Detect Cycle

Problem Statement

This challenge is part of a tutorial track by MyCodeSchool

You're given the pointer to the head node of a linked list. Find whether the list contains any cycle (or loop). A linked list is said to contain cycle if any node is *re-visited* while traversing the list. The head pointer given may be null meaning that the list is empty.

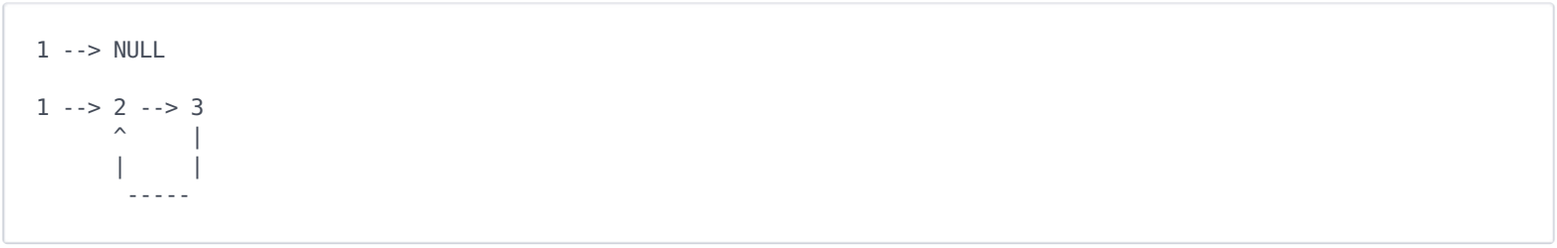
Input Format

You have to complete the `int HasCycle(Node* head)` method which takes one argument - the head of the linked list. You should NOT read any input from stdin/console. Number of nodes in a linked list doesn't exceed 100.

Output Format

Check whether the linked list has a cycle and `return 1` if there is a cycle. Otherwise, `return 0`. Do NOT print anything to stdout/console.

Sample Input



Sample Output

```
0
1
```

Explanation

1. First list has no cycle, hence return 0
2. Second list is shown to have a cycle, hence return 1.

Note

After first solving the problem by yourself, see Floyd's cycle-finding algorithm for an efficient solution which uses O(N) time and O(1) additional memory.