Problem A. Almost Acyclic

Input file:	`standard input`
Output file:	`standard output`
Time limit:	10 seconds
Memory limit:	512 megabytes

We call a **connected** undirected graph *almost-acyclic*, if the graph has no cycles, or all the simple cycles in it share at least one common point.

You are given a complete undirected graph $G = (V, E)$ with n vertices. Each edge (i, j) has a weight $w_{i,j}$. Calculate ($f(G)$ is 1 if G is almost-acyclic, or 0 otherwise):

$$\sum_{E' \subseteq E,\ G'=(V,E')} f(G') \prod_{(i,j)\in E'} w_{i,j} \mod 10^9+7$$

Input

The first line contains a single integer T ($1 \le T \le 16$), denoting the number of test cases.

For each test case, the first line contains an integer n ($1 \le n \le 16$).

The next n lines each contains n integers. The j-th number in the i-th line denotes $w_{i,j}$ ($0 \le w_{i,j} < 10^9+7$).

It is guaranteed that $w_{i,j} = w_{j,i}$, $w_{i,i} = 0$.

It is guaranteed that for each $1 \le i \le 16$, there is at most one test case satisfying $n = i$.

Output

For each test case, output one line with an integer denoting the answer.

Example

standard input	standard output
2 3 0 1 2 1 0 1 2 1 0 5 0 1 0 1 1 1 0 1 1 1 0 1 0 1 0 1 1 1 0 1 1 1 0 1 0	7 120

Problem B. Assignment

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

You are given two sequences a, b of length n and a cost matrix A of size $n \times n$. **The matrix A satisfies $A_{i,j} \ge A_{i,j-1}$ for all** $1 \le i \le n, 1 < j \le n$. You can do the following operation arbitrary number of times:

- Select three integers l, r, x satisfying $1 \le l \le r \le n$ and $1 \le x \le n$, then assign x to a_i for all indices i between l and r, inclusive. The cost of this operation is $A_{x,r-l+1}$.

For all $i \in [0, k]$, find the minimum sum of costs to make a has at most i positions differing from b.

Input

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

For each test case, the first line contains two integers n, k ($1 \le n \le 100$, $1 \le k \le 10$).

The second line contains n integers $a_1, a_2, \cdots, a_n$ ($1 \le a_i \le n$), denoting the sequence a.

The third line contains n integers $b_1, b_2, \cdots, b_n$ ($1 \le b_i \le n$), denoting the sequence b.

The next n lines, each contains n integers. The j-th integer in the i-th line denotes $A_{i,j}$ ($1 \le A_{i,j} \le 10^6$).

It is guaranteed that for all $1 \le i \le n$, $1 < j \le n$, $A_{i,j} \ge A_{i,j-1}$.

Output

For each test case, output one line with $k + 1$ integers denoting the answer.

Example

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

Problem C. Many Topological Problems

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

Once you created the following problem:

Topological Problem

You are given a labeled rooted tree with n vertices and an integer k. We call a permutation a of length n good if $a_i > a_{par_i}$ and $a_i \leq a_{par_i} + k$ for each i with a parent par_i.

Find the number of good permutations.

Now, thinking this problem is too easy, you create the following problem:

Many Topological Problems

You are given two integers n, k. For all different labeled rooted trees T with n vertices, find the sum of the answer to the *Topological Problem* of T, modulo $10^9 + 7$.

Please solve **Many Topological Problems**.

Two labeled rooted trees are considered different, if and only if their roots differ, or one edge exists in one tree but not in the other.

Input

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

For each test case, the only line contains two integers n, k $(1 \leq k \leq n \leq 10^6)$.

Output

For each test case, output one line with an integer denoting the answer.

Example

standard input	standard output
3 2 2 5 1 114514 1919	2 120 354463397

Problem D. Do You Like Interactive Problems?

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

There is an integer x satisfying $1 \le x \le n$. You know n but you don't know x.

You can do the following guessing: pick an random integer y uniformly satisfying $1 \le y \le n$ (your y may equal to previous queries), and you will be told if $x < y$, $x > y$ or $x = y$. You will stop asking if there is only one possible x satisfying all the conditions.

Given n, if x is picked randomly uniformly, what's your expected number of queries?

Input

The first line contains an integer T ($1 \le T \le 100$) denoting the number of test cases.

For each test case, the only line contains an integer n ($1 \le n \le 10^9$).

Output

For each test case, output one integer denoting the expected number of queries modulo 998244353.

Formally, it can be proven that the sought expected value can be represented as an irreducible fraction p/q which satisfies $q \not\equiv 0 \bmod 998244353$, and there is a unique integer r satisfies $0 \le r < 998244353$ and $qr \equiv p \bmod 998244353$. Find this r.

