Problem A. Nearest Neighbor Search

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

Bobo has a point p and a cube C in 3-dimension space. The point locates at coordinate (x_0, y_0, z_0), while

$$C = \{(x, y, z) : x_1 \leq x \leq x_2, y_1 \leq y \leq y_2, z_1 \leq z \leq z_2\}.$$

Bobo would like to find another point q which locates inside or on the surface of the cube C so that the square distance between point p and q is minimized.

Note that the square distance between point (x, y, z) and (x', y', z') is $(x - x')^2 + (y - y')^2 + (z - z')^2$.

Input

The first line contains 3 integers x_0, y_0, z_0.

The second line contains 3 integers x_1, y_1, z_1.

The third line contains 3 integers x_2, y_2, z_2.

$(|x_i|, |y_i|, |z_i| \leq 10^4, x_1 < x_2, y_1 < y_2, z_1 < z_2)$

Output

An integer denotes the minimum square distance.

Examples

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

Problem B. Odd Discount

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

In the store of ICPCCamp, there are n items to be sold with m bundles offered.

The i-th bundle is described by c_i and k_i **distinct** integers $a_{i,1}, a_{i,2}, \dots, a_{i,k_i}$. It means that one gets c_i dollars discount if among the $a_{i,1}, a_{i,2}, \dots, a_{i,k_i}$-th items, he buys exactly odd number of them. Bundles can be combined.

Bobo wants to buy a non-empty subset of the items. It is clear there are $(2^n - 1)$ different sets for him. Find out $(d_1^2 + d_2^2 + \cdots + d_{2^n-1}^2)$ modulo $(10^9 + 7)$ where d_i is the sum of discount for the i-th set.

Input

The first line contains 2 integers n, m $(1 \le n \le 20, 1 \le m \le 10^5)$.

The i-th of the following m lines contains integers c_i, k_i, followed by k_i integers $a_{i,1}, a_{i,2}, \dots, a_{i,k_i}$ $(1 \le c_i \le 10^4, 1 \le a_{i,1}, a_{i,2}, \dots, a_{i,k_i} \le n)$.

Output

An integer denotes $(d_1^2 + d_2^2 + \cdots + d_{2^n-1}^2)$ modulo $(10^9 + 7)$.

Examples

standard input	standard output
2 2 1 1 1 2 2 1 2	14
1 1 1 1 1	1

Note

In the first sample, there are 3 possibilities for Bobo.

- He buys the first item and uses both bundles.
- He buys the second item and uses the second bundle solely.
- He buys both items and uses the first bundle.

Therefore, $d_1 = 3, d_2 = 2, d_3 = 1$ and $d_1^2 + d_2^2 + d_3^2 = 14$.

Problem C. Eight Queens

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

In ICPCCamp, there is a chessboard with n rows and m columns.

Bobo places k distinguishable queens in k different cells on the chessboard. There are $t = \binom{n\times m}{k}$ different configurations where

$$\binom{n}{k} = \frac{n(n-1)\dots(n-k+1)}{k(k-1)(k-2)\dots 1}.$$

If c_i is the number of cells attacked by at least one queen in the i-th configuration, find out $(c_1+c_2+\cdots+c_t)$ modulo (10^9+7).

Note that a queen can attack all cells in the same row, column and diagonal including the cell she stands on.

Input

3 integers n, m, k $(1 \le n, m \le 10^9, 1 \le k \le \min\{n \times m, 8\})$.

Output

An integer denotes $(c_1 + c_2 + \cdots + c_t)$ modulo $(10^9 + 7)$.

Examples

standard input	standard output
2 2 2	24
8 8 8	723759469

Problem D. Longest Common Subsequence

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

Bobo learnt how to solve Longest Common Subsequence Problem in ICPCCamp, However, he feels it is too hard for himself and he decides to make an easier one.

The Longest Common Subsequence Problem is to find a longest sequence C which is the subsequence of given sequences A and B. Note that a sequence $A = (a_1, a_2, \dots, a_n)$ is subsequence of sequence $B = (b_1, b_2, \dots, b_m)$ only if there exists $1 \le i_1 < i_2 < \dots < i_n \le m$ where $a_1 = b_{i_1}, a_2 = b_{i_2}, \dots, a_n = b_{i_n}$.

