Problem A. Zero Sum

Input file:	*standard input*
Output file:	*standard output*
Time limit:	7 seconds
Memory limit:	256 mebibytes

You are given a matrix a of size $n \times (2k+1)$, which contains integers, rows are numbered from 1 to n, and columns are numbered from $-k$ to k.

You need to choose the sequence of numbers $x_1, x_2, \ldots, x_n$, such that contraints $(-k \le x_i \le k)$ and $(x_1 + x_2 + \ldots + x_n = 0)$ will hold, and, under this, the value of $a_{1,x_1} + a_{2,x_2} + \ldots + a_{n,x_n}$ will be as small as possible.

Input

The first line contains two integers n and k $(1 \le n \le 35\,000, 1 \le k \le 3)$, separated by a space: the dimensions of the matrix a.

The following n lines contain $(2k+1)$ integers separated by a space: the j-th number in the i-th of these lines denotes $(j-k-1)$-th element of i-th row of the matrix a $(-10^9 \le a_{i,j-k-1} \le 10^9)$.

Output

Print one integer: the minimum possible value of the sum $a_{1,x_1} + a_{2,x_2} + \ldots + a_{n,x_n}$ under the constraints $(-k \le x_i \le k)$ and $(x_1 + x_2 + \ldots + x_n = 0)$.

Examples

standard input	standard output
3 1 3 14 15 -3 -5 -35 2 71 82	-19
5 2 1 2 5 14 42 1 2 3 5 8 1 2 4 8 16 1 2 3 4 5 1 2 6 24 120	16

Note

In the first sample optimal solution is to choose sequence $0, 1, -1$, which will give the required answer, which equals $15 + (-35) + 2 = -19$.

Problem B. MST

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

You are given an array $x_1, x_2, \ldots, x_n$.

Let's create an undirected graph on n vertices, which is initially empty.

After that, for each pair (u, v) such that $u < v$ let's add to the graph edge between vertices u and v with weight $x_v - x_u$.

Your goal is to find the weight of the minimum spanning tree in this graph.

Input

The first line of input contains one integer t ($1 \le t \le 300\,000$): the number of test cases.

The first line of each test case contains one integer n ($1 \le n \le 300\,000$): the number of integers in the given array.

The next line of each testcase contains n space-separated integers $x_1, x_2, \ldots, x_n$ ($-300\,000 \le x_i \le 300\,000$): the given array.

It is guaranteed that the sum of n is at most $300\,000$.

Output

For each test case one integer: the weight of the minimum spanning tree in the described graph.

Example

standard input	standard output
2 5 1 2 3 4 5 3 10 45 10	4 -35

Problem C. Tree Circles

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

You have a tree on n vertices, edges are numbered by distinct integers from 1 to $n-1$.

Let's call a circle from v with radius r a set of vertices in the connected component of v if you will leave only edges with numbers $\leq r$.

You need to answer several queries on the given tree.

In each query you are given k and k vertices $v_1, v_2, \ldots, v_k$.

You need to find the number of ways to pick a radius for each given vertex, such that all circles won't intersect.

In other words, you need to calculate the number of tuples $(r_1, r_2, \ldots, r_k)$ $(0 \leq r_1, r_2, \ldots, r_k \leq n-1)$ such that $circle(v_i, r_i) \cap circle(v_j, r_j) = \emptyset$ for $i \neq j$.

As the number may very big, you only need to find it modulo 998 244 353.

Input

The first line of input contains one integer n $(2 \leq n \leq 300\,000)$: the number of vertices in the given tree.

Next $(n-1)$ lines contain the description of edges, each line contain two integers u_i, v_i $(1 \leq u_i, v_i \leq n; u_i \neq v_i)$ describing edge connecting vertices u_i and v_i with number i in the tree.

It is guaranteed that the given graph is a tree.

The next line of input contains one integer q $(1 \leq q \leq n)$: the number of queries.

Next q lines contain the description of edges, each line contain one integer k $(1 \leq k \leq n)$, and k **distinct** integers after, $v_1, v_2, \ldots, v_k$ $(1 \leq v_i \leq n)$: the current query.

It is guaranteed that the sum of k is at most 300 000.

Output

For each query output one integer: the number of tuples $(r_1, r_2, \ldots, r_k)$ $(0 \leq r_1, r_2, \ldots, r_k \leq n-1)$ such that $circle(v_i, r_i) \cap circle(v_j, r_j) = \emptyset$ for $i \neq j$, modulo 998 244 353.

Example

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

Problem D. Angle Beats 2.0

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

You have a rectangular board consisting of $n \times m$ squares. Each square contains a character which is either "*" or ".".

