DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

Implement a program to check if the link list is a palindrome

```
#include<iostream>
#include<stdlib.h>
using namespace std;

struct node
{
    int data;
    struct node *next;
}
*list=NULL,*p,*s,*q,*r;  //*p is used for new node

class LinkPal
{
public:
int choice,value;

void get()
{
    do
            {
                cout<<"0.Exit\n1.Add Elemnet \n2.Check if
Pallindrome \n3.Display \n";
                cout<<"Enter Your Choice : "<<" ";
                cin>>choice;
                switch(choice)
                {
                    case 0:
                    break;
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
                    case 1:
                        insert();
                        break;

                    case 2:
                        check();
                        break;

                    case 3:
                        display();
                        break;

                    default:
                        cout<<"invalid input"<<endl<<endl;
                }
            }while(choice!=0);

}

void insert()
{
            cout<<"Enter the value : ";
            cin>>value;
            p=(struct node*)malloc(sizeof(node));
            p->data=value;
            if(list == NULL)
            {
                p->next=NULL;
                list=p;
                display();
            }
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
            else
            {
                q=list;
                while(q->next != NULL)
                {
                    q=q->next;
                }
                q->next=p;
                p->next=NULL;
                display();
            }
}

int count_ele()
{
    int c=0;
    p=list;
    while(p != NULL)
    {
        p=p->next;
        c++;
    }
    return c;
    // cout<<"The Number of Elements is : "<<c<<endl<<endl;
}


void check()
{
            if(list == NULL)
            {
                cout<<endl<<"List is Empty "<<endl<<endl;
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
            }
        else
        {
            int flag = count_ele()/2;
            p=list;
            q=NULL;
            while(flag > 0)
            {
                r=list;
                while(r->next != q)
                {
                    r=r->next;
                }
                if(p->data == r->data)
                {
                    p = p->next;
                    q=r;
                    flag--;
                }
                else
                    flag = -1;
            }
            if(flag == -1)
                cout<<"The List is a Pallindrome";
            else
                cout<<"The List is not a Pallindrome";
        }
        cout<<endl<<endl;
}

void display()
{
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
    if(list==NULL)
            {
                cout<<endl<<"List is Empty "<<endl<<endl;
            }
            else
            {
                cout<<"The List is : ";
                q=list;
                while(q !=NULL)
                {
                    cout<<q->data<<"|----->";
                    q=q->next;
                }
                cout<<endl<<endl;
            }
}


};

int main()
{
    LinkPal s;
    s.get();
    return 0;
}
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

Output :

```
C:\Users\gupta\Desktop\DS Practical\DS CHALLENGING PROGRAM>g++ linkpal.cpp -o lp.exe

C:\Users\gupta\Desktop\DS Practical\DS CHALLENGING PROGRAM>lp.exe
0.Exit
1.Add Elemnet
2.Check if Pallindrome
3.Display
Enter Your Choice :  1
Enter the value : 10
The List is : 10|----->

0.Exit
1.Add Elemnet
2.Check if Pallindrome
3.Display
Enter Your Choice :  1
Enter the value : 20
The List is : 10|----->20|----->

0.Exit
1.Add Elemnet
2.Check if Pallindrome
3.Display
Enter Your Choice :  2
The List is a Pallindrome

0.Exit
1.Add Elemnet
2.Check if Pallindrome
3.Display
Enter Your Choice :  0

C:\Users\gupta\Desktop\DS Practical\DS CHALLENGING PROGRAM>
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

