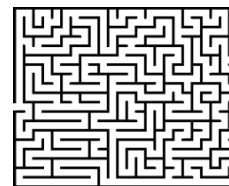


1 Finding an Exit from a Maze

Problem Introduction

A maze is a rectangular grid of cells with walls between some of adjacent cells. You would like to check whether there is a path from a given cell to a given exit from a maze where an exit is also a cell that lies on the border of the maze (in the example shown to the right there are two exits: one on the left border and one on the right border). For this, you represent the maze as an undirected graph: vertices of the graph are cells of the maze, two vertices are connected by an undirected edge if they are adjacent and there is no wall between them. Then, to check whether there is a path between two given cells in the maze, it suffices to check that there is a path between the corresponding two vertices in the graph.



Problem Description

Task. Given an undirected graph and two distinct vertices u and v , check if there is a path between u and v .

Input Format. An undirected graph with n vertices and m edges. The next line contains two vertices u and v of the graph.

Constraints. $2 \leq n \leq 10^3$; $1 \leq m \leq 10^3$; $1 \leq u, v \leq n$; $u \neq v$.

Output Format. Output 1 if there is a path between u and v and 0 otherwise.

Time Limits.

language	C	C++	Java	Python	C#	Haskell	JavaScript	Ruby	Scala
time (sec)	1	1	1.5	5	1.5	2	5	5	3

Memory Limit. 512MB.

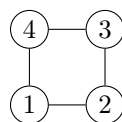
Sample 1.

Input:

```
4 4
1 2
3 2
4 3
1 4
1 4
```

Output:

```
1
```



In this graph, there are two paths between vertices 1 and 4: 1-4 and 1-2-3-4.

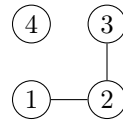
Sample 2.

Input:

```
4 2
1 2
3 2
1 4
```

Output:

```
0
```



In this case, there is no path from 1 to 4.

Starter Files

The starter solutions for this problem read the input data from the standard input, pass it to a blank procedure, and then write the result to the standard output. You are supposed to implement your algorithm in this blank procedure if you are using C++, Java, or Python3. For other programming languages, you need to implement a solution from scratch. Filename: **reachability**

What To Do

To solve this problem, it is enough to implement carefully the corresponding algorithm covered in the lectures.