A tromino is a figure formed by a square of the board, called the center, and two other squares, each sharing an edge with the center. A tromino is L-shaped if these two squares have a common vertex.

You can draw some disjoint L-shaped trominoes on the board. The center of an L-shaped tromino must contain "*", and each "*" should be a center of some tromino.

All non-center squares of all trominoes must contain ".".

Your goal is to find the number of ways to draw L-shaped trominoes under these constraints.

As the answer may very big, you only need to find it modulo 998 244 353.

Input

The first line of input contains one integer t ($1 \le t \le 250\,000$): the number of test cases.

The first line of each test case contains two integers n and m: the number of rows and columns of the board ($2 \le n, m \le 100$).

Each of the next n lines contains m characters, and each character is either "*" or ".". Together, these lines describe the board.

It is guaranteed that sum of $n \cdot m$ is at most 1 000 000.

Output

For each test case print one integer: the number of ways to draw L-shaped trominoes under given constraints.

Example

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

Problem E. LIS

Input file:	*standard input*
Output file:	*standard output*
Time limit:	7 seconds
Memory limit:	256 mebibytes

You have four sequences of integers $a_1, a_2, \ldots, a_n$; $b_1, b_2, \ldots, b_n$; $x_1, x_2, \ldots, x_n$; $y_1, y_2, \ldots, y_n$.

Let's build a directed graph, where the edge from i to j will be in the graph if $i < j$ and $a_i \cdot x_j + b_i \ge y_j$.

You need to find the longest path in this graph.

Input

The first line of input contains one integer t ($1 \le t \le 300\,000$): the number of test cases.

The first line of each test case contains one integer n ($1 \le n \le 150\,000$): the number of integers in the sequences.

Each of the next n lines contains four integers a_i, b_i, x_i, y_i ($0 \le a_i, x_i \le 300\,000; 0 \le b_i, y_i \le 10^{11}$).

It is guaranteed that the total sum of n is at most $300\,000$.

Output

For each test case print one integer: the longest path in the described graph.

Example

standard input	standard output
3 3 1 1 1 1 2 2 2 2 3 3 3 3 3 1 1 1 1 2 2 2 10 3 3 3 100 1 35 35 35 35	3 1 1

Problem F. Good Coloring

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

You have an undirected graph, each vertex is colored in one of k possible colors, the graph is properly colored into k colors, i.e two ends of any edge are colored in different colors.

Your goal is to find another (or maybe the same) coloring of this graph into x colors, such that $x \le k$, and there exists a path of length x, which contains all possible colors.

It is guaranteed that it is always possible.

Input

The first line of input contains one integer t ($1 \le t \le 600\,000$): the number of test cases.

The first line of each test case contains three integers n, m and k: the number of vertices, edges, and the number of colors you are using of the graph ($1 \le n \le 300\,000; 0 \le m \le 300\,000; 1 \le k \le n$).

The next line contains n space-separated integers $c_1, c_2, \ldots, c_n$ ($1 \le c_i \le k$): colors of vertices.

It is guaranteed that the given coloring is correct.

Each of the next m lines contains two integers, u and v ($1 \le u, v \le n; u \ne v$): indices of vertices connected by edge.

It is guaranteed that in each test case there are no multiple edges in the graph.

It is guaranteed that the sum of $n + m$ is at most $600\,000$.

Output

For each test case output $n + 1$ integers, x ($1 \le x \le k$), $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le x$): new coloring.

This coloring should be proper, i.e two ends of any edge are colored in different colors.

Also for each test case in next line print x integers $v_1, v_2, \ldots, v_x$ ($1 \le v_i \le n$), there should exists an edge between vertices v_i and v_{i+1}, and all colors of vertices should be different, so $p_{v_i} \ne p_{v_j}$ for all pairs $1 \le i < j \le x$.

Example

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

Problem G. Circle Convertation

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

You have two strings of zeroes and ones, $s_0, s_1, \ldots, s_{n-1}$ and $t_0, t_0, \ldots, t_{n-1}$.

In one operation you can choose i, such that $s_i = s_{(i+1) \bmod n}$, and invert s_i and $s_{(i+1) \bmod n}$. Invert s_i means set new value of s_i to '0' if it was equal to '1', and set it to '1' otherwise.

Your goal is to make $s_i = t_i$ for all i in at most 100 000 operations.

For each test in this problem, the solution exists. Note that for some pairs of strings you can't get one from other (for example "0101" and "1010"), but there are no such strings in the tests of this problem.

Input

The first line of input contains a binary string s.

The second line of input contains a binary string t.

$2 \le |s| = |t| \le 100$.

Output

In the first line print m ($0 \le m \le 100\,000$): the number of operations.

