Problem A. Timer

Input file:	`stdin`
Output file:	`stdout`
Time limit:	3 seconds

You discovered a time capsule in your backyard. It has a timer consisting of a single number k and a set of instructions.

The instructions state that each time it is the y_1-th day of the month on the Maya calendar and the y_2-th day of the month on the Azteq calendar, the number on the timer decreases by 1 (it decreases at the very beginning of the day). The capsule will open when the timer becomes 0.

You would like to know in how many days are left until the capsule opens.

How fortunate that you took a minor last year in ancient civilizations not so long ago! You remember that there are T_1 days in a month of the Maya calendar and T_2 days in the Azteq one (yeah, you are in a parallel world). You also know that today is the x_1-th (resp. x_2) day of the month in the Maya (resp. Azteq) calendar .

Input

The input consits of multiple test cases (no more than 10 test cases by input file).
Each test case consists of 7 space separated integers on a single line.
They represent in order: k, x_1, x_2, y_1, y_2, T_1 and T_2.
The input ends by 7 zeros on a single line (see sample input).

The constraints are as follows:

$1 \leq k \leq 10^9$,
$1 \leq x_1, y_1 \leq T_1$,
$1 \leq x_2, y_2 \leq T_2$,
$1 \leq T_1, T_2 \leq 1000$,

Output

For each test case, you output the answer on a single line or -1 if the capsule never opens.
Please use %Ld modifiers or cin/cout to handle 64-bit integers IO in C/C++

Sample input and output

stdin
1 1 3 2 2 2 3 2 1 1 2 2 3 3 1 1 1 1 1 1 1 0 0 0 0 0 0 0
stdout
5 4 1

For the first example, the states on the calendar for each day are:
(1,3), (2,1), (1,2), (2,3), (1,1), (2,2).
On the beginning of the last day (state (2,2)), the timer is decreased to 0 and the capsul opens.
Thus the answer is 5.

Problem B. Cycle detector

Input file:	`stdin`
Output file:	`stdout`
Time limit:	3 seconds

Fox and Lin are bored so they decide to play a game. Initially there is a multigraph (can have more than one edge between the same two vertices and loops) with n vertices and no edges. Lin has to perform k actions.
There are two type of actions. For an action of type 0, Lin chooses two distinct vertices x and y and adds the edge xy to the graph. For an action of type 1, Lin chooses a vertex z and asks Fox whether the connected component containing z is a tree.

You are given each action of Lin in the chronological order, can you please help Fox answer the questions?

Input

The first line contains two integers, the number n of vertices in the graph and the number k of actions to be done by Lin.
Each of the next k lines describes an action which is either of modification on the graph or a question.
A line describing an action of type 0 contains 3 integers. 0 is the first integer and the 2 remaining integers x_i and y_i are distinct and represent the edge to be added to the graph.
A line describing an action of type 1 contains 2 integers. 1 is the first integer and the second integer z_i (the z in the statement).

The constraints are as follows:
$2 \leq n \leq 10^4$,
$1 \leq k \leq 10^4$,
$1 \leq x_i, y_i, z_i \leq n$, $x_i \neq y_i$

Output

For each question, output "1" (without the quotes) on a new line if the connected component is a tree and "0" otherwise.

Sample input and output

stdin
2 4 0 1 2 1 1 0 1 2 1 2
stdout
1 0

At the time of the first question, there is a single edge in the graph so it is a tree.
For the second question there is two edges between the only two vertices of the graph. It is clearly not a tree, so you should output "0".

Problem C. How much is that?

Input file:	`stdin`
Output file:	`stdout`
Time limit:	3 seconds

How convenient, a very short problem statement for once :)
Find the number of connected graph with n vertices numbered from 1 to n and m edges.
Since the answer can be pretty huge, output it modulo $10^9 + 7$.

Input

The input consists of multiple test cases. Each test case is described on a single line by two integers n and m, the number of vertices and the number of edges of the graph.
The input end by a line with two zeros separated by a space.

