1, 2, 1]$
- $[2, 1, 1]$

Problem C. Vaslui

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



Little IR12660 visited a school in Vaslui. He went to talk with the principal of the school to get his insight into education. The principal told him that there is only one thing important in the Vasluian culture: the social endurance. And the only purpose of the school is to instill this endurance into the students. However, school is too short to actually help all students, so the principal is happy if the school managed something close enough. More formally:

You are given a binary string s of length n . In one operation, you can flip a character (change 0 to 1 or 1 to 0).

Your task is to determine the minimum number of operations needed to ensure that, for any contiguous subarray of length greater than or equal to k , the character 1 is the majority element (it appears at least as many times as 0 does).

Input

The first line contains a single integer t ($1 \leq t \leq 10^4$) — the number of test cases.

Each test case consists of:

- A single line containing two integers n and k ($1 \leq k \leq n \leq 2 \cdot 10^5$) — the length of the binary string and the minimum length of subarrays to consider.
- The second line contains a binary string s of length n .

It is guaranteed that the total length of all strings s across 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.

Example

standard input	standard output
5	2
5 2	0
00010	4
12 4	2
110111101110	2
4 1	
0000	
5 3	
11000	
9 2	
110101010	

Note

In the first testcase, to ensure that 1 is the majority in all subarrays of length ≥ 2 , you can perform 2 flips:

- Flip the second character to get 01010.
- Flip the third character to get 01110.

The result is a string where 1 is the majority in all subarrays of length ≥ 2 .

In the second testcase, no flips are required because 1 is already the majority for all subarrays of length ≥ 4 .

Problem D. Crasna

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



There are a total of 426 inhabitants in Crasna, according to the 2021 census. This brings great sadness to Little IR12660, who wants this village to flourish. So he planted a tree there.

The tree Little IR12660 planted is a very special one that will grow to have N nodes. Little IR12660 knows that each node will have an initial label and a *friendship* label. All initial labels are pairwise distinct values from 1 to N , and all friendship labels are pairwise distinct values from 0 to $N - 1$. Little IR12660 also knows how the nodes will be connected among them according to the initial labels. However, he doesn't know anything about the friendship labels, as they develop according to many unpredictable environmental factors.

Little IR12660 doesn't have time to watch his tree actually grow. So he wonders in how many ways can the friendship labels be associated to the nodes such that the sum of the MEX[‡] of all the paths in the tree is minimized.

[‡]: The MEX of a set is the smallest natural number that does not belong to the set.

Input

The first line contains a single integer T ($1 \leq T \leq 30\,000$), the number of test cases. For each test case:

The first line contains one integer N ($1 \leq N \leq 200\,000$), the number of nodes of the tree.

Then, $N - 1$ lines follow, each with two integers, x_i and y_i ($1 \leq x_i, y_i \leq N, x_i \neq y_i$), denoting there is an edge between the node with initial label x_i and the one with initial label y_i .

It is guaranteed that the sum of N over all testcases is at most 200 000.

Output

For each testcase, print the answer modulo 998 244 353.

Example

standard input	standard output
3	34560
9	5040
6 8	2
4 8	
2 9	
5 7	
1 9	
4 7	
7 3	
7 1	
8	
2 7	
3 6	
6 8	
4 3	
6 1	
2 4	
6 5	
3	
1 2	
3 1	

Note

For the third testcase, the two ways the friendship labels can be associated to the nodes with initial labels $[1, 2, 3]$ are $[0, 2, 1]$ and $[1, 2, 0]$.

Problem E. Bârlad

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes



Bârlad is the home of some rocky landscapes and very bad drivers. The train passed Bârlad way too fast for Little IR12660 to actually visit it. So now he can only imagine what Bârlad might look like. He knows Bârlad is a city stretching along a single line, and he also knows that if the landscape across this line gets too complicated for the locals, they might crash. So, he formulates the following problem:

Count the number of permutations of order N that satisfy M restrictions of the following types:

- The subarray from l to r is a **valley**.
- The subarray from l to r is a **mountain**.

