Problem A. Sequence

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

ZZX has a sequence of boxes numbered $1, 2, \ldots, n$. Each box can contain at most one ball.

You are given the initial configuration of the balls. For $1 \le i \le n$, if the i-th box is empty, then $a_i = 0$, otherwise the i-th box contains exactly one ball, the color of which is a_i, a positive integer. Balls of the same color cannot be distinguished.

ZZX will perform m operations in order. During i-th operation, he collects all the balls from boxes $l_i, l_i + 1, \ldots, r_i - 1, r_i$, and then arbitrarily puts them back into these boxes. Note that each box should always contain at most one ball.

ZZX wants to change the configuration of the balls from $a_1, a_2, \ldots, a_n$ to $b_1, b_2, \ldots, b_n$ using these operations. Please tell ZZX whether it is possible to achieve his goal.

Input

The first line contains an integer $T \le 60$. Then T test cases follow. In each test case:

The first line of the test case contains two integers n and m ($1 \le n \le 1000$, $0 \le m \le 1000$, sum of n over all test cases does not exceed 2000, sum of m over all test cases does not exceed 2000).

The second line contains $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le n$). The third line contains $b_1, b_2, \ldots, b_n$ ($0 \le b_i \le n$). Each of the next m lines contains two integers l_i and r_i ($1 \le l_i \le r_i \le n$).

Output

For each test case, print "`Yes`" or "`No`" on a separate line.

Example

standard input	standard output
5 4 1 0 0 1 1 0 1 1 1 1 4 4 1 0 0 1 1 0 0 2 2 1 4 4 2 1 0 0 0 0 0 0 1 1 3 3 4 4 2 1 0 0 0 0 0 0 1 3 4 1 3 5 2 1 1 2 2 0 2 2 1 1 0 1 3 2 4	No No Yes No Yes

Problem B. Colored Graphs

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

You are given a directed graph which is constructed as follows:

- Pick a connected undirected graph with exactly n vertices and n edges. The vertices are numbered 1 through n.
- Convert each undirected edge into a directed edge in such a way that each vertex has outdegree 1.

Additionally, you are given m different colors to color the vertices. Your task is to calculate the number of different colored graphs that can be made.

Two colored graphs A and B are considered the same if and only if there exists a mapping P between their sets of vertices which satisfies the following constraints:

- Vertex u in graph A has the same color as vertex $P(u)$ in graph B.
- For any two different vertices u and v in graph A, $P(u)$ and $P(v)$ are different vertices in graph B.
- For any directed edge $u \to v$ in graph A, there exists a corresponding directed edge $P(u) \to P(v)$ in graph B.

Print the answer modulo $10^9 + 7$.

Input

The first line of the input contains two space-seperated integers n and m ($3 \le n \le 10^5$, $1 \le m \le 10^9$), representing the number of vertices in the graph and the number of colors you have.

Then, n lines follow. The i-th of them contains an integer f_i ($1 \le f_i \le n$, $f_i \ne i$), denoting a directed edge from vertex i to vertex f_i in the given graph.

Output

Print a single line containing the answer.

Example

standard input	standard output
6 3 2 3 4 1 1 3	378

Problem C. Graph Coloring 2

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

You are given an undirected graph with n vertices numbered 0 through $n-1$. Obviously, the set of vertices have $2^n - 1$ non-empty subsets. For a non-empty subset S, a proper coloring of S is a way to assign each vertex in S a color, so that no two vertices in S with the same color are directly connected by an edge. Assume we used k different kinds of colors in a proper coloring. The *chromatic number* of subset S is the minimum possible k among all the proper colorings of S.

Now your task is to compute the chromatic number of every non-empty subset of n vertices.

Input

The first line contains an integer T. Then T test cases follow.

The first line of each test case contains an integer n. Each of then next n lines contains a string consisting of '`0`' and '`1`'. For $0 \le i \le n-1$ and $0 \le j \le n-1$, if the j-th character of the i-th line is '`1`', then vertices i and j are directly connected by an edge, otherwise they are not directly connected.

The i-th character of the i-th line is always '`0`'. The i-th character of the j-th line is always the same as the j-th character of the i-th line.

For all test cases, $1 \le n \le 18$. There are no more than 100 test cases with $1 \le n \le 10$, no more than 3 test cases with $11 \le n \le 15$, and no more than 2 test cases with $16 \le n \le 18$.