Example

standard input	standard output
2 1 2	0 1

Problem E. Equivalence

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

You are given two trees T_1, T_2, both with n vertices. The lengths of edges of T_1 are given. **The length of each edge is non-negative.**

A tree T with n vertices is good, if there is a way to assign each edge on T_2 with a length which satisfies the following condition:

- For each pair i, j satisfying $1 \le i < j \le n$, the distances of i and j on T and T_2 are the same.

You can perform the following operation on T_1: select an edge on T_1 and replace its length with any **non-negative** integer.

Find the minimum number of operations to make T_1 good.

Input

The first line of input contains a single integer T ($1 \le T \le 8600$), denoting the number of test cases.

For each test case, the first line contains one integer n ($2 \le n \le 10^6$).

The second line contains $n-1$ integers $p_2, p_3, \cdots, p_n$ ($1 \le p_i \le n$).

The third line contains $n-1$ integers $val_2, val_3, \cdots, val_n$ ($0 \le val_i \le 10^9$).

These two lines denotes $n-1$ edges (u, p_u) with weight val_u on T_1.

The fourth line contains $n-1$ integers $p'_2, p'_3, \cdots, p'_n$ ($1 \le p'_i \le n$), denoting $n-1$ edges (u, p'_u) on T_2.

It is guaranteed that $\sum n \le 1.1 \cdot 10^6$.

Output

For each test case, the only line contains one integer denoting the answer.

Example

standard input	standard output
1 5 1 5 2 2 0 2 3 1 5 5 5 1	1

Problem F. Fences

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

A village consists of n buildings. Each building can be represented as a point on a two-dimensional plane. The coordinates of building i are (x_i, y_i).

The villagers want to put up fences around the village with the following requirements:

- Fences must form a simple polygon;
- Every building is in the polygon (including borders);
- Building k, as the entrance of the village, must lie on the fences.

Find the minimum total length of fences required to form a valid polygon.

Input

The first line contains a single integer T ($1 \le T \le 10^4$), denoting the number of test cases.

For each test case, the first line contains two integers n, k ($3 \le n \le 2 \times 10^5$, $1 \le k \le n$), denoting the number of buildings and the entrance.

Each of the following n lines contains two integers x_i, y_i ($|x_i|, |y_i| \le 10^6$), denoting the coordinates of building i.

For each test case, it is guaranteed that the given points are distinct, and there are at least three points that are not collinear.

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

Output

For each test case, output one real number in a single line denoting the minimum total length of fences. (rounding to 3 decimal places)

Example

standard input	standard output
1 5 3 0 0 0 2 1 1 2 0 2 2	8.828

Problem G. Make 2

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

For a sequence a consisting of n positive integers, you can perform the following operation several times:

- Choose an index i which satisfies $1 < i < n$ and $a_i > 1$, then decrease a_i by 1, and add 1 to a_{i-1} and a_{i+1}.

A sequence consisting of n positive integers is considered good if it is possible to make $a_i = 2$ for each $1 \le i \le n$, by using several (possibly, zero) such operations.

Now you need to calculate the number of good sequences that satisfy m constraints, the i-th constraint can be represented as a pair (x_i, y_i) which requires $a_{x_i} = y_i$.

It can be proven that the answer is finite. Output the answer modulo $10^9 + 7$.

Input

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

For each test case, the first line contains two integers n, m ($1 \le n \le 10^{18}$, $0 \le m \le \min(n, 100)$).

The next m lines each contains two integers. The i-th line contains x_i, y_i ($1 \le x_1 < x_2 < \cdots < x_m \le n$, $1 \le y_i \le 10^9$).

Output

For each test case, output one line with an integer denoting the answer modulo $10^9 + 7$.

Example

standard input	standard output
3 3 1 2 2 5 2 1 2 5 1 114514 0	1 2 158552999

Problem H. XOR Subsequence

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

Alice used to have a sequence $a_1, \cdots, a_n$, but she has forgotten about it now. Fortunately, she noticed that she had calculated the XOR sum for each non-empty subsequence of the sequence and obtained $2^n - 1$ results, but their order was disrupted.

Now she hopes you can help restore the sequence. If there are multiple possible sequences, please tell her the sequence with the **smallest lexicographical order**, or report there is no correct sequence.

Input

The first line contains a single integer T $(1 \leq T \leq 5000)$, denoting the number of test cases.

For each test case, the first line contains an integer n $(1 \leq n \leq 18)$.

The next line contains $2^n - 1$ non-negative integers strictly less than 2^{30}, denoting the results.

It is guaranteed that the sum of 2^n over all test cases does not exceed 2^{20}.

Output

For each test case, output one line. If there is no correct sequence, output -1; otherwise, output n integers denoting the answer.

Example

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

Problem I. Far Away from Home

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

You have decided to move your house to a new place, by a straight road. There are n stores lying on the road. The i-th store has a distance x_i to the leftmost of the road.