The constraints are as follows:
$1 \leq n \leq 100$,
$0 \leq m \leq 100$

Output

Output one line for each test case consisting of the answer.

Sample input and output

stdin
3 2 2 1 0 0
stdout
3 1

For the first instance, the three graphs are: $\{(1,2),(2,3)\}, \{(1,2),(1,3)\}$ and $\{(2,3),(1,3)\}$

Problem D. Bad divisor

Input file:	stdin
Output file:	stdout
Time limit:	6.0 seconds

Little Petya really hates some number z. He defines the score of an integer x as the number of time z divides such integer. More formally, the score of x is the largest integer y for which z^y divides x.

Today little Petya is on the top-left cell (coordinates $(1, 1)$) of a grid with r rows and c columns with a number inside each cell. He can only move one cell down or one cell to the left, without ever going outside the grid. He keeps moving until he reaches the bottom-right cell (coordinates (r, c)) of the grid.

When he is on the bottom-right cell, he is given a number which is the product of every integer he has been on during his epic journey.

Help him find the minimum score he can achieve.

Input

On the first line, 3 integers representing r, c and z. Each of the next r lines contains c integers. The $j-th$ integer a_{ij} of the $i-th$ line corresponds to the number is the cell (i, j) of the grid.

The constraints are as follows: $1 \leq r, c \leq 1000$,
$1 \leq z \leq 5 \times 10^4$,
$0 \leq a_{ij} \leq 10^{18}$,

Please use %Ld modifiers or cin/cout to handle 64-bit integers IO in C/C++

Output

Print a single integer, the minimum score achievable for little Petya. If the answer is not defined (goes to infinity), print "-1".

Sample input and output

stdin
2 2 2 2 6 4 5
stdout
2

Little Petya starts on the cell with the number 2 and should end on the cell with number 5.

In order to minimize his score, he should go to the left at the beginning, visiting the cell with number 6. Like this, he will be given the integer 40 which has a score 2.

If he went down at the beginning, he would have had a score of 3.

Problem E. Race day in Berland

Input file:	`stdin`
Output file:	`stdout`
Time limit:	3 seconds

It is race day in a famous Berland city. The city has n junctions and $n-1$ roads connecting these junctions. As any well designed city, you can reach any junction from any other by taking a succession of roads.

Race day consists of k races which takes place during a single day. The $i-th$ race starts at the junction x_i, ends at the junction y_i and c_i cars will be participating in it. These cars will start on x_i and they will have to go to the junction y_i using some roads.

Sadly today is also the day when the road network reparations start. One road has to be rebuilt thus we have to cancel every race which cannot avoid using this road. The mayor has asked you to pick the road such that the maximum number of drivers can still run today.

Input

The first line contains n, the number of junctions in the city.
Each of the next $n-1$ lines describes a road. The i-th line contains 2 integers a_i and b_i, the two endpoints of the i-th road. The next line contains a single integer k, the number of races.
Each of the next k lines describes a race.
The i-th line contains 3 integers, x_i, y_i and c_i.

The constraints are as follows:
$2 \leq n \leq 1 \times 10^5$,
$1 \leq a_i, b_i \leq n, a_i \neq b_i$,
$0 \leq k \leq 2 \times 10^5$,
$1 \leq x_i, y_i \leq n, x_i \neq y_i$,
$1 \leq c_i \leq 2 \times 10^3$

Output

Print two numbers on a single line separated by a space, the maximum number of drivers who will be able to run and the id of the road which should be rebuilt today (the id is the number of line where the road was described, 1-indexed and starting from the first road).

If there is more than one possible road, choose the one with the lowest id.

Sample input and output

stdin
5 1 3 4 1 3 5 2 3 3 2 5 7 2 4 5 4 2 5
stdout
10 3

The road to rebuild is between the junction number 3 and 5.