Section: B1 + B2 **Duration: 1 hour + 5 mins (submission)**

Use of STL/library functions is allowed for both problems.

Problem 1

Suppose you are given the floorplan of a building. The floorplan is represented by a rectangular grid, and each cell is either a floor or a wall.

Your task is to calculate the number of rooms on the floor. A room is defined as a maximal connected region of floor cells such that you can walk left, right, up, and down through the floor squares. See sample 2 for better understanding.

Input

The first input line has two integers n and m: the height and width of the grid. (n, m <= 1000)
After this there are n lines of m integers describing the grid. Each integer is either 0 or 1:
0: Floor
1: Wall

You can safely assume that the floor is always surrounded by walls.

Output

Print the number of rooms on the floor.

Sample Input	Sample Output	Explanation
5 8 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1	1	1 1 1 1 1 1 1 1 1 **0 0 0 0 0 0** 1 1 **0 0 0 0 0 0** 1 1 **0 0 0 0 0 0** 1 1 1 1 1 1 1 1 1
5 8 1 1 1 1 1 1 1 1 1 0 0 1 0 0 0 1 1 1 1 1 0 1 0 1 1 0 0 1 0 0 0 1 1 1 1 1 1 1 1 1	3	1 1 1 1 1 1 1 1 1 **0 0** 1 **0 0 0** 1 1 1 1 1 **0** 1 **0** 1 1 **0 0** 1 **0 0 0** 1 1 1 1 1 1 1 1 1 The red-colored cells belong to the same room, although there is a wall in between them.

Problem 2

You have to complete n courses. There are m requirements of the form "course a has to be completed before course b". Your task is to find an order in which you can complete the courses.

Input

The first input line has two integers n and m: the number of courses and requirements. The courses are numbered from 1 to n. (n, m <= 10^5)
After this, there are m lines describing the requirements. Each line has two integers a and b: course a has to be completed before course b.

Output

Print an order in which you can complete the courses. You can print any valid order that includes all the courses.
If there are no solutions, print -1.

Sample Input	Sample Output	Explanation
5 3 1 2 3 1 4 5	3 4 1 5 2	
6 6 1 2 2 3 4 3 4 5 5 6 6 4	-1	The dependency among courses 4, 5 and 6 cannot be resolved.

Submission Guidelines

1. Create a new folder and name it with your student ID (e.g. 2305001).
2. Copy **only the cpp/java/python files** to the newly created folder.
3. Rename your individual code files as **<ID_ProblemX>.<cpp/java/py>**. For example, if your student ID is 2305001, then for problem 1, the cpp/java/py file must be named 2305001_Problem1.<cpp/java/py>.
4. Zip the folder and name the zip file with your student ID (e.g. 2305001.zip).
5. **Submit the zip file only.**
6. Any violation of these instructions will result in a penalty.