Bobo has a sequence $A = (a_1, a_2, \dots, a_n)$, and a sequence B divided into m consecutive segments. The i-th segment consists of k_i elements $b_{i,1}, b_{i,2}, \dots, b_{i,k_i}$. Bobo is allowed to swap two elements in the same segment for arbitrary number of times. He would like to know the longest common subsequence of A and B after the swaps.

Input

The first line contains 3 integers n, m, l $(1 \le n, m, l \le 3000)$.

The second line contains n integers $a_1, a_2, \dots, a_n$ $(1 \le a_i \le l)$.

The i-th of the following m lines contains an integer k_i followed by k_i integers $b_{i,1}, b_{i,2}, \dots, b_{i,k_i}$ $(k_i \ge 1, 1 \le b_{i,j} \le l, k_1 + k_2 + \dots + k_m \le 3000)$.

Output

An integer denotes the length of the longest common subsequence after the swaps.

Examples

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

Problem E. Coins

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

In ICPCCamp, people usually use coins of value $1, 2, 3$.

Bobo was very poor, he had only a_1, a_2, a_3 coins of value $1, 2, 3$, respectively. He bought an item of an unknown value **without making change**.

The unknown item was of positive integral value. Find out the number of possible values of it.

Input

3 integers a_1, a_2, a_3 $(0 \le a_1, a_2, a_3 \le 10^9)$.

Output

An integer denotes the number of possible values of the unknown item.

Examples

standard input	standard output
`1 0 1`	`3`
`0 0 0`	`0`

Note

In the first sample, Bobo can only buy a item with value $1, 3$ or 4 without making change.

Problem F. Floyd-Warshall

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

In ICPCCamp, there are n cities and m (bidirectional) roads between cities. The i-th road is between the a_i-th city and the b_i-th city. There may be roads connecting a citie to itself and multiple roads between the same pair of cities.

Bobo has q travel plans. The i-th plan is to travel from the u_i-th city to the v_i-th city. He would like to know the smallest number of roads needed to travel for each plan. It is guaranteed that cities are connected.

Input

The first line contains 3 integers n, m, q ($1 \le n \le 10^5, 0 < m - n < 100, 1 \le q \le 10^5$).

The i-th of the following m lines contains 2 integers a_i, b_i ($1 \le a_i, b_i \le n$).

The i-th of the last q lines contains 2 integers u_i, v_i ($1 \le u_i, v_i \le n$).

Output

n lines with integers $l_1, l_2, \dots, l_n$. l_i denotes the smallest number of roads travelling from city u_i to city v_i.

Examples

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

Problem G. Road History

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

Bobo is studying the history of roads in ICPCCamp. In ICPCCamp, there are n cities with m bidirectional roads. The i-th road connects the a_i-th and b_i-th cities.

There were no roads initially. Eventually, roads were built in the order $1, 2, \ldots m$.

Bobo would like to know the number of pairs of cities which allow an *odd drive* after the i-th road was built. An *odd drive* between cities u and v is possible only if there exists $v_1, v_2, \ldots, v_{2k}$ for some positive integers k such that $v_1 = u, v_{2k} = v$ and there is a road connecting cities v_i and v_{i+1}. Passing by a city more than once is allowed.

Input

The first line contains 2 integers n, m $(1 \le n, m \le 10^5)$.

The i-th of the following m lines contains 2 integers a_i, b_i $(1 \le a_i, b_i \le n)$.

Output

m lines with integers $w_1, w_2, \ldots, w_m$ where w_i denotes the number of pairs allowing an odd drive after the i-th road was built.

Examples

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

Problem H. Around the World

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

In ICPCCamp, there are n cities and $(n-1)$ (bidirectional) roads between cities. The i-th road is between the a_i-th and b_i-th cities. It is guaranteed that cities are connected.