Implement a program of double stack using single array

```
#include<iostream>
#include<stdlib.h>
using namespace std;

class SingleArrayDoubleStack
{
public:
int choice,value,ltop,rtop,size,rElement,count;
int arr[100];

SingleArrayDoubleStack()
{
    ltop=-1;
    cout<<"Enter Size of Array (Less than 100) : ";
    cin>>size;
    cout<<endl;

    for(int i=0;i<size;i++)
    {
        arr[i]=0;
    }

    rtop=size;
}

void get()
{
    do
            {
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
                cout<<"0.Exit\n01.Push an Element ( Left Side )\n02.Pop an Element ( Left Side )\n03.Push an Element ( Right Side )\n04.Pop an Element ( Right Side )\n05.Display\n";
                cout<<"Enter Your Choice : "<<" ";
                cin>>choice;
                switch(choice)
                {
                    case 0:
                    break;

                    case 1:
                        lpush();
                        break;

                    case 2:
                        lpop();
                        break;

                    case 3:
                        rpush();
                        break;

                    case 4:
                        rpop();
                        break;

                    case 5:
                        display();
                        break;

                    default:
                        cout<<"invalid input"<<endl<<endl;
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
                }
            }while(choice!=0);

}

bool isEmptyr()
{
    if(rtop==size)
    {
        return true;
    }
    else
    {
        return false;
    }
}

bool isFullr()
{
    if(arr[rtop-1] != 0)
    {
        return true;
    }
    else
    {
        return false;
    }
}

bool isEmptyl()
{
    if(ltop==-1)
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
    {
        return true;
    }
    else
    {
        return false;
    }
}

bool isFulll()
{
    if(arr[ltop+1] != 0)
    {
        return true;
    }
    else
    {
        return false;
    }
}

void lpush()
{

    cout<<"Enter value : ";
    cin>>value;
    cout<<endl;
    if(isFulll())
    {
        cout<<"Sorry OverFlowwwww "<<endl;
    }
    else
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
    {
    ltop=ltop+1;
    arr[ltop]=value;
    }
    display();
    cout<<endl;
}

void lpop()
{
    cout<<endl;
    if(isEmptyl())
    {
        cout<<"Sorry UnderFlow "<<endl;
    }
    else
    {
        rElement=arr[ltop];
        arr[ltop]=0;
        cout<<"The removed Element : "<<rElement<<" from
Position : "<<ltop<<endl;
        ltop=ltop-1;
    }
    display();
    cout<<endl;
}

void rpush()
{

    cout<<"Enter value : ";
    cin>>value;
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
    cout<<endl;
    if(isFullr())
    {
        cout<<rtop<<" ";
        cout<<"Sorry OverFlow "<<endl;
    }
    else
    {
    rtop=rtop-1;
    arr[rtop]=value;
    }
    display();
    cout<<endl;
}

void rpop()
{
    cout<<endl;
    if(isEmptyr())
    {
        cout<<"Sorry UnderFlow "<<endl;
    }
    else
    {
        rElement=arr[rtop];
        arr[rtop]=0;
        cout<<"The removed Element : "<<rElement<<" from
Position : "<<rtop<<endl;
        rtop=rtop+1;
    }
    display();
    cout<<endl;
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
}

void display()
{
    cout<<"The Elements in Stack are :"<<endl;
    for(int i=0;i<size;i++)
    {
        cout<<"Position : "<<i<<" value : "<<arr[i]<<endl;
    }
}

};

int main()
{
    SingleArrayDoubleStack d;
    d.get();
    return 0;
}
```

Output :

