Experiment No. 7: Circular Linked List Operations

Aim: Implementation of Circular Linked List ADT

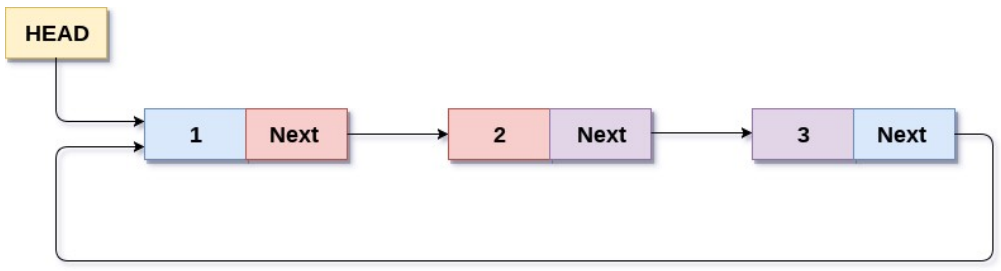
Objective: Circular Linked Lists can be used to manage the computing resources of the computer. Data structures such as stacks and queues are implemented with the help of the circular linked lists

Theory :

In a circular Singly linked list, the last node of the list contains a pointer to the first node of the list. We can have circular singly linked list as well as circular doubly linked list.

We traverse a circular singly linked list until we reach the same node where we started. The circular singly liked list has no beginning and no ending. There is no null value present in the next part of any of the nodes.

The following image shows a circular singly linked list.



Circular Singly Linked List

Algorithm

Algorithm :

```
o  Step 1: IF PTR = NULL
        Write OVERFLOW
       Go to Step 11
       [END OF IF]
```

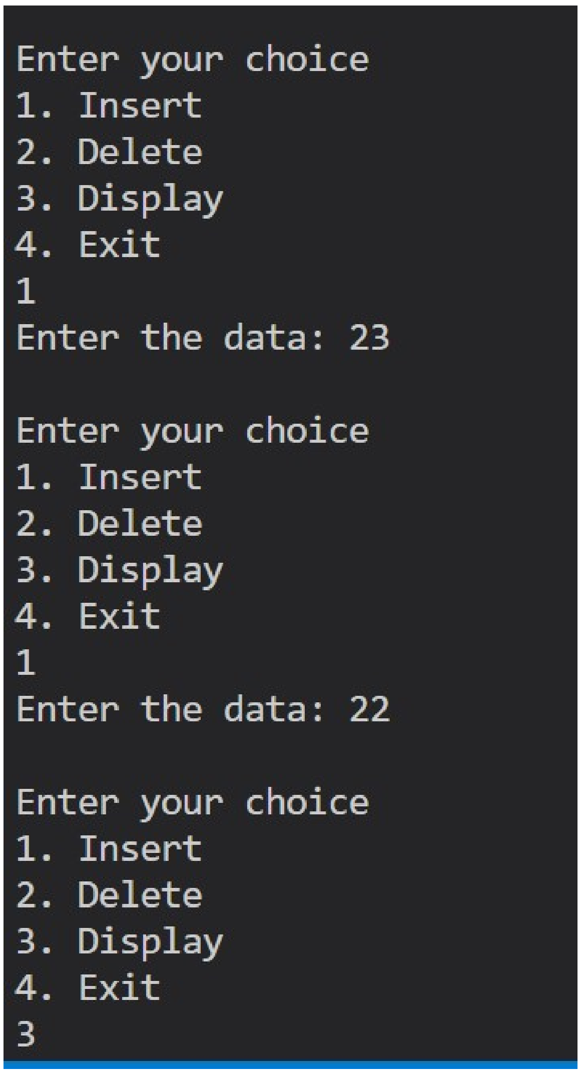
- **Step 2:** SET NEW_NODE = PTR
- **Step 3:** SET PTR = PTR -> NEXT
- **Step 4:** SET NEW_NODE -> DATA = VAL
- **Step 5:** SET TEMP = HEAD
- **Step 6:** Repeat Step 8 while TEMP -> NEXT != HEAD
- **Step 7:** SET TEMP = TEMP -> NEXT

  [END OF LOOP]
- **Step 8:** SET NEW_NODE -> NEXT = HEAD
- **Step 9:** SET TEMP → NEXT = NEW_NODE
- **Step 10:** SET HEAD = NEW_NODE
- **Step 11:** EXIT

Code:
```
#include <stdio.h>
#include <stdlib.h>

struct node
{
   int data;
   struct node *next;
};

struct node *newnode, *head, *temp;

void insert()
{
   newnode = (struct node *)malloc(sizeof(struct node));
```

```
    printf("Enter the data: ");
    scanf("%d", &newnode->data);
    newnode->next = 0;

    if (head == 0)
    {
        head = newnode;
        newnode->next = head;
    }
    else
    {
        temp = head;
        while (temp->next != head)
        {
            temp = temp->next;
        }
        temp->next = newnode;
        newnode->next = head;
    }
}

void display()
{
    int count = 0;
    temp = head;
    printf("Elements in the circular linked list are: ");
    do
    {
        printf("%d\t", temp->data);
        temp = temp->next;
        count++;
    } while (temp != head);
    printf("\nNumber of nodes: %d\n", count);
}

void delete()
{
    if (head == 0)
    {
        printf("No element to delete from the node\n");
    }
    else
    {
        temp = head;
        while (temp->next != head)
        {
            temp = temp->next;
        }
        temp->next = head->next;
        printf("Deleted node data is: %d\n", head->data);
        free(head);
        head = temp->next;
    }
}

int main()
{
    int ch;
    while (1)
    {
        printf("\nEnter your choice\n1. Insert\n2. Delete\n3. Display\n4. Exit\n");
        scanf("%d", &ch);
        switch (ch)
        {
        case 1:
            insert();
            break;
        case 2:
            delete();
            break;
        case 3:
            display();
            break;
        case 4:
            exit(0);
            break;
        default:
            printf("Invalid choice\n");
            break;
        }
    }
    return 0;
}
```

Output:



```
Enter your choice
1. Insert
2. Delete
3. Display
4. Exit
1
Enter the data: 23

Enter your choice
1. Insert
2. Delete
3. Display
4. Exit
1
Enter the data: 22

Enter your choice
1. Insert
2. Delete
3. Display
4. Exit
3
```

Conclusion: In a circular Singly linked list, the last node of the list contains a pointer to the first node of the list. We can have circular singly linked list as well as circular doubly linked list.