Output

For each test case, print an integer on a separate line. This integer is determined as follows: We define the identity number of subset S as $id(S) = \sum_{v \in S} 2^v$. Let the chromatic number of S be $f_{id(S)}$. You need to output

$$\left(\sum_{id(S)=1}^{2^n-1} f_{id(S)} \cdot 233^{id(S)} \right) \bmod 2^{32}.$$

Example

standard input	standard output
2 4 0110 1010 1101 0010 4 0111 1010 1101 1010	1022423354 2538351020

Note

For the first test case, $ans[1..15] = \{1, 1, 2, 1, 2, 2, 3, 1, 1, 1, 2, 2, 2, 2, 3\}$.

Problem D. Simple Graph

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

An undirected simple graph G can be divided into connected components. Let x be the number of trees among these components. Then the value of graph G is defined as x^k.

Given n and k, your task is to calculate the sum of values of all undirected simple graphs with exactly n labeled vertices. Print the answer modulo 998 244 353.

Note that a simple graph is an undirected graph in which both multiple edges and loops are disallowed. A connected component (or just component) of an undirected graph is a subgraph in which any two vertices are connected to each other by paths, and which is not connected to any other vertex in the graph.

Input

The first line contains an integer $T \le 100$, denoting the number of test cases. Each of next T lines contains two space-seperated integers n and k ($1 \le n \le 10^4$, $1 \le k \le 20$).

Output

For each test case, print a single line containing the answer.

Example

standard input	standard output
2 3 1 4 2	12 150

Problem E. Reachable Sequences

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

ZZX has a sequence a, which is a permutation of $1, 2, \ldots, n$. Now ZZX wants to perform some modifications on this sequence. For each modification, he can choose a pair of integers i and j, satisfying $1 \le i < j \le n$ and $a_i > a_j$, and then swap a_i and a_j.

If a permutation b can be obtained by performing some (possibly zero) modifications on the initial sequence a, then ZZX says b is *reachable* from a.

Now JRY has m sequences $a^{(1)}$, $a^{(2)}$, …, $a^{(m)}$. Each of them is a permutation of $1, 2, \ldots, n$. He wants to know how many pairs (i, j) such that $1 \le i \le m$ and $1 \le j \le m$ have the property that a_i is reachable from a_j.

Input

The first line contains an integer T. Then T test cases follow. In each test case:

The first line contains two integers n and m. After that, m lines follow. The k-th of them contains n integers $a_1^{(k)}$, $a_2^{(k)}$, …, $a_n^{(k)}$. Each $a^{(k)}$ is a permutation of $1, 2, \ldots, n$.

There are at most 1000 small test cases and 1 large test case. The small test cases satisfy $1 \le n \le 5$ and $1 \le m \le 500$. The large test case satisfies $1 \le n \le 9$ and $1 \le m \le 3 \cdot 10^5$.

Output

For each test case, print the answer on a separate line.

Example

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

Problem F. Physics

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

There are n balls on a smooth horizontal straight track. The track can be considered to be a number line. The balls can be considered to be particles with the same mass.

At the beginning, ball i is at position X_i. It has an initial velocity of V_i and is moving in direction $D_i = \pm 1$.

Additionally, a constant C is given. At any moment, ball i has acceleration A_i and velocity V_i, they have the same direction, and magically satisfy the equation that $A_i \cdot V_i = C$.

As there are multiple balls, they may collide with each other during the movement. We suppose all collisions are perfectly elastic collisions.

There are multiple queries. Each query consists of two integers t and k. Your task is to find out the k-th smallest velocity of all the balls exactly t seconds after the beginning.

Note that *perfectly elastic collision* is defined as one in which there is no loss of kinetic energy in the collision.

Input

The first line of the input contains two integers n and C ($1 \le n \le 10^5$, $1 \le C \le 10^9$).

Then n lines follow. The i-th of them contains three integers V_i, X_i, D_i. V_i denotes the initial velocity of ball i ($1 \le V_i \le 10^5$), X_i denotes the initial position of ball i ($1 \le X_i \le 10^9$), D_i denotes the direction ball i moves in.

The next line contains an integer $1 \le q \le 10^5$, denoting the number of queries. After that, q lines follow. Each line contains two integers $1 \le t \le 10^9$ and $1 \le k \le n$.

Output

For each query, print a single line containing the answer with absolute error at most 10^{-3}.

