Problem A. Timer

During a recent expedition, Indiana Bones discovered an old Mazteq temple. Inside the temple he found a working timer consisting of a single number k.

The instructions next to the timer 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). Moreover if the timer reaches 0 the world ends immediately, how original!.

You have been asked to find out how many days are left until the end of the world.

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. You also know that it is today 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:

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

Output

Output the answer on a single line.
Please use cout or %Ld to print 64 integers 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, so it is the end of the world.
Thus there is 5 days left until the end of the world.

Problem B. Moving along

For this problem only, we will use the term graph instead of multigraph (can have more than one edge between the same 2 vertices).
Fox is playing a game alone. He starts on a vertex of a graph (undirected) and move along its edges. He would like to know whether or not it is possible to move indefinitely. This game would be trivial without any restrictions so Fox has decided that he cannot move along an edge if it was also used during the previous move.
After a few games he found the game too easy and decided to call his friend Lin to play the game with him.
Lin plays the gamemaster. Initially he chooses the number of vertices n of the graph on which they will play (the graph has originally no edges).
Then Lin has to perform a number k of actions. One action can be to either add an edge between two vertices of the graph (even if it already exists) or ask Fox to solve the game on the graph from a vertex of Lin's choice. By solving we mean deciding whether or not it is possible to move indefinitely from the starting position.

Fox has only a short time to give his answer, can you help him?

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 a modification 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 a question contains 2 integers. 1 is the first integer and the second integer z_i is the starting vertex at which Fox has to answer if he can play indefinitely or not.

The constraints are as follows:
$1 \leq n \leq 10^3$,
$1 \leq k \leq 10^3$,
$1 \leq x_i, y_i, z_i \leq n$, $x_i \neq y_i$

Output

For each question asked, print on a new line "Possible" if Fox can play indefinitely and "Impossible" otherwise.

Sample input and output

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

stdout
Impossible Possible

For the first question there is a single edge in the graph so Fox clearly cannot move infinitely.
For the second question there is two edges between the only two vertices of the graph. Fox can alternate between these two edges to move infinitely along the graph without violating the restriction on his movements.

Problem C. How much is that?

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 one 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 to the problem asked.

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

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 and a number is inside each cell. His only move allowed are one cell down or one cell 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 of the integer he can possibly get.

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 cell (i, j) of the grid.

The constraints are as follows: $1 \leq r, c \leq 2000$,
$1 \leq z \leq 5 \times 10^4$,
$1 \leq a_{ij} \leq 10^18$,

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 right 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

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 on the junction a_i, ends on the junction b_i and c_i cars will be participating in it. These cars will start on a_i and they will have to go to the junction b_i using some roads.

Sadly today is also the day when the road network renewal starts. 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 contains 2 integers: $1 \le a \le n$ and $1 \le b \le n$, $a \ne b$ representing a pair of junctions connected by a direct 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, a_i, b_i and c_i.

The constraints are as follows:
$2 \le n \le 2 \times 10^5$,
$1 \le a_i, b_i \le n$,
$1 \le k \le 2 \times 10^5$,
$1 \le c_i \le 2 \times 10^3$

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.