Problem A. Challenge Matrix Multiplication II

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

As is well known, the APSP (All Possible Shortest Path) problem has a profound connection with min+ matrix multiplication. Today, we want you to solve the following problem:

Given x, construct an **undirected unweighted connected graph** (i.e., all edge weights are 1) such that $\sum_{i=1}^{n} \sum_{j=i+1}^{n} dis(i, j) = x$. Here, $dis(i, j)$ represents the shortest path length between node i and node j.

This problem is certainly very simple... Unfortunately, your resources are limited, so you hope to find a graph with no more than 85 nodes that meets the requirements.

Input

A single test case contains multiple data sets.

The first line of input is the number of data sets T ($1 \le T \le 2000$), representing the number of data sets contained in this test case.

For each data set, there is one line with an integer x ($1 \le x \le 10^5$), representing the constraint that needs to be satisfied. It is guaranteed that there exists at least one graph with no more than 85 nodes that meets the condition.

Output

For each data set, output the number of nodes n you used on the first line ($2 \le n \le 85$). Then output $n - 1$ lines, where the i-th line is a binary string of length $n - i$. The j-th character of the i-th line is 1 if and only if there is an edge between node i and node $i + j$.

Example

standard input	standard output
4 1 3 6 8	2 1 3 11 1 4 111 11 1 4 110 10 1

Problem B. Cutting Chocolate

Input file: standard input
Output file: standard output
Time limit: 1.5 seconds
Memory limit: 1024 megabytes

Little L has a piece of chocolate of size $L \times W \times H$. Inside the chocolate, there are n almonds, with the i-th almond located at the center of the unit cube $[x_i - 1, x_i] \times [y_i - 1, y_i] \times [z_i - 1, z_i]$. No two almonds occupy the same position.

Now, Little L wants to cut this piece of chocolate into several smaller pieces. Specifically, he wants to make p, q, r cuts along the three dimensions, respectively. Each cut must be made along integer coordinates, no two cuts can be made at the same coordinate in the same dimension, and cuts cannot be made at the edges of the chocolate (i.e., at coordinates 0 or L, W, H).

Formally, a cut in the x direction at coordinate k $(0 < k < L)$ means cutting the chocolate along the plane $x = k$. The same applies for the y and z directions.

Additionally, for the $(p+1)(q+1)(r+1)$ pieces of chocolate after the cuts, Little L hopes that each piece contains the same number of almonds. Little L wants to know how many essentially different cutting methods satisfy this condition. Two cutting methods are considered different if there exists a position in one dimension such that the first method cuts at this position while the second does not. Output the result modulo $10^9 + 7$.

Input

The first line of each test case contains three integers L, W, H $(1 \le L, W, H \le 10^9)$, representing the size of the chocolate.

The second line contains three integers p, q, r $(0 \le p, q, r \le 10^6)$, representing the number of cuts in the x, y, and z directions, respectively.

The third line contains an integer n $(1 \le n \le 10^6)$, representing the number of almonds.

The following n lines each contain three integers x_i, y_i, z_i $(1 \le x_i \le L, 1 \le y_i \le W, 1 \le z_i \le H)$, representing the position of an almond. It is guaranteed that all triples (x_i, y_i, z_i) are distinct.

Output

A single line containing an integer that represents the result modulo $10^9 + 7$.

Examples

standard input	standard output
4 4 1 1 1 0 4 1 2 1 2 4 1 3 1 1 4 3 1	1
3 3 3 1 1 1 3 1 1 1 2 2 2 3 3 3	0
9 9 9 1 1 1 8 1 1 2 2 2 8 1 9 1 2 8 9 8 2 1 9 2 9 9 8 1 8 9 7	180

Problem C. Basic Counting Practice Problems

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

Consider a tree with n nodes, where node 1 is the root. Each node i in the tree has a weight p_i, such that $p_1, p_2, \ldots, p_n$ is a permutation.

Let $L_{u,v}$ denote the maximum weight of any node on the simple path from node u to node v (including u and v). Let $subtree(i)$ represent the set of nodes that make up the subtree of node i. We define $S_i = \{u | u \in subtree(i) \wedge L_{u,i} \leq p_i\}$, which is the set of nodes that can be reached starting from i, only moving to children and only passing through nodes with weights less than or equal to p_i. Let $f_i = |S_i|$, which represents the size of the set of nodes.

We define the value of a node as q_{f_i}, where q is a given value sequence $q_1, q_2, \ldots, q_n$.

