

Name: Tanzeela Asghar

Reg no: 2021 BSE 032

Section no: III-A

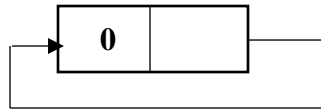
Course: Data Structure

Submitted to: Sir Rehan Ahmed Siddiqui

Lab#10

Task 1:

1. Draw a circular linked list of integers with a single node having a value 0.



2. Consider the following two circular linked lists with pointers on their last nodes. Write C++ statements to merge the two lists into one single list with contents POWERPOINT.

//Suppose list 1 head=head1 and pointer pointing last node of list 1 is *p and list 2 head=head2 pointer pointing last node of list 2 is *q

```
p->next=head2;  
q->next=head1;
```

Code task 1

Implement the (class) Circular Linked List to create a list of integers. You need to provide the implementation of the member functions as described in the following.

CODE:

```
#include "stdafx.h"  
#include<iostream>  
using namespace std;  
class CList  
{  
    struct node  
    {  
        int data;  
        node *next;  
    }*head;  
public:  
    CList()  
    {  
        head=NULL;  
    }  
    bool empty_list()  
    {  
        if(head==NULL)  
            return true;  
        else  
            return false;  
    }  
}
```

```

void insert (int pos, int value)
{
    int count=1;
    node *p,*q;
    p=new node;
    p->data=value;
    q=head;
    while(count!=pos)
    {
        count++;
        q=q->next;
    }
    if(count==pos)
    {
        p->next=q->next;
        q->next=p;
    }
}

void insert_begin(int value)
{
    node *p,*q;
    p=new node;
    p->data=value;
    if(empty_list())
    {
        head=p;
        head->next=head;
    }
    else
    {
        q=head;
        while(q->next!=head)
        {
            q=q->next;
        }
        p->next=head;
        head=p;
        q->next=p;
    }
}

void insert_end(int value)
{

```

```

        node *p,*q;
        p=new node;
        p->data=value;
        p->next=head;
        q=head;
        if(empty_list())
        {
            head=p;
            head->next=head;
        }
        else
        {
            while(q->next!=head)
            {
                q=q->next;
            }
            q->next=p;
        }
    }
void delete_begin()
{
    node *q;
    q=head;
    if(empty_list())
    {
        cout<<"List is empty...\n";
    }
    else
    {
        while(q->next!=head)
        {
            q=q->next;
        }
        q->next=head->next;
        delete head;
        head=q->next;
    }
}
void delete_end()
{
    node *q,*p;
    p=head;
    if(empty_list())

```

```

        {
            cout<<"List is empty...\n";
        }
        else
        {
            while(p->next!=head)
            {
                q=p;
                p=p->next;
            }
            q->next=p->next;
            delete p;
        }
    }
    void traverse()
    {
        node *p;
        p=head;
        while(p->next!=head)
        {
            cout<<p->data<<" ";
            p=p->next;
        }
        if(p->next==head)
        {
            cout<<p->data;
        }
        cout<<endl;
    }
};

int _tmain(int argc, _TCHAR* argv[])
{
    CList cl1;
    cout<<"Inserting new nodes at the begining of the list:\n";
    cl1.insert_begin(6);
    cl1.insert_begin(8);
    cl1.insert_begin(7);
    cl1.traverse();
    cout<<"Inserting new nodes at the end of the list:\n";
    cl1.insert_end(0);
    cl1.insert_end(9);
    cl1.traverse();
    cout<<"Adding node after specfic position:\n";

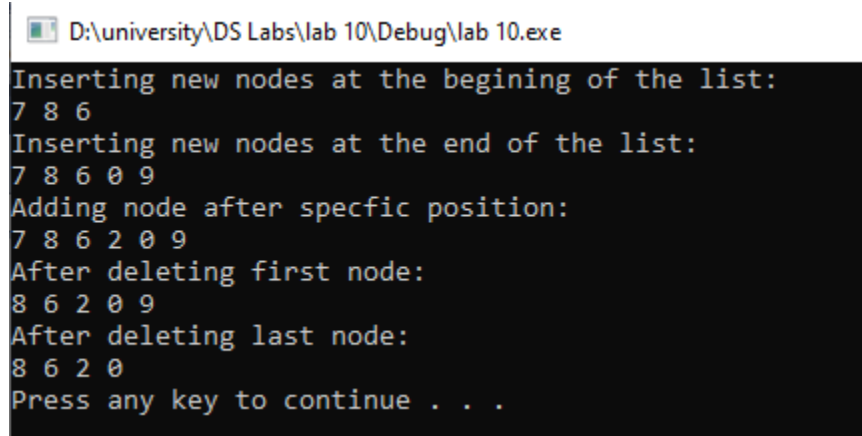
```

```

        cl1.insert(3,2);
        cl1.traverse();
        cout<<"After deleting first node:\n";
        cl1.delete_begin();
        cl1.traverse();
        cout<<"After deleting last node:\n";
        cl1.delete_end();
        cl1.traverse();
        system("pause");
        return 0;
}

```

OUTPUT:



```

D:\university\DS Labs\lab 10\Debug\lab 10.exe
Inserting new nodes at the beginning of the list:
7 8 6
Inserting new nodes at the end of the list:
7 8 6 0 9
Adding node after specfic position:
7 8 6 2 0 9
After deleting first node:
8 6 2 0 9
After deleting last node:
8 6 2 0
Press any key to continue . . .

```