An array is considered:

- A **valley** if it strictly decreases to a minimum point and then strictly increases. For example, $[5, 3, 1, 4, 6]$ is a valley.
- A **mountain** if it strictly increases to a maximum point and then strictly decreases. For example, $[1, 3, 5, 4, 2]$ is a mountain.

A strictly increasing or strictly decreasing array is considered neither a valley nor a mountain.

Input

The first line contains a single integer t ($1 \leq t \leq 500$) — the number of test cases.

The description of each test case is as follows:

- The first line of each test case contains two integers N ($3 \leq N \leq 2000$) and M ($0 \leq M \leq 2000$) — the size of the permutation and the number of restrictions.
- The next M lines each contain three integers l , r , and $type$ ($1 \leq l, l+1 < r \leq N$, $1 \leq type \leq 2$):
 - l and r specify the range of the subarray.
 - $type = 1$ means the subarray must be a valley.
 - $type = 2$ means the subarray must be a mountain.

It is guaranteed that the sum of N across all test cases does not exceed 2000 and the sum of M across all test cases does not exceed 2000.

Output

For each test case, output a single integer — the number of permutations of size n that satisfy all m restrictions modulo $10^9 + 7$.

Example

standard input	standard output
5	20
5 2	6
1 3 1	0
2 5 2	0
4 1	120
1 4 1	
6 3	
1 4 2	
2 5 1	
3 6 2	
3 2	
1 3 1	
1 3 2	
5 0	

Problem F. Tecuci

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 512 megabytes



Little IR12660 arrived in Tecuci, land of legendary mustard sauce. He is currently at a mustard fair, where various mustard sauces are being presented. While the mustard sauces were spinning around on a conveyor belt, he came up with the following idea:

You are given an array A of length n and an integer k . You are allowed to perform at most k cyclic shifts on any subarray of A .

Determine the maximum sum of a contiguous subarray that can be obtained after performing at most k such cyclic shifts.

A **cyclic shift** in any direction on a subarray means:

- For a **right cyclic shift**, the last element of the subarray is moved to the front, and all other elements are shifted one position to the right.
 - Example: For $A = [1, 2, 3, 4, 5]$, a right cyclic shift on the subarray $[2, 3, 4]$ (indices 2 to 4) results in $A = [1, 4, 2, 3, 5]$.
- For a **left cyclic shift**, the first element of the subarray is moved to the back, and all other elements are shifted one position to the left.
 - Example: For $A = [1, 2, 3, 4, 5]$, a left cyclic shift on the subarray $[2, 3, 4]$ (indices 2 to 4) results in $A = [1, 3, 4, 2, 5]$.

Input

The first line contains a single integer t ($1 \leq t \leq 10^4$) — the number of test cases.

Each test case consists of:

- A single line containing two integers n and k ($1 \leq n \leq 10^6, k \geq 0, n \cdot k \leq 10^6$) — the size of the array and the maximum number of allowed cyclic shifts.

- A second line containing n integers a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$) — the elements of the array.

It is guaranteed that the sum of $n \cdot k$ across all testcases does not exceed 10^6 and the sum of n across all testcases does not exceed 10^6 .

Output

For each test case, output a single integer — the maximum sum of a contiguous subarray that can be obtained after performing at most k cyclic shifts.

Example

standard input	standard output
4	6
5 2	10
-1 2 -3 4 -5	2
7 3	-1
4 -2 1 3 -1 -3 2	
6 1	
-1 -1 2 -1 -1 -1	
5 5	
-1 -1 -1 -1 -1	

Problem G. Focşani

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



Little IR12660 doesn't know that much about Focşani, apart from the fact that it is the closest he will ever come to Transylvania. So he just remembers a problem he got as homework at school:

You are given a tree with N nodes, where each node i has an initial label a_i , an integer in the range $[1, k]$. A tree is an undirected and connected graph, with $N - 1$ edges.

Each label belongs to an alphabet of size K . Changing the label of the i -th node to x incurs a cost $C_{i,x}$.

A path in the tree is a set of nodes (x_1, x_2, \dots, x_m) such that x_i is connected to x_{i+1} for all $1 \leq i < m$. A path is **palindromic** if the sequence of labels along the path forms a palindrome.

