

3. Linked List Cycle

(LeetCode 3)

Code:

class Solution {

public:

```
bool hasCycle(ListNode *head) {  
    if (head == NULL) return false;
```

```
    ListNode *slow = head;
```

```
    ListNode *fast = head;
```

```
    while (fast != NULL && fast->next != NO
```

```
        slow = slow->next;
```

```
        fast = fast->next->next;
```

```
        if (slow == fast) {
```

```
            return true;
```

}

3

```
return true;
```

g

g

```
return false;
```

3

g;

Output:

case 1

Input:

head =

[3, 2, 0, -4]

pos =

1

Output:

true

Expected:

true

case 2

Input:

head =

[1, 2]

pos =

0

Output:

true

Expected:

true

case 3

Input:

head =

[1]

pos =

-1

Output:

false

Expected:

false