Codeforces Round #348 (VK Cup 2016 Round 2, Div. 1 Edition)

A. Little Artem and Matrix

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Artem likes electronics. He can spend lots of time making different schemas and looking for novelties in the nearest electronics store. The new control element was delivered to the store recently and Artem immediately bought it.

That element can store information about the matrix of integers size $n \times m$. There are $n + m$ inputs in that element, i.e. each row and each column can get the signal. When signal comes to the input corresponding to some row, this row cyclically shifts to the left, that is the first element of the row becomes last element, second element becomes first and so on. When signal comes to the input corresponding to some column, that column shifts cyclically to the top, that is first element of the column becomes last element, second element becomes first and so on. Rows are numbered with integers from 1 to n from top to bottom, while columns are numbered with integers from 1 to m from left to right.

Artem wants to carefully study this element before using it. For that purpose he is going to set up an experiment consisting of q turns. On each turn he either sends the signal to some input or checks what number is stored at some position of the matrix.

Artem has completed his experiment and has written down the results, but he has lost the chip! Help Artem find any initial matrix that will match the experiment results. It is guaranteed that experiment data is consistent, which means at least one valid matrix exists.

Input

The first line of the input contains three integers n, m and q ($1 \le n, m \le 100$, $1 \le q \le 10\,000$) — dimensions of the matrix and the number of turns in the experiment, respectively.

Next q lines contain turns descriptions, one per line. Each description starts with an integer t_i ($1 \le t_i \le 3$) that defines the type of the operation. For the operation of first and second type integer r_i ($1 \le r_i \le n$) or c_i ($1 \le c_i \le m$) follows, while for the operations of the third type three integers r_i, c_i and x_i ($1 \le r_i \le n$, $1 \le c_i \le m$, $-10^9 \le x_i \le 10^9$) are given.

Operation of the first type ($t_i = 1$) means that signal comes to the input corresponding to row r_i, that is it will shift cyclically. Operation of the second type ($t_i = 2$) means that column c_i will shift cyclically. Finally, operation of the third type means that at this moment of time cell located in the row r_i and column c_i stores value x_i.

Output

Print the description of any valid initial matrix as n lines containing m integers each. All output integers should not exceed 10^9 by their absolute value.

If there are multiple valid solutions, output any of them.

Examples

input
2 2 6 2 1 2 2 3 1 1 1 3 2 2 2 3 1 2 8 3 2 1 8
output
8 2 1 8

input
3 3 2 1 2 3 2 2 5
output
0 0 0 0 0 5 0 0 0

B. Little Artem and Dance

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Artem is fond of dancing. Most of all dances Artem likes rueda — Cuban dance that is danced by pairs of boys and girls forming a circle and dancing together.

More detailed, there are n pairs of boys and girls standing in a circle. Initially, boy number 1 dances with a girl number 1, boy number 2 dances with a girl number 2 and so on. Girls are numbered in the clockwise order. During the dance different moves are announced and all pairs perform this moves. While performing moves boys move along the circle, while girls always stay at their initial position. For the purpose of this problem we consider two different types of moves:

1. Value x and some direction are announced, and all boys move x positions in the corresponding direction.
2. Boys dancing with even-indexed girls swap positions with boys who are dancing with odd-indexed girls. That is the one who was dancing with the girl 1 swaps with the one who was dancing with the girl number 2, while the one who was dancing with girl number 3 swaps with the one who was dancing with the girl number 4 and so one. It's guaranteed that n is even.

Your task is to determine the final position of each boy.

Input

The first line of the input contains two integers n and q ($2 \le n \le 1\,000\,000$, $1 \le q \le 2\,000\,000$) — the number of couples in the rueda and the number of commands to perform, respectively. It's guaranteed that n is even.

Next q lines contain the descriptions of the commands. Each command has type as the integer 1 or 2 first. Command of the first type is given as x ($-n \le x \le n$), where $0 \le x \le n$ means all boys moves x girls in clockwise direction, while $-x$ means all boys move x positions in counter-clockwise direction. There is no other input for commands of the second type.

Output

Output n integers, the i-th of them should be equal to the index of boy the i-th girl is dancing with after performing all q moves.

Examples

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

input
2 3 1 1 2 1 -2
output
1 2

input
4 2 2 1 3
output
1 4 3 2

C. Little Artem and Random Variable

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Artyom decided to study probability theory. He found a book with a lot of nice exercises and now wants you to help him with one of them.

Consider two dices. When thrown each dice shows some integer from 1 to n inclusive. For each dice the probability of each outcome is given (of course, their sum is 1), and different dices may have different probability distributions.

We throw both dices simultaneously and then calculate values $max(a, b)$ and $min(a, b)$, where a is equal to the outcome of the first dice, while b is equal to the outcome of the second dice. You don't know the probability distributions for particular values on each dice, but you know the probability distributions for $max(a, b)$ and $min(a, b)$. That is, for each x from 1 to n you know the probability that $max(a, b)$ would be equal to x and the probability that $min(a, b)$ would be equal to x. Find any valid probability distribution for values on the dices. It's guaranteed that the input data is consistent, that is, at least one solution exists.

Input

First line contains the integer n ($1 \le n \le 100\,000$) — the number of different values for both dices.

Second line contains an array consisting of n real values with up to 8 digits after the decimal point — probability distribution for $max(a, b)$, the i-th of these values equals to the probability that $max(a, b) = i$. It's guaranteed that the sum of these values for one dice is 1. The third line contains the description of the distribution $min(a, b)$ in the same format.

Output

Output two descriptions of the probability distribution for a on the first line and for b on the second line.