Your task is to determine the minimum cost required to relabel the tree such that no path of length greater than 1 in the tree is palindromic.

Input

The first line contains two integers N and K ($2 \leq N \leq 1000$, $1 \leq K \leq 20$) — the number of nodes and the size of the alphabet.

The second line contains N integers a_1, a_2, \dots, a_N ($1 \leq a_i \leq K$) — the initial labels of the nodes.

The next $N - 1$ lines each contain two integers u and v ($1 \leq u, v \leq N$), representing an edge between nodes u and v .

The following N lines each contain K integers. The j -th integer in the i -th line represents $C_{i,j}$ ($0 \leq C_{i,j} \leq 10^9$) — the cost of changing node i to label j .

The input for this problem is generated uniformly at random! (except N and K)

Output

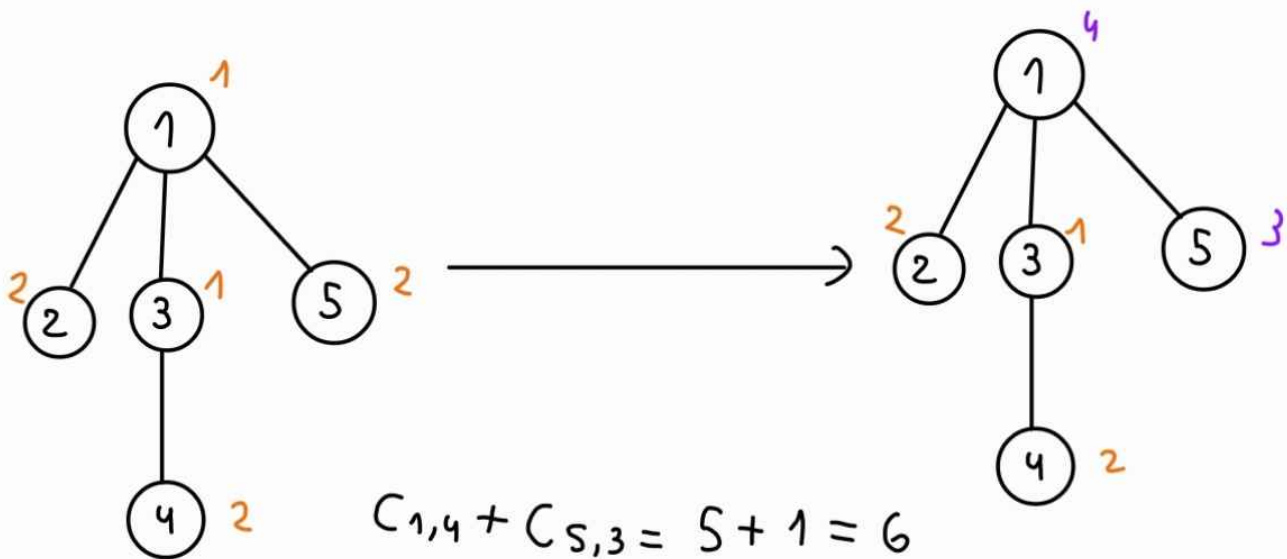
Output a single integer — the minimum cost required to relabel the tree such that no path of length greater than 1 is palindromic.

If there is no solution, output **-1**.

Examples

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

Note



Analysis of the example.

Problem H. Râmnicu Sărat

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



After reaching Râmnicu Sărat, Little IR12660 felt the weary aura the city has after being the battleground of multiple crucial historic battles.

Little IR12660 heard a very interesting rumor about Râmnicu Sărat; more specifically, he heard that exactly N different rulers controlled this city during the middle ages. Moreover, while some were ruled centuries apart from each other, a constant seemed to foretell this statistic: if we were to take the list of casualties in all the battles fought in Râmnicu Sărat, the lowest common multiple of any two numbers in this list was equal to N .

However, this list of casualties has been long lost in history. The number N hasn't. Little IR12660 wonders what the maximum size of this list could be and what one such maximal size list can look like. Of course, he then asks you to find this maximum size and one list of this size.