There are c types of groceries you need to buy. For each type, your cost to buy it is the distance between your house and the nearest store which sells this. Your total cost is the sum of costs of each type.

Note that even you may buy some types of groceries in the same store, you still need to calculate the distance multiple times.

You need to choose a place for your house to minimize the total cost.

Input

Each test contains multiple test cases. The first line contains an integer T ($1 \le T \le 5$) denoting the number of test cases.

For each test case, the first line contains two integers n, c ($1 \le n \le 10^5$, $1 \le c \le 5 \cdot 10^5$).

The next n lines each, contains two integers x_i and t_i first ($1 \le x_i \le 10^9$, $t_i \ge 1$), denoting the coordinate of store i and the number of types of groceries store i sells, and following t_i distinct integers $a_{i,1}, a_{i,2}, \cdots, a_{i,t_i}$ ($1 \le a_{i,j} \le c$), denoting the types of groceries store i sells. It is guaranteed that $1, 2, \cdots, c$ each appears at least once in all the types the stores sell.

For each test case, it is guaranteed that $\sum t_i \le 5 \cdot 10^5$.

Output

For each test case, output one line with an integer, denoting the minimum total cost.

Example

standard input	standard output
1 4 4 1 1 4 5 1 4 9 3 1 3 4 2 2 2 3	7

Problem J. Border Queries

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

Given a string S of length n consisting of lowercase English letters. A partition of S into three non-empty substrings s_1, s_2, s_3 is considered good if and only if s_1 is the border of $s_1 + s_2$ and s_3 is the border of $s_2 + s_3$. We say a string s good if and only if s is a substring of S and there exists a good partition of S into s_1, s_2, s_3 such that $s_2 = s$.

Define the value of a string as the number of its good substrings. Two substrings are considered different if and only if the start position is different or the end position is different.

Given a string T of length m consisting of lowercase English letters and q queries. In each query, you are given two integers l, r. You need to calculate the value of $T[l \cdots r]$.

Input

Each test contains multiple test cases. The first line contains an integer T ($1 \le T \le 60$) denoting the number of test cases.

For each test case, the first line contains three integers n, m, q ($3 \le n \le 10^6$, $1 \le m, q \le 10^6$).

The second line contains a string S of length n.

The third line contains a string T of length m.

The next q lines each contains two integers l_i and r_i, denoting a query ($1 \le l_i \le r_i \le m$).

It is guaranteed that $\sum n, \sum m, \sum q$ over all test cases does not exceed 10^6.

Output

For each query, output one line with an integer denoting the answer.

Please do not output trailing spaces.

Example

standard input	standard output
1 7 7 2 abacaba cabacab 1 4 3 7	0 2

Problem K. Werewolves

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

There are n players sitting in a row and m kinds of identity cards. The players are numbered from $1 \sim n$. The number is public, which means everyone knows the number of each other.

A moderator will give each player an identity card. However, the receiver isn't allowed to view their identity.

Everybody will shut their eyes. Then the moderator will call out each player in turn. All other players' identity cards, disordered, will be shown to that player. The player should guess their identity and shut their eyes afterward. All other players will remain their eyes closed during the procedure.

The players have enough time to discuss before the game starts and want to make sure that at least $\lfloor \frac{n}{m} \rfloor$ of the guesses are correct. Please help them make a strategy.

Input

The first line contains an integer T, denoting the number of testcases.

Each testcase contains two integers n, m, separated by a space.

The input guarantees that $2 \le m \le n, m^n \le 2.1 \times 10^6, \sum m^n \le 1.4 \times 10^7$.

Output

For each testcase, output n lines, line p denoting the strategy of player p.

Denote a sequence s valid if and only if s is a non-descending sequence of length $n-1$ and contains integers in $[1, m]$. Denote the count of valid sequence c, then output c integers between 1 and m, the k-th integer representing what the player will guess when the multiset of identity cards seen is equal to the multiset of the k-th valid sequence sorted in lexicographical order.

Example

standard input	standard output
1 2 2	1 2 2 1

Problem L. Equalize the Array

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

You are given an array a consisting of n integers.

In one move, you can choose a positive integer x, such that x is one of the modes of the array, then add 1 to each x in a.

An integer x is a mode of an array a if and only if x appears most frequently in a. Note that an array may have multiple modes (e.g. $2, 3$ are both the modes of $[2, 2, 1, 3, 3]$).

Find out if it is possible to get an array that all elements in it are equal through several (possibly zero) such moves.

Input

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

For each test case, the first line contains an integer n ($1 \le n \le 10^6$).

The next line contains n integers. The i-th number denotes a_i ($1 \le a_i \le n$).

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^6$.

Output

For each test case, output a string. If it is possible, output `YES`; otherwise, output `NO`.

Example

standard input	standard output
3 5 1 2 3 4 5 5 4 4 1 4 4 4 2 2 2 2	YES NO YES