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

Command Prompt - lp.exe

```
Microsoft Windows [Version 10.0.18363.815]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\gupta>cd C:\Users\gupta\Desktop\DS Practical\DS CHALLENGING PROGRAM

C:\Users\gupta\Desktop\DS Practical\DS CHALLENGING PROGRAM>g++ linkpal.cpp -o lp.exe

C:\Users\gupta\Desktop\DS Practical\DS CHALLENGING PROGRAM>lp.exe
0.Exit
1.Add Elemnet
2.Check if Pallindrome
3.Display
Enter Your Choice :  1
Enter the value : 10
The List is : 10|----->

0.Exit
1.Add Elemnet
2.Check if Pallindrome
3.Display
Enter Your Choice :  1
Enter the value : 20
The List is : 10|----->20|----->

0.Exit
1.Add Elemnet
2.Check if Pallindrome
3.Display
Enter Your Choice :  2
The List is a Pallindrome

0.Exit
1.Add Elemnet
2.Check if Pallindrome
3.Display
Enter Your Choice :
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
C:\Users\gupta\Desktop\DS Practical\DS CHALLENGING PROGRAM>g++ stack.cpp -o stack.exe

C:\Users\gupta\Desktop\DS Practical\DS CHALLENGING PROGRAM>stack.exe
Enter Size of Array (Less than 100) : 5

0.Exit
01.Push an Element ( Left Side )
02.Pop an Element ( Left Side )
03.Push an Element ( Right Side )
04.Pop an Element ( Right Side )
05.Display
Enter Your Choice :  1
Enter value : 10

The Elements in Stack are :
Position : 0 value : 10
Position : 1 value : 0
Position : 2 value : 0
Position : 3 value : 0
Position : 4 value : 0

0.Exit
01.Push an Element ( Left Side )
02.Pop an Element ( Left Side )
03.Push an Element ( Right Side )
04.Pop an Element ( Right Side )
05.Display
Enter Your Choice :  1
Enter value : 20

The Elements in Stack are :
Position : 0 value : 10
Position : 1 value : 20
Position : 2 value : 0
Position : 3 value : 0
Position : 4 value : 0

0.Exit
01.Push an Element ( Left Side )
02.Pop an Element ( Left Side )
03.Push an Element ( Right Side )
04.Pop an Element ( Right Side )
05.Display
Enter Your Choice :  3
Enter value : 32
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
The Elements in Stack are :
Position : 0 value : 10
Position : 1 value : 20
Position : 2 value : 0
Position : 3 value : 0
Position : 4 value : 32

0.Exit
01.Push an Element ( Left Side )
02.Pop an Element ( Left Side )
03.Push an Element ( Right Side )
04.Pop an Element ( Right Side )
05.Display
Enter Your Choice :  3
Enter value : 54

The Elements in Stack are :
Position : 0 value : 10
Position : 1 value : 20
Position : 2 value : 0
Position : 3 value : 54
Position : 4 value : 32

0.Exit
01.Push an Element ( Left Side )
02.Pop an Element ( Left Side )
03.Push an Element ( Right Side )
04.Pop an Element ( Right Side )
05.Display
Enter Your Choice :  1
Enter value : 55

The Elements in Stack are :
Position : 0 value : 10
Position : 1 value : 20
Position : 2 value : 55
Position : 3 value : 54
Position : 4 value : 32

0.Exit
01.Push an Element ( Left Side )
02.Pop an Element ( Left Side )
03.Push an Element ( Right Side )
04.Pop an Element ( Right Side )
05.Display
Enter Your Choice :  1
Enter value : 20
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
Sorry OverFlowwwww
The Elements in Stack are :
Position : 0 value : 10
Position : 1 value : 20
Position : 2 value : 55
Position : 3 value : 54
Position : 4 value : 32

0.Exit
01.Push an Element ( Left Side )
02.Pop an Element ( Left Side )
03.Push an Element ( Right Side )
04.Pop an Element ( Right Side )
05.Display
Enter Your Choice :  2

The removed Element : 55 from Position : 2
The Elements in Stack are :
Position : 0 value : 10
Position : 1 value : 20
Position : 2 value : 0
Position : 3 value : 54
Position : 4 value : 32

0.Exit
01.Push an Element ( Left Side )
02.Pop an Element ( Left Side )
03.Push an Element ( Right Side )
04.Pop an Element ( Right Side )
05.Display
Enter Your Choice :  2

The removed Element : 20 from Position : 1
The Elements in Stack are :
Position : 0 value : 10
Position : 1 value : 0
Position : 2 value : 0
Position : 3 value : 54
Position : 4 value : 32
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
01.Push an Element ( Left Side )
02.Pop an Element ( Left Side )
03.Push an Element ( Right Side )
04.Pop an Element ( Right Side )
05.Display
Enter Your Choice :  4

The removed Element : 54 from Position : 3
The Elements in Stack are :
Position : 0 value : 10
Position : 1 value : 0
Position : 2 value : 0
Position : 3 value : 0
Position : 4 value : 32

0.Exit
01.Push an Element ( Left Side )
02.Pop an Element ( Left Side )
03.Push an Element ( Right Side )
04.Pop an Element ( Right Side )
05.Display
Enter Your Choice :  4

The removed Element : 32 from Position : 4
The Elements in Stack are :
Position : 0 value : 10
Position : 1 value : 0
Position : 2 value : 0
Position : 3 value : 0
Position : 4 value : 0

0.Exit
01.Push an Element ( Left Side )
02.Pop an Element ( Left Side )
03.Push an Element ( Right Side )
04.Pop an Element ( Right Side )
05.Display
Enter Your Choice :  2

The removed Element : 10 from Position : 0
The Elements in Stack are :
Position : 0 value : 0
Position : 1 value : 0
Position : 2 value : 0
Position : 3 value : 0
Position : 4 value : 0
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

Implement a program of Priority Queue