Input

The first line contains a single integer T ($1 \leq T \leq 100$) — the number of test cases.

Each of the next T lines contains a single integer N ($1 \leq N \leq 10^{12}$).

Output

For each test case, output an integer K — the maximum size of a list satisfying the given condition.

On the following line, output K integers that respect the given propriety.

If there are multiple solutions, output any of them.

Example

standard input	standard output
4	1
1	1
2	2
9	1 2
12	2
	1 9
	3
	3 4 12

Problem I. Buzău

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



Little IR12660 is gone meeting with a friend flying in from Nice. He has a lot of catching up to do over a good cup of joe and pretzels, so he is unavailable to solve the following problem. Please, do it in his stead:

You are given a tree with N ($1 \leq N \leq 100\,000$) nodes and $N - 1$ edges. Each node is attributed a color, which we will conveniently label with numbers from 1 to K ($1 \leq K \leq N$). We define a chain as a set of nodes that lie on the minimum length path between two nodes. We call these two nodes the endpoints of the chain. Then, we define a **correct chain arrangement** as a set of chains such that:

- For every chain, the endpoints of it have the same color,
- For every color c ($1 \leq c \leq k$) there is **exactly one** chain such that the color of its endpoints is c .

For such a correct chain arrangement, let it be called A , we define d_u as the number of chains in A of which node u is part of. More formally, $d_u = \text{card}(\{C \mid C \subset A, u \in C\})$, where $\text{card}(S)$ denotes the number of elements in that set.

Determine the lexicographically maximal array $A_V = [d_1, d_2, \dots, d_N]$ such that there exists a correct chain arrangement A that yields it.

Input

You are given K ($1 \leq K \leq N$) and N ($1 \leq N \leq 100\,000$).

On the next line, you are given N numbers, the i -th of which denotes c_i ($1 \leq c_i \leq N$), the color of the i -th node. Within the tests used, every color has at least one node that contains it.

Each of the next $N - 1$ lines contains two integers u and v ($1 \leq u, v \leq N, u \neq v$), meaning that there is an edge between u and v in the tree.

Output

Print N numbers, forming the lexicographically maximal array that could be generated by a correct chain arrangement

Example

standard input	standard output
3 7	2 1 2 1 2 0 0
1 1 3 2 2 1 2	
5 1	
5 3	
1 4	
1 7	
3 6	
3 2	

Note

The chains used in the first example are as follows:

- For color 1, the chain with endpoints in 1 and 2
- For color 2, the chain with endpoints in 4 and 5
- For color 3, the chain containing only node 3

Problem J. Mizil

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



Anisie, after suffering a series of successive disappointments at her former workplace, decides to escape the urbanism of Bucharest for that of Mizil. Mizil is kind of like New York in Romania. There, she finds a job as a tour guide.

Mizil can be described as a graph with N tourist points and M bidirectional streets connecting these points. Each street can be described by the points x_i, y_i that it connects and a *congestion* value c_i , which indicates how unpleasant it is to traverse completely due to various factors, one of them being the presence of traffic jams. It is guaranteed that any two tourist points have at most one street connecting them and that it is possible to reach any tourist point from any other by traversing the given streets.

Anisie organizes tours through Mizil. A tour is defined as a sequence of tourist points t_1, t_2, \dots, t_k , where any two successive tourist points are connected by a street. When planning these tours, two parameters must be taken into account:

- The level of **discomfort**, which is the highest *congestion* value of any street from t_i to t_{i+1} traversed during the tour.
- The level of **luminosity**. For a tourist point, we define its *neon* coefficient as the number of streets originating from it. The level of luminosity will be the smallest *neon* coefficient of any tourist point in the tour.

Ideally, Anisie would minimize the discomfort and maximize the luminosity. However, since each tourist has different needs, she will have to determine, given Mizil's map, for Q triplets u_i, v_i, l_i , what is the **minimum discomfort** possible for a tour that starts at u_i , ends at v_i , and has a luminosity level of **at least** l_i . If no such tour exists, the result will be -1 .

