Problem A. Topological sorting

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You are given a directed graph. Find it's topological sorting.

Input

First line contains two integers n and m — number of vertices and edges in the graph, respectively ($1 \le n \le 100\,000, m \le 100\,000$).

Next m lines describe edges of the graph. Each line contains two integers v and u — describing the edge starting at v and ending at u ($1 \le v, u \le n$; $v \ne u$).

Output

If no topological sorting exists, output `-1`.

Otherwise output the sequence of vertices which describes the topological ordering. If several orderings exist, output any.

Example

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

Problem B. Hamiltonian path

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You are given directed acyclic graph. Find whether graph contains path passing through all the vertices.

Input

First line contains two integers n and m — number of vertices and edges of graph, respectively. Next m lines contain description of graph edges. Edge i is described by two integers b_i and e_i — starting and ending vertices, respectively ($1 \le b_i, e_i \le n$).

Given graph doesn't contain loops or cycles.

$1 \le n \le 100\,000$, $0 \le m \le 200\,000$.

Output

If the graph contains hamiltonian path output `YES`. Otherwise, output `NO`.

Examples

standard input	standard output
3 3 1 2 1 3 2 3	YES
3 2 1 2 1 3	NO

Problem C. Shortest path

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You are given directed acyclic graph. You are to find the weight of shortest path from vertex s to t.

Input

First line contains four integers n, m, s and t — number of vertices and edges of graph, starting and ending vertex of a path, respectively. Next m lines contain description of graph edges. Edge i is described by three integers b_i, e_i and w_i — starting vertex, ending vertex and an edge weight, respectively ($1 \le b_i, e_i \le n$; $|w_i| \le 1000$).

Given graph doesn't contain loops or cycles.

$1 \le n \le 100\,000$, $0 \le m \le 200\,000$.

Output

First line should contain single integer — the weight of shortest path from s to t.

If there is no path from s to t, output `Unreachable`.

Examples

standard input	standard output
2 1 1 2 1 2 -10	-10
2 1 2 1 1 2 -10	Unreachable

Problem D. Milk scheduling

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Farmer John's n cows are conveniently numbered $1 \dots n$. Each cow i takes t_i units of time to milk. Unfortunately, some cows must be milked before others, owing to the layout of FJ's barn. If cow A must be milked before cow B, then FJ needs to completely finish milking A before he can start milking B.

In order to milk his cows as quickly as possible, FJ has hired a large number of farmhands to help with the task — enough to milk any number of cows at the same time. However, even though cows can be milked at the same time, there is a limit to how quickly the entire process can proceed due to the constraints requiring certain cows to be milked before others. Please help FJ compute the minimum total time the milking process must take.

Input

The first line of input contains number n and m, number of cows and constrains($1 \le n \le 10\,000, 1 \le m \le 50\,000$). Next n lines contain time to milk for each cow, ($1 \le t_i \le 100\,000$). Next m lines contain a and b — description of constrains($1 \le a, b \le n$).

Output

Output the minimum amount of time required to milk all cows.

Example

standard input	standard output
3 1 10 5 6 3 2	11

Problem E. Graph condensation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Find the number of edges in a condensation of a given oriented graph. Note: condensation doesn't have multiedges.

Input

The first line of the input file contains two integers n and m — the number of vertices and the number of edges respectively ($n \leq 10\,000,\ m \leq 100\,000$). Next m lines contain edges description, one line describes one edge. Edge number i is represented by two numbers b_i, e_i — the start and the end of the edge respectively($1 \leq b_i,\ e_i \leq n$). Graph may have multiedges and loops.

Output

Print one integer —the number of edges in a condensation of the graph.

Example

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

Problem F. Firesafe

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

The Marks city has n houses, some of them are connected by unidirectional roads.

There are many fires in Marks lately, so citizens decided to build several firefighting stations. But the following problem occured: when a car goes to fight the fire, it can omit the directions of the roads, but after fighting the fire, the car must obey the directions of the roads (the citizens of Marks city worship these rules!).