```
#include<iostream>
#include<stdlib.h>
using namespace std;

class PriorityQueue
{
public:
int choice,value,rear,front,rElement,count,size;
int arr[100],priority[100];

PriorityQueue()
{
    front=0;
    rear=-1;
    cout<<"Enter Size of Array (Less than 100) : ";
    cin>>size;
    cout<<endl;

    for(int i=0;i<size;i++)
    {
        arr[i]=0;
    }

    for(int i=0;i<size;i++)
    {
        priority[i]=111;
    }
}
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
void get()
{
    do
            {
                cout<<"0.Exit\n01.Enter an Element\n02 Delete an
Element\n03.Display\n";
                cout<<"Enter Your Choice : "<<" ";
                cin>>choice;
                switch(choice)
                {
                    case 0:
                    break;

                    case 1:
                        insert();
                        break;

                    case 2:
                        delete_ele();
                        break;

                    case 3:
                        display();
                        break;

                    default:
                        cout<<"invalid input"<<endl<<endl;
                }
            }while(choice!=0);

}
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
bool isEmpty()
{
    if(front>rear)
    {
        return true;
    }
    else
    {
        return false;
    }
}

bool isFulll()
{
    if(rear>=size-1)
    {
        return true;
    }
    else
    {
        return false;
    }
}

void insert()
{
    cout<<"Enter value : ";
    cin>>value;
    cout<<endl;
    if(isFulll())
    {
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
        cout<<"Sorry OverFlowwwww "<<endl;
    }
    else
    {
    rear=rear+1;
    arr[rear]=value;
    cout<<"Enter priority : ";
    cin>>priority[rear];
    cout<<endl;
    }
    display();
    cout<<endl;
}

void delete_ele()
{
    int lowest_priority,index_lowest_priority;
    cout<<endl;
    if(isEmpty())
    {
        cout<<"Sorry UnderFlow "<<endl;
    }
    else
    {
        lowest_priority=priority[0];
        for(int z=0;z<size;z++)
        {
            if(lowest_priority >= priority[z])
            {

                    lowest_priority=priority[z];
                    index_lowest_priority=z;
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
            }
        }
        rElement=arr[index_lowest_priority];
        arr[index_lowest_priority]=0;
        priority[index_lowest_priority]=111;
        cout<<"The removed Element : "<<rElement<<" from
Position : "<<index_lowest_priority<<endl;
        front=front+1;
    }
    display();
    cout<<endl;
}

void display()
{
    cout<<"The Elements in Queue are :"<<endl;
    for(int i=0;i<size;i++)
    {
        cout<<"Position : "<<i<<"    value : "<<arr[i]<<"
priority : "<<priority[i]<<endl;
    }
}

