Checking if a linked list is circular linked list or not!

Approach:

- We need to check if a given linked list is circular, meaning the last node should point back to some previous node in the list instead of NULL
- Traverse the list by starting from the first node till the last one.
- If we reach the head node again during traversal, the list is circular. If we reach NULL, the list is not circular

Program:

```
#include <stdio.h>
#include <stdlib.h>
struct Node {
    int data;
    struct Node* next;
};
struct Node* createNode(int data);
// Function to check if the linked list is circular
int isCircular(struct Node* head) {
    // If head is null, list is empty, circular
    if (!head) return 1;
    struct Node* temp = head;
    // Traverse until the end is reached or next node
    // equals the head
    while (head && head->next != temp)
        head = head->next;
    // If end reached before finding head again, list is
    // circular
    if (!head || !(head->next))
        return 0;
    return 1;
}
```

```
struct Node* createNod(int data){
    struct Node* newNode = malloc(sizeof(struct Node));
    newNode -> data = data;
    newNode -> next = NULL;
    return newNode;
}

int Main(){
    struct Node* head = createNode(1);
    head -> next = createNode(2);
    head -> next -> next = createNode(3);
    head -> next -> next -> next = createNode(4);

    // check if the Linked list is Circular
    isCircular(head) ? printf("Yes\n") : printf("No\n");

    // Making the linked list circular
    head -> next -> next -> next -> next = head;

    // check again if the linked list is circular
    isCircular(head) ? printf("Yes\n") : printf("No\n");
    return 0;
}
```

o/p:

No

Yes

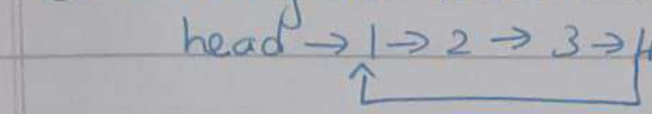
Working of Code:

① creating a node:
* A linked list is created with four nodes

head → 1 → 2 → 3 → 4 → null

② checking if the linked list is Circular or not
* The function isCircular(head) is called
* Since head is not NULL, we proceed
* A temporary pointer temp is set to head
* We start traversing
* The traversal reaches NULL, meaning list is not
* Prints "No"

③ Making the list Circular



④ checking again if the list is Circular:
* This time the traversal reaches head again, t
The list is Circular
* The function returns 1, and "Yes" is printed