Problem A. Two Trees

Input file:	*standard input*
Output file:	*standard output*
Time limit:	8 seconds
Memory limit:	256 mebibytes

Given are trees T_1 and T_2. Each tree has n vertices numbered from 1 through n. Let $d(v, u, T)$ denote the number of edges on the path between vertices v and u in tree T. Calculate the following sum:

$$\sum_{v=1}^{n}\sum_{u=1}^{n}\left(d(v,u,T_1)+d(v,u,T_2)\right)^2 .$$

As the answer may be large, find it modulo 2^{32}.

Input

The first line contains one integer n: the number of vertices in each tree ($1 \le n \le 100\,000$).

Each of the next $n-1$ lines contains two integers, u and v, denoting an edge between vertices u and v in tree T_1 ($1 \le u, v \le n$).

Each of the last $n-1$ lines contains two integers, u and v, denoting an edge between vertices u and v in tree T_2 ($1 \le u, v \le n$).

Output

Print the answer modulo 2^{32}.

Examples

standard input	*standard output*
3 1 2 1 3 1 2 1 3	24
3 1 2 1 3 1 2 2 3	22

Problem B. Tarzan Jumps

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

In a forest near Almaty, there are N trees arranged in a row, numbered from 1 through N from left to right. Tree number i has height H_i.

In one jump, Tarzan can move from the top of tree i to the top of tree j ($i < j$) if all the trees between them are either strictly lower or strictly higher than both trees i and j. In particular, he can jump from tree i to tree $i+1$. More formally, the jump is possible if at least one of the following conditions holds:

- $j = i + 1$,
- for all k ($i < k < j$): $H_i > H_k$ and $H_j > H_k$,
- for all k ($i < k < j$): $H_i < H_k$ and $H_j < H_k$.

Tarzan is currently standing on tree 1, and he wants to reach tree N. Tarzan's ICPC teammate, Abay, can help him. Specifically, he can perform the following change any number of times: choose a number i ($1 \le i \le n$), an integer x ($0 \le x \le 10^{18}$), and set $H_i = x$.

For each k from 1 to N, find the least number of changes that Abay must perform so that Tarzan could get to tree N in no more than k jumps.

Input

The first line contains a single integer t, the number of test cases ($1 \le t \le 150\,000$). The description of test cases follows.

The first line of each test case contains an integer N, the number of trees ($2 \le N \le 300\,000$).

The second line of each test case contains N integers $H_1, H_2, \ldots, H_N$ ($1 \le H_i \le 10^9$).

It is guaranteed that the sum of N over all test cases does not exceed $300\,000$.

Output

For each test case, print N integers: for each k from 1 to N, print the least number of changes that Abay must perform so that Tarzan could get from tree 1 to tree N in no more than k jumps.

Example

standard input	*standard output*
2 3 2 2 4 2 1 1	1 0 0 0 0

Note

In the first test case, for $k = 1$, Abay can change the height of tree 1 to 3, and Tarzan will be able to jump to the last tree. For $k = 2$ and $k = 3$, Tarzan can reach the last tree without any changes.

Problem C. Inversions

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

For a permutation p, denote the number of inversions in it as $inv(p)$. An inversion is a pair of indices $1 \le i < j \le |p|$ such that $p_i > p_j$.

Given are integers n and k. Find the sum of $inv(p)^k$ over all permutations p of length n. As the answer can be very large, find it modulo 998 244 353.

Input

The only line contains two integers, n and k ($1 \le n \le 10^{18}$, $1 \le k \le 1000$).

Output

Print the answer modulo 998 244 353.

Examples

standard input	*standard output*
3 2	19
5 3	22500

Note

In the first example:

In permutation $(1, 2, 3)$, there are 0 inversions.

In $(1, 3, 2)$, there is 1 inversion.

In $(2, 1, 3)$, there is 1 inversion.

In $(2, 3, 1)$, there are 2 inversions.

In $(3, 1, 2)$, there are 2 inversions.

In $(3, 2, 1)$, there are 3 inversions.