Now we want to compute, for each pair (i, j), the sum of the values of node i for all weight configurations that satisfy $p_i = j$. Since the answer may be very large, please output the result modulo $10^9 + 7$.

Input

The first line contains one integer n ($1 \leq n \leq 700$), representing the number of nodes in the tree.

The next line contains n integers, representing $q_1, q_2, \ldots, q_n$ ($0 \leq q_i < 10^9 + 7$).

The following $n - 1$ lines each contain two integers u, v, representing an edge in the tree.

Output

Output n lines, each containing n integers. The i-th line and the j-th integer represent the corresponding answer modulo $10^9 + 7$.

Examples

standard input	standard output
3 1 1 2 1 2 2 3	2 2 4 2 2 2 2 2 2
4 1 2 3 4 1 2 2 3 1 4	6 10 16 24 6 8 10 12 6 6 6 6 6 6 6 6
7 3 7 2 4 1 8 9 1 2 1 3 2 4 4 5 3 6 4 7	2160 3120 3168 2952 2160 3720 6480 2160 2640 2640 2412 2208 2280 2880 2160 2640 3120 3600 4080 4560 5040 2160 3120 3648 3744 3408 2640 1440 2160 2160 2160 2160 2160 2160 2160 2160 2160 2160 2160 2160 2160 2160 2160 2160 2160 2160 2160 2160 2160

Problem D. Diameter of a Tree

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 1024 megabytes

Given an unrooted tree with n nodes, you need to assign a **distinct** weight to each node, with weights forming a permutation of the numbers from 1 to n.

For any assignment scheme, we consider all diameters of the tree (i.e., the longest simple path in the tree). Among these diameters, we select the one with the **lexicographically largest weight sequence** (arranging the weights of the nodes along the path in order and comparing lexicographically).

Your goal is to **minimize** the lexicographic order of the weight sequence corresponding to this "lexicographically largest diameter"among all possible assignment schemes.

Input

A single test case contains multiple data groups.

The first line of input is the number of data groups T $(1 \le T \le 10^5)$, representing the number of data groups in this test case.

For each data group, the first line contains an integer n $(2 \le n \le 2 \cdot 10^5)$ indicating the number of nodes.

The next $n-1$ lines each contain two integers u, v $(1 \le u, v \le n)$, representing an edge of the tree.

It is guaranteed that across all data, $\sum n \le 2 \cdot 10^5$.

Output

For each set of data, output a line of several integers representing the smallest possible lexicographical order of the diameter with the largest lexicographical order.

Example

standard input	standard output
5 5 1 2 1 3 2 4 2 5 4 1 2 1 3 1 4 6 1 2 1 3 2 4 2 5 3 6 8 1 2 2 3 1 4 2 5 3 6 1 7 7 8 10 1 2 2 3 2 4 4 7 4 8 1 5 5 6 6 9 1 10	3 4 5 1 3 4 2 3 4 5 6 1 2 3 4 5 6 1 3 4 5 6 7 8 1

Problem E. Doo Doo Doo

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Little Ika: Doo Doo Doo $\sim$.

Hyacinthia wants to go to the center of the "Cloud End Ruins"Eye of Dawn to search for loot, but there are n walls blocking her path. The i-th wall is a circle centered at the origin $(0, 0)$ with a radius of r_i. There are k_i doors on the wall, and the coordinates of the j-th door are given by $\left(r_i \cdot \cos\left(\frac{2\pi}{M} \cdot p_{i,j}\right), r_i \cdot \sin\left(\frac{2\pi}{M} \cdot p_{i,j}\right)\right)$. The thickness of the wall and the width of the doors are negligible, and Hyacinthia cannot fly over the walls.

Hyacinthia wants to ask you m times. Each time she will tell you her starting point, and you need to answer what the shortest distance to the center is. The starting point is guaranteed to be on the inner side of a wall and right next to the wall surface.

Input

The first line contains two integers n, m $(2 \le n \le 2 \cdot 10^5, 1 \le m \le 2 \cdot 10^5)$, representing the number of walls and the number of queries.