The answer will be considered correct if each value of max(a, b) and min(a, b) probability distribution values does not differ by more than 10^{-6} from ones given in input. Also, probabilities should be non-negative and their sums should differ from 1 by no more than 10^{-6}.

Examples

input
2 0.25 0.75 0.75 0.25
output
0.5 0.5 0.5 0.5

input
3 0.125 0.25 0.625 0.625 0.25 0.125
output
0.25 0.25 0.5 0.5 0.25 0.25

D. Little Artem and Time Machine

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Artem has invented a time machine! He could go anywhere in time, but all his thoughts of course are with computer science. He wants to apply this time machine to a well-known data structure: *multiset*.

Artem wants to create a basic multiset of integers. He wants these structure to support operations of three types:

1. Add integer to the multiset. Note that the difference between set and multiset is that multiset may store several instances of one integer.
2. Remove integer from the multiset. Only one instance of this integer is removed. Artem doesn't want to handle any exceptions, so he assumes that every time remove operation is called, that integer is presented in the multiset.
3. Count the number of instances of the given integer that are stored in the multiset.

But what about time machine? Artem doesn't simply apply operations to the multiset one by one, he now travels to different moments of time and apply his operation there. Consider the following example.

- First Artem adds integer 5 to the multiset at the 1-st moment of time.
- Then Artem adds integer 3 to the multiset at the moment 5.
- Then Artem asks how many 5 are there in the multiset at moment 6. The answer is 1.
- Then Artem returns back in time and asks how many integers 3 are there in the set at moment 4. Since 3 was added only at moment 5, the number of integers 3 at moment 4 equals to 0.
- Then Artem goes back in time again and removes 5 from the multiset at moment 3.
- Finally Artyom asks at moment 7 how many integers 5 are there in the set. The result is 0, since we have removed 5 at the moment 3.

Note that Artem dislikes exceptions so much that he assures that after each change he makes all delete operations are applied only to element that is present in the multiset. The answer to the query of the third type is computed at the moment Artem makes the corresponding query and are not affected in any way by future changes he makes.

Help Artem implement time travellers multiset.

Input

The first line of the input contains a single integer n $(1 \le n \le 100\,000)$ — the number of Artem's queries.

Then follow n lines with queries descriptions. Each of them contains three integers a_i, t_i and x_i $(1 \le a_i \le 3,\ 1 \le t_i, x_i \le 10^9)$ — type of the query, moment of time Artem travels to in order to execute this query and the value of the query itself, respectively. It's guaranteed that all moments of time are distinct and that after each operation is applied all operations of the first and second types are consistent.

Output

For each ask operation output the number of instances of integer being queried at the given moment of time.

Examples

input
6 1 1 5 3 5 5 1 2 5 3 6 5 2 3 5 3 7 5

output
1 2 1

input
3 1 1 1 2 2 1 3 3 1

output
0

E. Little Artem and 2-SAT

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Artem is a very smart programmer. He knows many different difficult algorithms. Recently he has mastered in *2-SAT* one.

In computer science, 2-satisfiability (abbreviated as *2-SAT*) is the special case of the problem of determining whether a conjunction (logical *AND*) of disjunctions (logical *OR*) have a solution, in which all disjunctions consist of no more than two arguments (variables). For the purpose of this problem we consider only *2-SAT* formulas where each disjunction consists of exactly two arguments.

Consider the following *2-SAT* problem as an example: . Note that there might be negations in *2-SAT* formula (like for x_1 and for x_4).

Artem now tries to solve as many problems with *2-SAT* as possible. He found a very interesting one, which he can not solve yet. Of course, he asks you to help him.

The problem is: given two *2-SAT* formulas f and g, determine whether their sets of possible solutions are the same. Otherwise, find any variables assignment x such that $f(x) \neq g(x)$.

Input

The first line of the input contains three integers n, m_1 and m_2 ($1 \le n \le 1000$, $1 \le m_1, m_2 \le n^2$) — the number of variables, the number of disjunctions in the first formula and the number of disjunctions in the second formula, respectively.

Next m_1 lines contains the description of *2-SAT* formula f. The description consists of exactly m_1 pairs of integers x_i (- $n \le x_i \le n, x_i \neq 0$) each on separate line, where $x_i > 0$ corresponds to the variable without negation, while $x_i < 0$ corresponds to the variable with negation. Each pair gives a single disjunction. Next m_2 lines contains formula g in the similar format.

Output

If both formulas share the same set of solutions, output a single word " SIMILAR" (without quotes). Otherwise output exactly n integers x_i () — any set of values x such that $f(x) \neq g(x)$.

Examples

input
2 1 1 1 2 1 2
output
SIMILAR

input
2 1 1 1 2 1 -2
output
0 0

Note

First sample has two equal formulas, so they are similar by definition.

In second sample if we compute first function with $x_1 = 0$ and $x_2 = 0$ we get the result 0, because . But the second formula is 1, because .

F. Little Artem and Graph

time limit per test: 12 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Artem is given a graph, constructed as follows: start with some k-clique, then add new vertices one by one, connecting them to k already existing vertices that form a k-clique.

Artem wants to count the number of spanning trees in this graph modulo $10^9 + 7$.

Input

First line of the input contains two integers n and k ($1 \le n \le 10\,000$, $1 \le k \le min(n, 5)$) — the total size of the graph and the size of the initial clique, respectively.

Next $n - k$ lines describe $k + 1$-th, $k + 2$-th, ..., i-th, ..., n-th vertices by listing k distinct vertex indices $1 \le a_{ij} < i$ it is connected to. It is guaranteed that those vertices form a k-clique.

Output

Output a single integer — the number of spanning trees in the given graph modulo $10^9 + 7$.

Examples

input
3 2 1 2
output
3

input
4 3 1 2 3
output
16