

Linked List Cycle

The screenshot shows a code editor interface with the following details:

- Language:** C
- Compiler:** Auto
- Code:** A C function named `hasCycle` that takes a pointer to a `ListNode` struct as input. The function uses two pointers, `temp1` and `temp2`, both initialized to `head`. It then enters a loop where it moves `temp1` one step forward and `temp2` two steps forward until they meet or one reaches the end of the list.
- Test Result:** Accepted, Runtime: 3 ms
- Test Cases:** Case 1, Case 2, Case 3 (all passed)
- Input:** `head = [3, 2, 0, -4]`
- Output:** `true`
- Expected:** `true`

The screenshot shows a code editor interface with the following details:

- Language:** C
- Compiler:** Auto
- Code:** A C function named `hasCycle` that takes a pointer to a `ListNode` struct as input. The function uses two pointers, `temp1` and `temp2`, both initialized to `head`. It then enters a loop where it moves `temp1` one step forward and `temp2` two steps forward until they meet or one reaches the end of the list.
- Test Result:** Accepted, Runtime: 3 ms
- Test Cases:** Case 1, Case 2, Case 3 (all passed)
- Input:** `head = [1, 2]`
- Output:** `true`
- Expected:** `true`

C ✓ Auto

```
7  */
8 bool hasCycle(struct ListNode *head) {
9     struct ListNode *temp1, *temp2;
10    temp1=head;
11    temp2=head;
12    do {
```

Saved

Testcase | Test Result

Accepted Runtime: 3 ms

Case 1 Case 2 Case 3

Input

```
head =
[1]
```

pos =
-1

Output

```
false
```

Expected

```
false
```