Example

standard input	standard output
3 7 3 3 1 3 10 -1 2 7 1 3 2 3 1 2 3 3	6.083 4.796 7.141

Problem G. Parentheses

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

A *correct parentheses sequence* can be defined recursively as follows:

- The empty string is a correct sequence.
- If X and Y are correct sequences, then XY (the concatenation of X and Y) is a correct sequence.
- If X is a correct sequence, then `(`X`)` is a correct sequence.

Each correct parentheses sequence can be derived using the above rules.

For a parentheses sequence, you can make some operations with it.

- Each time you can choose two indices L and R such that $L \le R$. The operation modifies the characters on indices from L to R, inclusive.
- First, the order of these characters is reversed.
- Then, each character is toggled to the opposite one. That is, each '`(`' in the specified range changes to a '`)`' and vice versa.

The *value* of a parentheses sequence is the minimal number of the operations required to change it into a correct parentheses sequence. If it is impossible, the value of the sequence is equal to 10^{100}.

For example, the value of "`()((`" is 1, the value of "`()()`" is 0, and the value of "`(((`" is 10^{100}.

You are given an integer n. For each $1 \le i \le n$, find the number A_i of different parentheses sequence of length n which has value i, and then calculate the sum $\sum_{i=0}^{n}((i+1) \cdot A_i)$.

The answer may be very large, so print it modulo the given integer m.

Input

The first line of the input contains two integers n and m ($1 \le n \le 10^6$, $1 \le m \le 10^9$).

Output

Print one integer: the answer to the problem.

Examples

standard input	standard output
`1 100`	`0`
`10 100`	`68`

Problem H. Array and Operations

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1.5 seconds
Memory limit:	512 mebibytes

You have an array a with n integers. There are three types of operations:

- "`1` l r x": for each i in $[l, r]$, change a_i to $a_i + x$;
- "`2` l r": for each i in $[l, r]$, change a_i to $\lfloor\sqrt{a_i}\rfloor$;
- "`3` l r": sum up a_i for all i in $[l, r]$ and print the answer.

Your goal is to process the operations and print the answers of all type 3 operations.

Input

The first line of the input contains two integers n and q ($1 \le n, q \le 10^5$). The second line contains n integers $a_1, \ldots, a_n$. Then q lines follow, each line describes an operation.

It is guaranteed that $1 \le a_i, x \le 10^5$, $1 \le l \le r \le n$.

Output

For each operation of type 3, print a single line containing the required sum.

Example

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

Problem I. Value of the Array

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

Yuta has a sequence of n integers $a_1, \ldots, a_n$ and a number k. For any non-empty subsequence S of this sequence, the *value* of S is defined as the sum of the largest $\min(|S|, k)$ numbers in S. The value of the array a is equal to the sum of the values of all its non-empty subsequences.

Now Yuta shows the n integers, and he wants to know the value of the array for each k in $[1, n]$.

Input

The first line of the input contains an integer n ($1 \le n \le 10^5$), the length of the sequence Yuta has. The second line contains n integers $a_1, \ldots, a_n$ ($0 \le a_i \le 10^9$), the sequence itself.

Output

Print a line that contains exactly n integers. The i-th number mjust be the value of the array when $k = i$. The answers may be very large, so you must print them modulo $998\,244\,353$.

Examples

standard input	standard output
3 1 1 1	7 11 12
5 1 2 3 4 5	129 201 231 239 240

Problem J. Values on a Tree

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

You are given a tree with n vertices. The length of each edge is exactly 1. For any non-empty subset S of the vertices, $value(S)$ is equal to the maximum of $dis(u, v)$ over all pairs $(u, v) \in S$, where $dis(u, v)$ is equal to the distance between u and v in the tree.

It is easy to find that $value(S)$ satisfies $0 \le value(S) < n$. For each $0 \le K \le n - 1$, print the number of the subsets S such that $value(S) = K$.

Input

The first line of input contains an integer n ($1 \le n \le 3000$), the number of vertices in the graph. Then $n - 1$ lines follow. Each of them contains two integers u and v which mean that there is an edge between u and v ($1 \le u, v \le n$). It is guaranteed that the given graph is a tree.

Output

Print a line containing exactly n integers. The i-th integer must be the number of non-empty subsets S which satisfy $value(S) = i - 1$. The answers may be very large, so print each answer modulo 998 244 353.

Examples

standard input	standard output
2 1 2	2 1
4 1 3 2 4 4 1	4 3 4 4