The next n lines describe each wall. Each line starts with two integers r_i, k_i $(1 \le r_i \le 10^8, 0 \le k_i \le 2 \cdot 10^5)$, representing that the i-th wall is a circle with radius r_i and has k_i doors. Following this are k_i integers $p_{i,1}, p_{i,2}, \ldots, p_{i,k_i}$, representing that the coordinates of the j-th door are $\left(r_i \cdot \cos\left(\frac{2\pi}{M} \cdot p_{i,j}\right), r_i \cdot \sin\left(\frac{2\pi}{M} \cdot p_{i,j}\right)\right)$, where $M = 3.6 \cdot 10^8$. It is guaranteed that $0 \le p_{i,1} < p_{i,2} < \cdots < p_{i,k_i} < M$, $k_n = 0$, $\sum_{i=1}^{n} k_i \le 2 \cdot 10^5$, and $1 \le r_1 < r_2 < \cdots < r_n \le 10^8$.

The next m lines each contain two integers t_i, q_i, indicating that if Hyacinthia starts from the inside of the t_i-th wall at the point $\left(r_{t_i} \cdot \cos\left(\frac{2\pi}{M} \cdot q_i\right), r_{t_i} \cdot \sin\left(\frac{2\pi}{M} \cdot q_i\right)\right)$, what is the shortest distance to the origin (if it is not reachable, please output "-1").

Output

For each query, output a single line with a floating-point number representing the answer. If it is not reachable, output "-1". An answer is considered correct if the relative or absolute error does not exceed 10^{-6} compared to the standard answer.

Example

standard input	standard output
3 4 2 2 0 90000000 5 0 8 0 1 114514 2 0 2 180000000 3 233	2.0000000000 5.0000000000 7.4056093871 -1

Problem F. Challenge NPC II

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

Tajian is solving the following problem:

Given a graph with n points, the initial graph has no edges. Tajian has some edges to add to the graph, and he has divided these edges into k groups, where the i-th group contains c_i edges. For each $x \in [1, n]$, Tajian wants to know the minimum number of groups of edges he needs to select from these k groups to add to the graph in order to ensure that the number of connected components in the graph does not exceed x.

Although it is not immediately obvious, there are papers that indicate that the minimum coloring number of a spanning tree is an NPC problem. However, Tajian does not know this, so he wants you to help him solve this problem.

Input

A single test case contains multiple data sets.

The first line of input is the number of data sets T $(1 \le T \le 10^3)$, representing the number of data sets in this test case.

For each data set, the first line contains two integers n, k $(2 \le n \le 16,\ 1 \le k \le 200)$, representing the number of points and the number of groups of edges.

The next section describes the information for the k groups of edges. For the i-th group of edges, the first line contains an integer c_i $(1 \le c_i \le 200)$, representing the number of edges in this group. The following c_i lines each contain two integers u, v $(1 \le u, v \le n, u \ne v)$, representing an edge in this group.

For a single data set, it is guaranteed that $\sum c_i \le 200$. For all data, it is guaranteed that $\sum 2^n \le 2^{16}$.

Output

For each data set, output a line of n integers, where the i-th integer represents the answer when $x = i$. If it is impossible to ensure that the number of connected components does not exceed i under any circumstances, output ”-1”.

Example

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

Problem G. Plus Xor

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Given integers a, b, c, you can perform one of the following operations each time:

- $a \leftarrow a + b$
- $a \leftarrow a \oplus b$

where $\oplus$ represents the binary XOR operation.

You can perform any number of operations, and you want to know if you can ultimately make a equal to c.

Input

A single test case contains multiple data sets.

The first line of the input is the number of data sets T ($1 \leq T \leq 10^3$), representing the number of data sets in this test case.

For each data set, there is a line with three integers a, b, c ($0 \leq a, c \leq 10^{18}, 1 \leq b \leq 1000$).

It is guaranteed that in a single test case, $\sum b^2 \leq 10^6$.

Output

For each data set, if it is possible to ultimately make a equal to c, output "YES"; otherwise, output "NO".

Example

standard input	standard output
5 1 6 7 7 5 13 8 3 16 7 6 17 2 7 8	YES NO YES YES NO

Problem H. Random Shuffle

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 1024 megabytes

There is an array a of length n, where the i-th number is a_i. Additionally, each number has a score, and the score of the i-th number is c_i. Alice and Bob take turns, with Alice going first.

In the current round, the player first copies the array a to array b and runs a random shuffle program to randomly shuffle array b. Specifically, the random shuffle program will select one of the permutations of b with equal probability and replace the original array b with it. Suppose the current length of array a is m, then the player can choose an integer k such that $0 \le k \le m$ and $a[1 \dots k] = b[1 \dots k]$, and then delete $a[1 \dots k]$ (i.e., the prefix of length k of a) and obtain the score of the deleted numbers. The game ends when the array a is emptied.

