Problem A. Timer

Input file: `timer.in`
Output file: `timer.out`

Recently during an expedition Indiana Bones discovered an old Mazteq ruins. Inside the temple he found a ingenious timer consisting of a single number k. The instructions next to the timer state that each time it is day y_1 on the Maya calendar and day y_2 on the Azteq calendar the number on the timer decreases by 1 (it decreases at the very beginning of the day). Moreover it states that if the timer reaches 0, the world ends, 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 in ancient civilizations not so long ago! You have no doubt that the following integers are going to help you find the answer: T_1, T_2, x_1 and x_2 which represent respectively the number of days in the Maya calendar, the number of days in the Azteq calendar, the current day in the Maya calendar and finally the current day in the Azteq calendar.

Input

The input consists of the 7 integers described in the statement: k, y_1, y_2, T_1, T_2, x_1 and x_2.
The constraints are as follows: $1 \leq k \leq 10^9$,
$1 \leq x_1, y_1 \leq T_1$,
$1 \leq x_2, y_2 \leq T_1$,
$1 \leq T_1, T_2 \leq 1000$,

Output

Output the answer on a single line.
Please use cout or %Ld to print 64 integers in C/C++.

Sample input and output

timer.in
0 1 12
timer.out
ans

Problem B. Moving along

Input file:	`movalong.in`
Output file:	`movalong.out`

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 is not allowed to 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 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.

There is a countdown so Fox has to be very quick solving the game, can you help him?

Input

On the first line a single integer n, the number of vertices in the graph.

The next line contains the number k of operations/questions asked by Lin (in order).

The next k lines describe either an operation or a question.

A line describing an operation 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.

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

movalong.in
`0 1 12`
movalong.out
`ans`

Problem C. How much is that?

Input file:	`howmuch.in`
Output file:	`howmuch.out`

Oh, 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: $1 \leq n, m \leq 100$,

Output

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

Sample input and output

howmuch.in
0 1 12
howmuch.out
ans

Problem D. Bad divisor

Input file:	`divisor.in`
Output file:	`divisor.out`

Little Petya really hates some number z. He defines the score of an integer 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. He can only move either down and to the right without ever going outside the grid. He keeps moving until he reaches the bottom-right cell (coordinates (r, c)) of the grid. When he finishes, he is given as number the product of every integer he has been on during his epic journey. Help him find the minimum score possible of the number given to him.

Input

On the first line, the 3 integers r, c and z has described in the statement. 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: $1 \leq r, c \leq 2000$,
$1 \leq z \leq 50000$,
$1 \leq a_{ij} \leq 10^18$,

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

divisor.in
`0 1 12`
divisor.out
`ans`

Problem E. Race day in Berland

Input file:	`raceday.in`
Output file:	`raceday.out`

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 will be participating in it.

However today is also 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. 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 $(1 \le a_i, b_i \le n)$ and c_i.

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

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 where the work should take place (the id is the number of line where the road was described, 1-indexed).

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

Sample input and output

raceday.in
raceday.out