};

int main()
{
    PriorityQueue d;
    d.get();
    return 0;
}
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

Output :

```
C:\Users\gupta\Desktop\DS Practical\DS CHALLENGING PROGRAM>g++ priorityqueue.cpp -o pq.ex

C:\Users\gupta\Desktop\DS Practical\DS CHALLENGING PROGRAM>pq.exe
Enter Size of Array (Less than 100) : 5

0.Exit
01.Enter an Element
02 Delete an Element
03.Display
Enter Your Choice :  1
Enter value : 10

Enter priority : 1

The Elements in Queue are :
Position : 0    value : 10    priority : 1
Position : 1    value : 0    priority : 111
Position : 2    value : 0    priority : 111
Position : 3    value : 0    priority : 111
Position : 4    value : 0    priority : 111

0.Exit
01.Enter an Element
02 Delete an Element
03.Display
Enter Your Choice :  1
Enter value : 20

Enter priority : 2

The Elements in Queue are :
Position : 0    value : 10    priority : 1
Position : 1    value : 20    priority : 2
Position : 2    value : 0    priority : 111
Position : 3    value : 0    priority : 111
Position : 4    value : 0    priority : 111
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
0.Exit
01.Enter an Element
02 Delete an Element
03.Display
Enter Your Choice :  1
Enter value : 30

Enter priority : 5

The Elements in Queue are :
Position : 0    value : 10    priority : 1
Position : 1    value : 20    priority : 2
Position : 2    value : 30    priority : 5
Position : 3    value : 0    priority : 111
Position : 4    value : 0    priority : 111

0.Exit
01.Enter an Element
02 Delete an Element
03.Display
Enter Your Choice :  1
Enter value : 65

Enter priority : 8

The Elements in Queue are :
Position : 0    value : 10    priority : 1
Position : 1    value : 20    priority : 2
Position : 2    value : 30    priority : 5
Position : 3    value : 65    priority : 8
Position : 4    value : 0    priority : 111

0.Exit
01.Enter an Element
02 Delete an Element
03.Display
Enter Your Choice :  1
Enter value : 32

Enter priority : 0
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
The Elements in Queue are :
Position : 0    value : 10    priority : 1
Position : 1    value : 20    priority : 2
Position : 2    value : 30    priority : 5
Position : 3    value : 65    priority : 8
Position : 4    value : 32    priority : 0

0.Exit
01.Enter an Element
02 Delete an Element
03.Display
Enter Your Choice :  2

The removed Element : 32 from Position : 4
The Elements in Queue are :
Position : 0    value : 10    priority : 1
Position : 1    value : 20    priority : 2
Position : 2    value : 30    priority : 5
Position : 3    value : 65    priority : 8
Position : 4    value : 0    priority : 111

0.Exit
01.Enter an Element
02 Delete an Element
03.Display
Enter Your Choice :  2

The removed Element : 10 from Position : 0
The Elements in Queue are :
Position : 0    value : 0    priority : 111
Position : 1    value : 20    priority : 2
Position : 2    value : 30    priority : 5
Position : 3    value : 65    priority : 8
Position : 4    value : 0    priority : 111

0.Exit
01.Enter an Element
02 Delete an Element
03.Display
Enter Your Choice :  2
```

DS PRACTICE QUESTION

Name : Abhishek Gupta **UID No : 2019450017**

```
The removed Element : 20 from Position : 1
The Elements in Queue are :
Position : 0    value : 0    priority : 111
Position : 1    value : 0    priority : 111
Position : 2    value : 30    priority : 5
Position : 3    value : 65    priority : 8
Position : 4    value : 0    priority : 111

0.Exit
01.Enter an Element
02 Delete an Element
03.Display
Enter Your Choice :  0

C:\Users\gupta\Desktop\DS Practical\DS CHALLENGING PROGRAM>
```