The answer is: $0^2 + 1^2 + 1^2 + 2^2 + 2^2 + 3^2 = 19$.

Problem D. Hidden Rook

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

This problem is interactive.

Roman hid a rook on an $n \times m$ chessboard. You need to find its exact position. You can ask Roman the following question **at most 4 times**: "How many cells (i, j), where $X_1 \le i \le X_2$ and $Y_1 \le j \le Y_2$, are under the hidden rook's attack?" A rook attacks all cells in the same row or column, including its own cell.

Input

The first line contains an integer t, the number of test cases ($1 \le t \le 15\,000$).

Interaction Protocol

The interaction in each test case starts with two integers, n and m: the chessboard dimensions ($3 \le n, m \le 15$).

To ask Roman a question, print "`?` X_1 Y_1 X_2 Y_2" ($1 \le X_1 \le X_2 \le n$, $1 \le Y_1 \le Y_2 \le m$). After that, you will receive an integer K: the number of cells (i, j), where $X_1 \le i \le X_2$ and $Y_1 \le j \le Y_2$, that are under the hidden rook's attack. You can ask at most 4 questions in each test case.

To report the answer, print "`!` X Y", where (X, Y) is the hidden rook's cell.

After making each query, do not forget to print the newline character and flush the output. You can use the following commands:

- fflush(stdout) or cout.flush() in C++;
- System.out.flush() in Java;
- flush(output) in Pascal;
- stdout.flush() in Python;

for other languages, see their documentation. You will get the "`Idleness limit exceeded`" verdict if you fail to do so.

Example

standard input	*standard output*
2	
6 6	
	? 1 1 3 6
8	
	? 2 2 2 3
2	
	! 2 3
7 5	
	? 1 1 7 5
11	
	? 1 1 1 4
4	
	! 1 4

Problem E. Mountains

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

Damir is climbing mountains. The mountain map can be represented as an $n \times m$ grid, in which a cell at the intersection of row i and column j is denoted as (i, j). The height of the peak in cell (i, j) is equal to a non-negative integer $a_{i,j}$. Damir starts his journey on the peak in cell $(1, 1)$ aiming to reach the peak in cell (n, m). If Damir is in on the peak in cell (i, j), then he can go either to the peak in cell $(i + 1, j)$ or to the peak in cell $(i, j + 1)$. Of course, he cannot go outside the boundaries of the map. To make the journey more interesting, he chooses the path with the *largest sum of peak heights* (kind of total climb).

Damir loves combinatorics, and he became curious: how many $n \times m$ maps are there such that the sum of peak heights on his path does not exceed k? As the answer may be large, find it modulo $10^9 + 7$.

Input

The only line of input contains three integers, n, m, and k ($1 \le n, m, k \le 100$).

Output

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

Examples

standard input	*standard output*
1 1 1	2
2 2 2	20
2 3 4	490

Problem F. Kill All Termites

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

A tree is an undirected connected graph with n vertices and $n-1$ edges.

You are given a tree. There are termites in some vertices of this tree. Your task is to kill them all. To do so, you can poison some of the vertices. If a termite visits a poisoned vertex, it immediately dies. Every second, each termite moves to an adjacent vertex. A termite cannot move along the same edge twice in a row, except when it gets into a leaf. Find the minimal possible number of vertices you can poison so that all the termites will eventually die, regardless of their initial positions and strategies.

Input

The first line contains one integer n, the size of the tree ($1 \le n \le 100\,000$).

The second line contains $n-1$ integers $p_2, p_3, \ldots, p_n$, meaning that there is an edge between vertices i and p_i for $2 \le i \le n$ ($1 \le p_i < i$).

Output

Print one integer: the answer.

Examples

standard input	*standard output*
1	1
2 1	1
8 1 1 2 1 2 3 2	2

Problem G. Maximal Subsequence

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

Let the *beauty* of a sequence be the length of its longest increasing subsequence.

You are given an array a consisting of n integers. Find the maximum length of a subsequence of array a such that the beauty of this subsequence is less than the beauty of the whole array a.