In the next line print m integers $i_1, i_2 \ldots, i_m$ ($0 \le i_j \le n-1$): operations in the order in which you need to perform them. Note, that when you are doing operation on index i, s_i should be equal to $s_{(i+1) \bmod n}$, and after this operation s_i and $s_{(i+1) \bmod n}$ will be changed.

Note that you don't necessarily need to minimize m.

It is guaranteed that there is at least one solution. If there are several possible solutions, you can print any.

Examples

standard input	standard output
000 011	1 1
0000 1111	2 0 2
110 011	2 0 1

Problem H. Equal MEX

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

You have an array $a_1, a_2, \ldots, a_n$.

You need to find the number of ways to split it into non-empty subsegments, such that all MEXes of these subsegments are equal. MEX of subsegment $[l \ldots r]$ is equal to minimal non-negative integer x, such that x is not present at this segment.

As this number may be very big, you only need to output it modulo 998 244 353.

Input

The first line of input contains one integer t ($1 \le t \le 300\,000$): the number of test cases.

The first line of each test case contains one integer n ($1 \le n \le 300\,000$): the number of integers in the given array. The next line of each testcase contains n space-separated integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le n$): the given array.

It is guaranteed that the sum of n is at most 300 000.

Output

For each test case one integer: the number of ways to split a given array into non-empty subsegments with equal MEX, modulo 998 244 353.

Example

standard input	standard output
4 6 0 0 0 1 1 1 5 0 1 0 1 0 4 0 0 0 0 3 3 3 3	1 3 8 4

Problem I. Cactus is Money

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

A **Cactus graph** is a simple connected undirected graph where each edge lies in at most one simple cycle.

You have a cactus graph, each edge has two non-negative integer weights a_i, b_i.

Your goal is to find the spanning tree of given cactus with a minimum value of $(\sum a_i) \cdot (\sum b_i)$, where the sum is taken among all edges which are present in spanning tree.

Input

The first line contains n, m, denoting the number of vertices and edges of the cactus graph. $(1 \le n \le 50\,000, 0 \le m \le 250\,000)$

In the next m lines, four integers s, e, a_i, b_i denoting endpoints of the i-th edge and its weights are given. $(1 \le s, e \le n, s \ne e, 0 \le a_i, b_i \le 50000)$.

It is guaranteed that the graph is connected, it does not contain loops or multiple edges, and every edge belongs to at most one simple cycle.

Output

Output one integer: minimum possible value of $(\sum a_i) \cdot (\sum b_i)$, where the sum is taken among all edges which are present in spanning tree.

Example

standard input	standard output
3 3 1 2 0 1000 2 3 0 1000 3 1 1 1	0

Problem J. Good Permutations

Input file:	*standard input*
Output file:	*standard output*
Time limit:	7 seconds
Memory limit:	256 mebibytes

Let's call a permutation of n elements **good**, if there are exactly m triples i, j, k such that $1 \le i < j < k \le n$ and $p_i < p_j < p_k$.

You need to calculate the total number of inversions of all good permutations of n elements, modulo 998 244 353 (prime).

Input

The first line of input contains two integers n and m ($1 \le n \le 100\,000, 0 \le m \le 3$).

Output

Output one integer: the sum of the number of inversions of all permutations $p_1, p_2, \ldots, p_n$, such that there are exactly m triples i, j, k such that $1 \le i < j < k \le n$ and $p_i < p_j < p_k$, modulo 998 244 353.

Examples

standard input	standard output
2 0	1
3 0	9
4 0	55
5 0	290
4 2	3
5 2	98
6 2	1074
5 3	21
6 3	484
7 3	5430

Problem K. Number Theory

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

You are given a prime p.

For integer x, such that $0 \le x < p$ let's call $f(x)$ the minimum non-negative integer a, such that there exists b, such that $(a^2 + b^2) \mod p = x$.

Your goal is to find $\max(f(0), f(1), \ldots, f(p-1))$.

It can be proved that for each prime p and each integer x you can find at least one pair a, b such that $(a^2 + b^2) \mod p = x \mod p$.

Input

The first line of input contains one integer p ($2 \le p \le 10^5$).

It is guaranteed that p is prime.

Output

Print one integer: $\max(f(0), f(1), \ldots, f(p-1))$.

Examples

standard input	standard output
2	0
3	1
5	2
7	2
99991	20

Problem L. Modulo Magic

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

You have a positive integer n.

You need to find the number of different integers among $n \mod 1, n \mod 2, \ldots, n \mod (n-1)$.

Input

The first line of input contains one integer n $(2 \le n \le 10^9)$.

Output

Print one integer: the number of different integers among $n \mod 1, n \mod 2, \ldots, n \mod (n-1)$.

Examples

standard input	standard output
2	1
3	2