It is obvious that wherever firefighting car occurs, it must have the possibility of returning back to the firefighting station from which it departed. However, the construction of such stations is expensive, so it was decided by the government of the city to build the minimal number of stations. Also it was decided to build the stations such that each of them is connected to some house.

Your task is to find the optimal location of firefighting stations.

Input

In the first line there is one integer n ($1 \le n \le 3\,000$). The next line contains one integer m-the number of roads ($1 \le m \le 100\,000$). Next lines contain the description of roads in format a_i, b_i meaning that i-th road allows to move from house a_i to house b_i ($1 \le a_i, b_i \le n$).

Output

In the first line, print the minimal number K of the firefighting stations, which we need to build. In the second line print K numbers in any order — houses, near to which the stations will be built. If there are multiple solutions, you can print any of them.

Example

standard input	standard output
5 7 1 2 2 3 3 1 2 1 2 3 3 4 2 5	2 4 5

Problem G. Air travel

Input file:	avia.in
Output file:	avia.out
Time limit:	2 seconds
Memory limit:	256 megabytes

The chief designer Petya was asked to develop a new aircraft model for the company "Air Bubundia". It turned out that the most difficult part is the selection of the optimal size of the fuel tank.

The chief cartographer of "Air Bubundia" Vasya made a detailed map of Bubundia. On this map, he noted the fuel consumption for the flight between each pair of cities.

Petya wants to make the size of the tank as small as possible, such that the plane can fly from any city to any other (possibly with refueling on the way).

Input

First line contains integer n ($1 \le n \le 1000$) — number of cities in Bubundia.

Next n lines contain n integers each. j-th number in i-th line equals to the consumption of fuel for the flight from i-th city to j-th city. All number are non-negative and doesn't exceed 10^9. i-th number in i-th row equals to zero.

Output

Output single integer — optimal tank size.

Example

avia.in	avia.out
4 0 10 12 16 11 0 8 9 10 13 0 22 13 10 17 0	10

Problem H. Stones

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

There're n stones on the table. Two players are playing, making alternating turns. In one turn a player can:

- 1 or 2 stones, if n is dividing by 3;
- 1 or 3 stones, if it gives remainder 1;
- 1, 2 or 3 stones, if it gives remainder 2.

Every turn can be made only if there're enough stones. The player who can't make move is losing.

Input

First line contains single integer n ($0 < n \leq 100$).

Output

Output number 1 or 2 — index of the player who will win if both play optimally.

Examples

standard input	standard output
1	1
2	1
6	2

Problem I. Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Two Maxwell demons play the next game. At the beginning of the game, there is a certain amount of elementary particles. Each particle belongs to one of four types: A, B, C or D. During its turn, each demon can select some particles so that they enter exactly one reaction and collapse. The moves are taken in turn until a situation arises in which it is impossible to choose a nonempty set of particles in this way. In this case, the first demon who cannot make a move loses.

The following reactions are possible (for each reaction, the particles participating in it are indicated; the result of each reaction is the mutual destruction of the particles participating in it).

- $A + A + B + D + D$
- $A + B + C + D$
- $C + C + D$
- $B + B + B$
- $A + D$

Every turn should be one of the above reactions.

Determine who will win the game.

Input

First line contains integer n — the number of test cases ($1 \le n \le 5$). Every of the next n lines contains a description of one test case. It contains four integers $n_a,\ n_b,\ n_c,\ n_d$ — number of particles of types A, B, C and D respectively ($0 \le n_a, n_b, n_c, n_d \le 30$).

Output

For every test case output in new line "`First`", if first demon will win, and "`Second`" otherwise.

Example

standard input	standard output
5 0 3 0 3 2 5 0 4 7 7 7 7 9 9 7 8 10 10 10 10	First First First First Second