Note that the array a and its score array c will not be shuffled throughout the entire process, but only their prefixes will be removed.

Both Alice and Bob want to maximize their scores. Alice wants to know what her expected score is when both players adopt optimal strategies.

Alice also made q groups of modifications (modifying the array and scores), and she wants you to find the answer for the initial array, as well as the answers after each group of modifications.

Input

The first line contains an integer n ($1 \le n \le 10^6$), representing the length of the initial array a.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le n$), representing the initial array a.

The third line contains n integers $c_1, c_2, \dots, c_n$ ($1 \le c_i \le 10^6$), representing the score of each number in the array.

The next line contains an integer q ($0 \le q \le 10$), representing the number of modifications.

The following describes q groups of modifications:

For each group of modifications, the first line contains k_i ($1 \le k_i \le 10^5$), indicating the number of modifications in this group.

The next k_i lines each contain three integers p_j, x_j, y_j ($1 \le p_j, x_j \le n, 1 \le y_j \le 10^6$), indicating that a_{p_j} is modified to x_j and c_{p_j} is modified to y_j.

It is guaranteed that $p_1 < p_2 < \dots < p_{k_i}$.

Output

Output a total of $q + 1$ lines, each line containing a floating-point number representing the answer.

The first line is the answer for the initial array, followed by q lines for the answers after each group of modifications.

An answer is considered correct if the relative or absolute error compared to the standard answer does not exceed 10^{-9}.

Examples

standard input	standard output
2 1 2 1 1 2 1 2 1 2 1 1 1 2	1.333333333333 3.000000000000 4.000000000000
6 1 1 4 5 1 4 2 3 3 3 3 3 0	9.013888888889

Note

This problem has a large input size, please use an efficient input method.

Problem I. Dumb Problem II

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	1024 megabytes

For a permutation p of length n, we define $f(p)$ to represent the position of the prefix maximum. Specifically, $f(p) = \{i | 1 \le i \le n, \text{s.t. } \forall 1 \le j < i, p_i > p_j\}$.

Similarly, let $g(p)$ represent the set of numbers of the prefix maximum, that is, $g(p) = \{p_i | i \in f(p)\}$.

Now there are k independently and uniformly randomly chosen permutations $p_1, p_2, \ldots, p_k$ from all permutations of length n (can be repeated). Little A wants to know how many essentially different $g(p)$ there are expected among these permutations. The goal is to find out how many essentially different sets are expected among $g(p_1), g(p_2), \ldots, g(p_k)$? The answer should be output modulo 998244353.

Input

A single line containing two integers n, k $(1 \le n, k \le 5000)$.

Output

A single line containing an integer representing the result of the answer modulo 998244353.

Examples

standard input	standard output
1 1	1
3 2	388206139
3 4	408232647
112 646	928854225

Note

For the first sample, the answer is 1.

For the second sample, the answer is $\frac{31}{18}$.

For the third sample, the answer is $\frac{1711}{648}$.

Problem J. Subrectangle Count

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 1024 megabytes

Given an integer sequence of length $n(n \geq 2)$, denoted as $a_{1,2,\ldots,n}$, and an integer sequence of length $m(m \geq 2)$, denoted as $b_{1,2,\ldots,m}$, we can construct an $n \times m$ matrix c such that $c_{i,j} = a_i \oplus b_j (1 \leq i \leq n, 1 \leq j \leq m)$, where $\oplus$ denotes the bitwise XOR operation.

Determine how many pairs $(i, j)(1 \leq i < n, 1 \leq j < m)$ satisfy that the subrectangle region of c given by $\begin{bmatrix} c_{i,j} & c_{i,j+1} \\ c_{i+1,j} & c_{i+1,j+1} \end{bmatrix}$ is of the form $\begin{bmatrix} x & x+1 \\ x+2 & x+3 \end{bmatrix}$, meaning that $c_{i,j+1} = c_{i,j} + 1$, $c_{i+1,j} = c_{i,j} + 2$, and $c_{i+1,j+1} = c_{i,j} + 3$.

Input

The first line contains an integer $T(1 \leq T \leq 10^5)$, representing the number of test cases.

For each test case:

The first line contains two space-separated integers $n, m(2 \leq n, m \leq 2 \times 10^5)$, representing the number of rows and columns of c.

The second line contains n space-separated integers, representing $a_1, a_2, \ldots, a_n (0 \leq a_i < 2^{30})$.

