class Solution {

public:

bool hasCycle(ListNode \*head) {

if (head == nullptr || head->next == nullptr) {

return false; // No cycle possible if list is empty or has only one node

}

ListNode \*slow = head;

ListNode \*fast = head;

while (fast != nullptr && fast->next != nullptr) {

slow = slow->next; // Slow pointer moves one step

fast = fast->next->next; // Fast pointer moves two steps

if (slow == fast) {

return true; // Cycle detected!

}

}

return false; // No cycle found

}

};