Recently, there are $2 \times c_i - 1$ new roads built between the a_i-th and b_i-th cities. Bobo soon comes up with an idea to travel around the world! He plans to start in city 1 and returns to city 1 after traveling along every road exactly once.

It is clear that Bobo has many plans to choose from. He would like to find out the number of different plans, modulo $(10^9 + 7)$.

Note that two plans A and B are considered different only if there exists an i where the i-th traveled road in plan A is different from the i-th road in plan B.

Input

The first line contains an integer n $(2 \le n \le 10^5)$.

The i-th of the following $(n - 1)$ lines contains 3 integers a_i, b_i, c_i $(1 \le a_i, b_i \le n, c_i \ge 1, c_1 + c_2 + \cdots + c_{n-1} \le 10^6)$.

Output

An integer denotes the number of plans modulo $(10^9 + 7)$.

Examples

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>3
1 2 1
2 3 1</pre></td><td><pre>4</pre></td></tr>
<tr><td><pre>3
1 2 1
1 3 2</pre></td><td><pre>144</pre></td></tr>
</table>

Problem I. Longest Increasing Subsequence

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

Bobo is tired of all kinds of hard LIS (Longest Increasing Subsequence) problems, so he decides to make himself some easier one.

Bobo wants to build a sequence of integers $\{x_1, x_2, \dots, x_n\}$, where x_i lies in the range $[a_i, b_i]$ (that is, $a_i \le x_i \le b_i$).

Let $\mathrm{LIS}(X)$ be the length of longest increasing subsequence of $\{x_1, x_2, \dots, x_n\}$. It's clear that $1 \le \mathrm{LIS}(X) \le n$. Bobo would like to find g_k which is the number of sequences whose $\mathrm{LIS}(X) = k$ for $k = 1, 2, \dots, n$.

Note that a sequence $\{i_1, i_2, \dots, i_k\}$ is a increasing sequence of $\{a_1, a_2, \dots, a_n\}$ only if:

- $1 \le i_1 < i_2 < \cdots < i_k \le n$
- $a_{i_1} < a_{i_2} < \cdots < a_{i_k}$

Input

The first line contains an integer n $(1 \le n \le 5)$.

The i-th of the following n lines contains 2 integers a_i, b_i $(1 \le a_i \le b_i \le 10^3)$.

Output

n integers $g_1, g_2, \dots, g_n$.

Examples

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

Problem J. Matrix Transformation

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

Bobo has a matrix of n rows and n columns. The rows are numbered by $0, 1, \dots, (n-1)$ from top to bottom, and the columns are numbered by $0, 1, \dots, (n-1)$ from left to right. The cell in the intersection of the i-th row and the j-th column is denoted as (i, j). For each cell (i, j), there is a number $i \times n + j$ written in.

Bobo is going to perform q transformations successively. The transformations are of 2 kinds. The i-th transformation is of t_i-th kind, and it's described by 3 parameters l_i, r_i, d_i.

If $t_i = 1$, the number in cell $(x, (y + d_i) \bmod n)$ where $l_i \le x \le r_i, 0 \le y < n$ will be transferred to the cell (x, y) by the transformation.

If $t_i = 2$, the number in cell $((x + d_i) \bmod n, y)$ where $0 \le x < n, l_i \le y \le r_i$ will be transferred to the cell (x, y) by the transformation.

Note that $a \bmod b$ means the remainder of a after division by b.

Bobo would like to know the final configuration of the matrix.

Input

The first line contains 2 integers n, q $(1 \le n \le 200, 1 \le q \le 10^5)$.

The i-th of the following q lines contains 4 integers t_i, l_i, r_i, d_i $(t_i \in \{1, 2\}, 0 \le l_i \le r_i < n, 0 \le d_i < n)$.

Output

n lines. The i-th line contains n integers $a_{i,0}, a_{i,1}, \dots, a_{i,n-1}$ which denotes the final number in cell (i, j).

Examples

standard input	standard output
3 2 1 1 1 1 2 1 1 1	0 5 2 4 7 3 6 1 8
3 1 1 0 2 1	1 2 0 4 5 3 7 8 6