Input

The first line contains a single integer n, the number of elements in array a ($1 \le n \le 5 \cdot 10^5$).

The second line contains n space-separated integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

Output

Print one integer: the maximum length of a subsequence of array a such that its beauty is less than the beauty of the whole array a.

Examples

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

Problem H. Aidana and Pita

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

Aidana loves pita. Yesterday she brought home n pitas. Each pita has a *tastiness* value, which is an integer. Today three friends of Aidana will come to dinner. She will distribute all n pitas among them; each pita will go to exactly one friend. The *happiness* of a friend is the sum of tastiness values of all pitas he or she received. Aidana wants to be fair. So help her find a distribution of pitas that minimizes the difference between her friends' maximum and minimum happiness.

Input

The first line contains one integer n, the number of pitas ($3 \le n \le 25$).

The second line contains n integers, $a_1, a_2, \ldots, a_n$, which are tastiness values for pitas ($1 \le a_i \le 10^7$).

Output

Print n integers: for each pita, print the friend's index to which this pita should go (friends' indices are 1, 2, and 3).

If there are several possible answers which minimize the difference between maximum and minimum happiness, print any one of them.

Examples

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

Problem I. Box Packing

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

An ordered pair of integers (x, y) is called a *box*. A sequence of boxes $(c_1, d_1),\ (c_2, d_2),\ \ldots,\ (c_m, d_m)$ is called a *chain* if the following inequalities hold:

$$c_1 \le c_2 \le \ldots \le c_m, \quad d_1 \le d_2 \le \ldots \le d_m.$$

You are given n boxes: $(a_1, b_1),\ (a_2, b_2),\ \ldots,\ (a_n, b_n)$. Find the maximum number of boxes that you can select from them and split into no more than k chains. You can reorder the boxes to form chains.

Input

The first line contains two integers, n and k ($1 \le n \le 10^5$, $1 \le k \le 100$).

The i-th of the following n lines contains two integers, a_i and b_i ($1 \le a_i,\ b_i \le 10^9$).

Output

Print one integer: the answer.

Examples

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

Problem J. Two Permutations

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

Find the number of pairs (p, q) of permutations of length n such that $\sum_{i=1}^{n} \max(p_i, q_i) = k$.

As the answer may be large, find it modulo $10^9 + 7$.

Input

The only line contains two integers, n and k ($1 \le n \le 100$, $1 \le k \le n^2$).

Output

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

Examples

standard input	*standard output*
2 4	2
3 7	12

Problem K. Fancy Arrays

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2.5 seconds
Memory limit:	256 mebibytes

Let us fix an integer m. Consider an array a consisting of n positive integers. The array a is *fancy* if each number in a is a divisor of m, and each two neighboring numbers in a are **not** coprime.

Find the total number of fancy arrays of length n. As the answer may be large, find it modulo $10^9 + 7$.

Input

The first line contains two integers, m and q: the number introduced above and the number of queries ($1 \le m \le 10^{16}$, $1 \le q \le 150$).

Each of the next q lines contains a single integer n ($1 \le n \le 10^{18}$).

Output

For each query, print the number of fancy arrays for the given m and n modulo $10^9 + 7$.

Example

standard input	*standard output*
12 3 1 2 3	6 21 91

Problem L. Restricted Arrays

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	256 mebibytes

Let n be a positive integer. Find the number of integers $1 \le M \le n$ for which there exists an array of integers $a[1..n]$ that satisfies the following conditions:

$$a[x_i] + 1 \equiv a[y_i] \pmod{M}, \quad 1 \le i \le q.$$

Input

The first line contains two integers, n and q: the array size and the number of conditions ($1 \le n, q \le 10^6$).

Each of the next q lines contains two integers, x_i and y_i: the indices describing the corresponding condition ($1 \le x_i, y_i \le n$).

Output

On the first line, print an integer t: the number of possible values of M. On the second line, print the t possible values of M in increasing order.

Examples

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