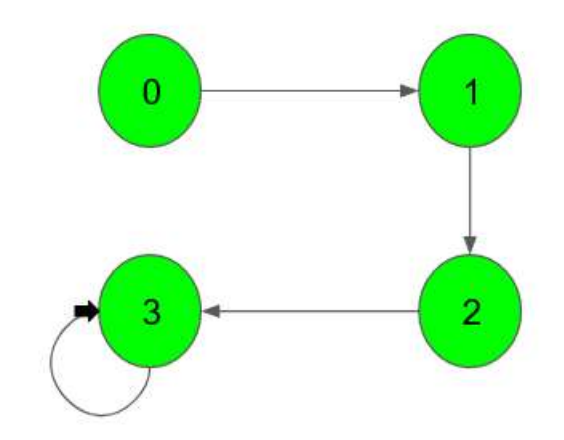
Detect cycle in a directed graph

Given a Directed Graph with **V** vertices (Numbered from **0** to **V-1**) and **E** edges, check whether it contains any cycle or not.

Example 1:

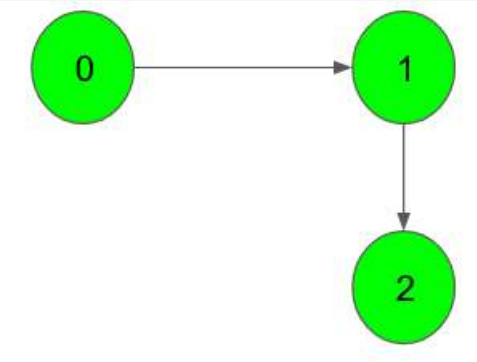
Input:



Output: 1
Explanation: 3 -> 3 is a cycle

Example 2:

Input:



```
Output: 0
Explanation: no cycle in the graph
```

Your task:
You dont need to read input or print anything. Your task is to complete the function **isCyclic()** which takes the integer V denoting the number of vertices and adjacency list as input parameters and returns a boolean value denoting if the given directed graph contains a cycle or not.

Expected Time Complexity: O(V + E)
Expected Auxiliary Space: O(V)

Constraints:
$1 \leq V, E \leq 10^5$

Input Format:

First line of the custom input must contain two space separated integers V denoting the number of vertices and E denoting the number of edges. Next E lines contains two space-separated integers denoting each edge.

Example:

```
4 4
0 1
1 2
2 3
3 3
```