Delete Head of Circular Linked List

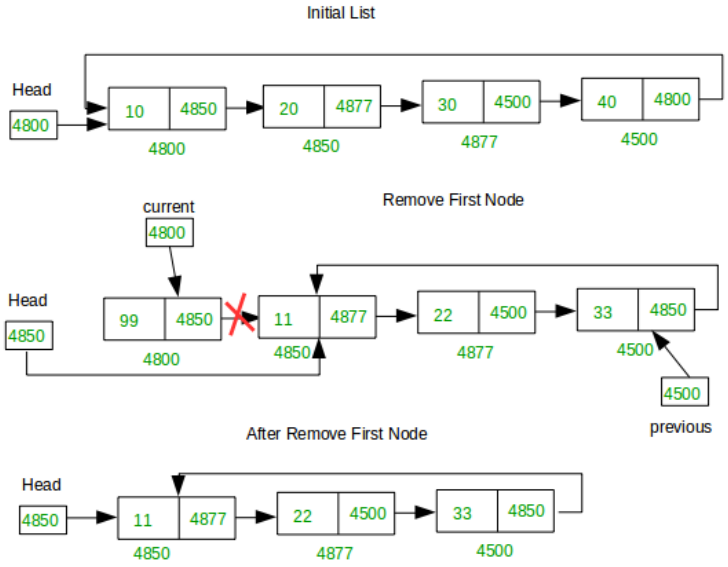
Given a Circular Linked List. The task is to write program to delete first node from Singly Circular Linked List

Deleting first node from Singly Circular Linked List

Examples:

```
Input: 99->11->22->33->44->55->66
Output: 11->22->33->44->55->66

Input: 11->22->33->44->55->66
Output: 22->33->44->55->66
```



Deleting First Node from Circular Linked List

Approach:

1. Take two pointers current and previous and traverse the list.
2. Keep the pointer current fixed pointing to the first node and move previous until it reaches the last node.
3. Once, the pointer previous reaches the last node, do the following:

- previous->next = current-> next
- head = previous -> next;

Function to delete first node from singly circular linked list:

```
// Function to delete First node of
// Circular Linked List
void DeleteFirst(struct Node** head)
{
    struct Node *previous = *head, *firstNode = *head;

    // check if list doesn't have any node
    // if not then return
    if (*head == NULL) {
        printf("\nList is empty\n");
        return;
    }

    // check if list have single node
    // if yes then delete it and return
    if (previous->next == previous) {
        *head = NULL;
        return;
    }

    // traverse second node to first
    while (previous->next != *head) {

        previous = previous->next;
    }

    // now previous is last node and
    // first node(firstNode) link address
    // put in last node(previous) link
    previous->next = firstNode->next;

    // make second node as head node
    *head = previous->next;
    free(firstNode);
```

```
    return;
}
```