Input

On the first line, the numbers N , M , and Q ($1 \leq N, Q \leq 100\,000, 1 \leq M \leq 300\,000$) will be present, representing, in order, the number of tourist points in Mizil, the number of streets connecting these points, and the number of queries.

Next, M lines follow, each containing 3 integers, x_i, y_i, c_i ($1 \leq x_i, y_i \leq N, x_i \neq y_i, 1 \leq c_i \leq M$), representing a street from x_i to y_i with a *congestion* value of c_i .

Then, Q lines follow, each containing 3 integers, u_i, v_i, l_i ($1 \leq u_i, v_i \leq N, u_i \neq v_i, 1 \leq l_i \leq N$), representing a query for the **minimum** possible discomfort level of a tour that starts at u_i , ends at v_i , and has a luminosity level of at least l_i .

Output

There will be Q lines displayed, each containing a single number, with the i -th one representing the answer to the i -th query.

Example

standard input	standard output
6 9	9
1 3 4	6
1 5 8	3
1 6 5	3
2 5 3	5
2 6 3	2
3 4 2	
3 5 6	
5 6 9	
4 6 2	
6	
5 6 4	
5 6 3	
5 6 2	
2 4 2	
3 6 3	
3 6 2	

Note



Train filled to the brim with American tourists lining up from Căpălnița to visit the magical city of Mizil.

Problem K. Ploiești Sud

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



After his long journey through the Moldovan landscape, Little IR12660 arrived at one of his last destinations: Ploiești Sud, a city renowned for its booming petrol industry and coincidentally booming pollution index. As soon as he got here, he realized the city is in dire need of modernization, i.e. dismantling the entire illumination system:

Ploiești Sud can be described as a (connected) tree with N nodes, each node representing an important landmark. Each landmark comes with its own streetlight which illuminates it. A *lighting plan* is a set of streetlights S which the city administration plans to turn on.

To actually turn them on, they have to pay an electricity company to send electricity to them. However, the electricity company has a trick up their sleeve; they don't have to manually power up all the streetlights in the lighting plan: if there is a streetlight v currently powered up, and there is a streetlight u in the lighting plan which is adjacent to v , u can draw current from v for free. As such, the electricity company will choose a subset $T \subset S$ of minimum cardinality such that after manually powering on all the streetlights in T , all other streetlights in S/T can draw current directly or indirectly from those in T . The cardinality of T is called the *cost* of a *lighting plan*.

Little IR12660 has friends on the city council that told him that the mayor only accepts suggestions for lighting plans whose cost is exactly K . Because Little IR12660 is a big fan of Petrolul Ploiești, he wants to propose a *lighting plan* that will pass the mayor that is also the **smallest lexicographic**[‡], to show that Petrolul is number one.

Ploiești Sud is, however, too big for Little IR12660 to handle, so he asks you to find the minimum lexicographic lighting plan in his stead.

[‡] : We call a set A smaller lexicographically than B iff $A \neq B$ and there exists an N_0 such that for all $x \in A \cup B$ with $x < N_0$, $x \in A \cap B$, and $N_0 \in A$ and $N_0 \notin B$, while there is a $y > N_0$ such that $y \in B$.

Input

The first line contains a single integer T ($1 \leq T \leq 30\,000$), the number of test cases. For each test case:

The first line contains two integers N and K ($1 \leq K \leq N \leq 300\,000$), denoting the count of streetlights and the mayor's condition for the cost of the lighting plan.

Then, $N - 1$ lines follow, each with two integers, x_i and y_i ($1 \leq x_i, y_i \leq N, x_i \neq y_i$), denoting there is an edge between streetlight x_i and y_i .

It is guaranteed that the sum of N over all testcases is at most 300 000.

Output

For each testcase, print a string of N bits, for which the i -th bit should be equal to 1 if the i -th streetlight should be included in the lighting plan, or 0 if not.

Example

standard input	standard output
1	000000110110110
15 6	
13 15	
2 5	
14 15	
4 2	
13 12	
10 9	
2 1	
4 7	
12 8	
12 10	
6 8	
3 8	
11 12	
8 5	