Practice Compete Jobs Rank Leaderboard

Dashboard > Data Structures > Linked Lists > Cycle Detection

Your Cycle Detection submission got 5.00 points. Share Tweet

Try the Next Challenge|Try a Random Challenge

# Cycle Detection

by harsha_s

Problem Submissions Leaderboard Discussions Editorial

A linked list is said to contain a *cycle* if any node is visited more than once while traversing the list.

Complete the function provided for you in your editor. It has one parameter: a pointer to a *Node* object named $head$ that points to the head of a linked list. Your function must return a boolean denoting whether or not there is a cycle in the list. If there *is* a cycle, return *true*; otherwise, return *false*.

**Note:** If the list is empty, $head$ will be *null*.

**Input Format**

Our hidden code checker passes the appropriate argument to your function. You are not responsible for reading any input from stdin.

**Constraints**

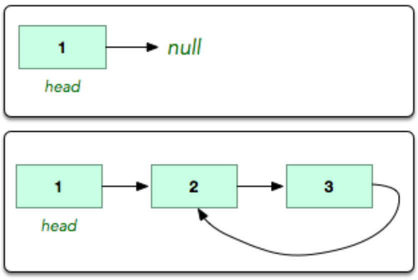
- $0 \leq$ *list size* $\leq 100$

**Output Format**

If the list contains a cycle, your function must return *true*. If the list *does not* contain a cycle, it must return *false*. The binary integer corresponding to the boolean value returned by your function is printed to stdout by our hidden code checker.

**Sample Input**

The following linked lists are passed as arguments to your function:



**Sample Output**

```
0
1
```

**Explanation**

1. The first list has no cycle, so we return *false* and the hidden code checker prints $0$ to stdout.
2. The second list has a cycle, so we return *true* and the hidden code checker prints $1$ to stdout.

**Submissions:**38305

**Max Score:**5

**Difficulty:** Medium

**Rate This Challenge:**

☆☆☆☆☆

More

**Current Buffer** (saved locally, editable)

C++14

```
/*
Detect a cycle in a linked list. Note that the head pointer may be 'NULL' if the list is empty.

A Node is defined as:
    struct Node {
        int data;
        struct Node* next;
    }
*/

bool has_cycle(Node* head) {
    // Complete this function
    // Do not write the main method
    struct Node *step1=head;
    struct Node *step2=head;
    if (head == NULL)
        return 0;
    while(step1->next!=NULL && step2->next->next!=NULL)
        {
            if(step1->next == step2->next->next)
                return 1;

            step1=step1->next;
            step2=step2->next->next;
        }
    //if(step1->next == step2->next->next)
    //            return 1;
    return 0;



}

```

Line: 27 Col: 7

Upload Code as File

□ **Test against custom input**

Run Code

Submit Code

## Congrats, you solved this challenge!

✓ Test Case #0

✓ Test Case #1

**You've earned 5.00 points!**

Next Challenge

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature