

LAB(2-12-2025)

INPUT:

```
1 bool hasCycle(struct ListNode *head) {
2     if (head == NULL || head->next == NULL)
3         return false;
4
5     struct ListNode *slow = head;
6     struct ListNode *fast = head;
7
8     while (fast != NULL && fast->next != NULL) {
9         slow = slow->next;
10        fast = fast->next->next;
11
12        if (slow == fast)
13            return true;
14    }
15
16    return false;
17 }
```

Example 1:

Input: head = [3,2,0,-4], pos = 1
Output: true
Explanation: There is a cycle in the linked list, where the tail connects to the 1st node (0-indexed).

Example 2:

Input: head = [1,2], pos = 0
Output: true
Explanation: There is a cycle in the linked list, where the tail connects to the 0th node.

Example 3:

Input: head = [1]
Output: false
Explanation: There is no cycle in the linked list.

OUTPUT:

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

```
head =
[3,2,0,-4]
```

pos =
1

Output

```
true
```

Expected

```
true
```

Problem List < >  Premium

Testcase > Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

```
head = [1,2]
```

pos = 0

Output

```
true
```

Expected

```
true
```

 Contribute a testcase

Problem List < >  Premium

Testcase > Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

```
head = [1]
```

pos = -1

Output

```
false
```

Expected

```
false
```

 Contribute a testcase