The third line contains m space-separated integers, representing $b_1, b_2, \ldots, b_m (0 \leq b_i < 2^{30})$.

It is guaranteed that the sum of n does not exceed 2×10^5 and the sum of m does not exceed 2×10^5.

Output

For each test case, output a single line containing an integer, representing how many pairs (i, j) satisfy the condition.

Example

standard input	standard output
5 2 2 0 2 0 1 3 3 6 4 2 6 7 10 5 4 4 6 5 1 3 1 0 2 3 7 2 7 3 1 4 0 0 1 6 5 5 5 1 7 5 3 1 3 2 7 7 6	1 2 1 0 6

Problem K. Parentheses and Swapping

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

A parentheses sequence is a string composed of the characters `(` and `)`. A parentheses sequence $s_{1,2,\ldots,m}$ is defined as valid if it satisfies one of the following conditions:

- s is empty.
- $s_1 =$ `(`, $s_m =$ `)`, and $s_{2,3,\ldots,m-1}$ is valid.
- There exists a position $k(1 \le k < n)$ such that $s_{1,2,\ldots,k}$ is valid, and $s_{k+1,k+2,\ldots,m}$ is valid.

Two positions $i, j(1 \le i < j \le m)$ in the parentheses sequence $s_{1,2,\ldots,m}$ are said to be matched if and only if both $s_{i,i+1,\ldots,j}$ and $s_{i+1,i+2,\ldots,j-1}$ are valid.

Given an integer sequence of length n, $a_1, a_2, \ldots, a_n$, where n is guaranteed to be even. We construct a new integer sequence $c_1, c_2, \ldots, c_n$ through a valid parentheses sequence $b_1, b_2, \ldots, b_n$, such that $\forall 1 \le i \le n, c_i = a_{p_i}$, where p_i is the position matched with position i in $b_{1,2,\ldots,n}$.

Find a valid parentheses sequence $b_{1,2,\ldots,n}$ such that the lexicographical order of $c_{1,2,\ldots,n}$ is minimized. If there are multiple solutions, output any one of them.

Input

The first line contains an integer $T(1 \le T \le 2.5 \times 10^5)$, indicating the number of test cases.

For each test case:

The first line contains an integer $n(1 \le n \le 5 \times 10^5)$, indicating the length of the sequence a, and it is guaranteed that n is even.

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

It is guaranteed that the sum of n does not exceed 5×10^5.

Output

For each test case, output a line containing a string of length n, representing $b_{1,2,\ldots,n}$.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>5
6
4 1 5 4 1 1
4
1 2 3 2
4
1 3 1 2
2
2 1
8
8 5 2 6 1 4 3 7</pre></td><td><pre>((()))
()()
(())
()
((()))()</pre></td></tr>
</table>

Problem L. Triangle Grid

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 1024 megabytes

Little L has a regular triangular grid, where each grid line is a segment connecting the corresponding n division points of adjacent edges. These grid lines divide the regular triangle into several congruent smaller regular triangles.

For each intersection point of the grid lines in the figure, we represent it with a coordinate (x, y). The x coordinate from top to bottom ranges from 1 to $n+1$. $x = 1$ represents the topmost point, while $x = n+1$ represents the bottom line. For the i-th row, the y coordinates of the grid points from left to right range from 1 to i.

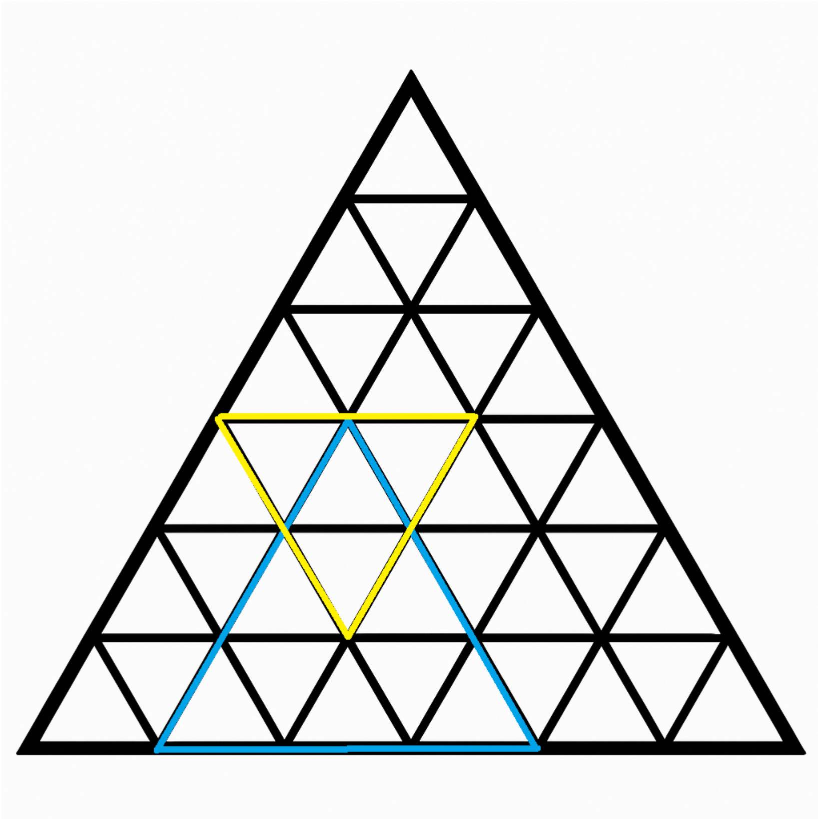
In the triangular grid, there are several regular triangles, which may be oriented upright or inverted. For each orientation of the triangle, we represent it with three numbers x, y, d. Here, (x, y) represents the coordinates of the diagonal vertices of its edge parallel to the grid's bottom line, and d represents its side length (the length of the segment).

For example, in the figure below, the blue regular triangle is an upright triangle, with $(x, y, d) = (4, 2, 3)$. The yellow regular triangle is an inverted triangle, with $(x, y, d) = (6, 3, 2)$.

Now, for each unit small regular triangle grid (a regular triangle with a side length of 1, regardless of orientation), there is a number written on it. Initially, all numbers on the grids are 0. Little L will perform the following operations:

- Choose a regular triangle and add w to all numbers on the grids within it.
- Choose a regular triangle and query the sum of all numbers on the grids within it.

He hopes you can compute the answer for each query operation. Since Little L does not like large numbers, you only need to output the result of the answer modulo 2^{32}.



Input

The first line of input contains two integers n, q ($2 \le n \le 10^5, 1 \le q \le 10^5$).

The next q lines each represent an operation.

Each line starts with five numbers $opt, type, x, y, d$ ($opt \in \{1, 2\}, type \in \{0, 1\}, 1 \le x \le n+1, 1 \le y \le x, 1 \le d \le n$). opt represents the type of operation, where $opt = 1$ is a modification operation, and $opt = 2$ is a query operation. $type$ indicates the type of triangle for this operation, where $type = 0$ is for an upright triangle, and $type = 1$ is for an inverted triangle. x, y, d describe this triangle. If $opt = 1$, there is an additional number w ($0 \le w < 2^{32}$) that indicates the number added during this modification operation.

It is guaranteed that all input triangles have a non-zero area and are valid triangles completely contained within the larger triangle.

Output

For each query with $opt = 2$, output a single number on a new line, representing the sum within the area modulo 2^{32}.

Examples

standard input	standard output
5 5 1 0 3 3 2 2 1 1 6 4 2 3 2 0 1 1 5 2 0 3 2 3 2 1 5 3 2	20 11 3
10 10 1 0 1 1 5 2 1 1 3 2 1 6 1 1 6 4 2 1 2 0 10 3 1 2 1 5 2 1 2 1 6 4 2 1 0 7 7 1 7 1 1 9 3 1 3 1 0 9 5 2 4 2 0 6 1 2	0 2 12 0

Problem M. Replacement

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Given a binary string s of length n (01 string). You need to replace one character in $s[2 \ldots n-1]$ with $\oplus$, so that the final value of the expression is as large as possible.

$\oplus$ represents the binary XOR operation. After the replacement, both sides of $\oplus$ are treated as binary numbers.

Input

A single test case contains multiple data sets.

The first line of input is the number of data sets T ($1 \le T \le 10^5$), representing the number of data sets in this test case.

For each data set, there is one line containing a binary string s ($3 \le |s| \le 5 \cdot 10^5$).

It is guaranteed that in a single test case, $\sum |s| \le 5 \cdot 10^5$.

Output

For each data set, output one line containing a binary string that represents the largest result in binary form (the number string cannot contain leading 0s except for the single digit 0).

Example

standard input	standard output
5 010 0110 1000 10101 01000	0 10 10 100 10