Practical 04

Abhishek Gupta 2019450017

4.1 Queue_17.cpp

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

class Queue
{
public:
int choice,value,fr,rear,size,rElement;
int arr[100];

Queue()
{
    rear=fr=-1;
    for(int i=0;i<100;i++)
    {
        arr[i]=0;
    }

    cout<<"Enter Size of Queue (Less than 100) : ";
    cin>>size;
    cout<<endl;
}

void get()
{
    do
            {
                cout<<"0.Exit\n01.Push an Element\n02.Pop an
Element\n03.Display\n";
                cout<<"Enter Your Choice : "<<" ";
```

```
                cin>>choice;
                switch(choice)
                {
                    case 0:
                    break;

                    case 1:
                        push();
                        break;

                    case 2:
                        pop();
                        break;

                    case 3:
                        display();
                        break;

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

}

bool isEmpty()
{
    if(rear == -1 && fr == -1)
    {
        return true;
    }
    else
```

Queue

```
    {
        return false;
    }
}

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

void push()
{

    cout<<"Enter value : ";
    cin>>value;
    cout<<endl;
    if(isFull())
    {
        cout<<"Sorry OverFlow "<<endl;
    }
    else if (rear == -1 and fr == -1)
    {
        rear=0;
        fr=0;
        arr[rear]=value;
```

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

void pop()
{
    cout<<endl;
    if(isEmpty())
    {
        cout<<"Sorry UnderFlow "<<endl;
    }
    else if (fr == rear)
    {
        rElement=arr[fr];
        arr[fr]=0;
        cout<<"The removed Element : "<<rElement<<" from
Position : "<<fr<<endl;
        fr=-1;
        rear=-1;
    }
    else
    {
        rElement=arr[fr];
        arr[fr]=0;
        cout<<"The removed Element : "<<rElement<<" from
Position : "<<fr<<endl;
```

Queue

```
        fr=fr+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]<<endl;
    }
}

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

Output :

Practical 04

Abhishek Gupta 2019450017

```
Enter Size of Queue (Less than 100) : 5

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

The Elements in Queue 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
02.Pop an Element
03.Display
Enter Your Choice :  1
Enter value : 20

The Elements in Queue 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
02.Pop an Element
03.Display
Enter Your Choice :  1
Enter value : 30

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

Practical 04

Abhishek Gupta 2019450017

```
0.Exit
01.Push an Element
02.Pop an Element
03.Display
Enter Your Choice :  1
Enter value : 40

The Elements in Queue are :
Position : 0 value : 10
Position : 1 value : 20
Position : 2 value : 30
Position : 3 value : 40
Position : 4 value : 0

0.Exit
01.Push an Element
02.Pop an Element
03.Display
Enter Your Choice :  1
Enter value : 50

The Elements in Queue are :
Position : 0 value : 10
Position : 1 value : 20
Position : 2 value : 30
Position : 3 value : 40
Position : 4 value : 50

0.Exit
01.Push an Element
02.Pop an Element
03.Display
Enter Your Choice :  1
Enter value : 60

Sorry OverFlow
The Elements in Queue are :
Position : 0 value : 10
Position : 1 value : 20
Position : 2 value : 30
Position : 3 value : 40
Position : 4 value : 50
```

Queue

Practical 04

Abhishek Gupta 2019450017

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

The removed Element : 10 from Position : 0
The Elements in Queue are :
Position : 0 value : 0
Position : 1 value : 20
Position : 2 value : 30
Position : 3 value : 40
Position : 4 value : 50

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

The removed Element : 20 from Position : 1
The Elements in Queue are :
Position : 0 value : 0
Position : 1 value : 0
Position : 2 value : 30
Position : 3 value : 40
Position : 4 value : 50

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

The removed Element : 30 from Position : 2
The Elements in Queue are :
Position : 0 value : 0
Position : 1 value : 0
Position : 2 value : 0
Position : 3 value : 40
Position : 4 value : 50
```

Queue

Practical 04

Abhishek Gupta 2019450017

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

The removed Element : 40 from Position : 3
The Elements in Queue are :
Position : 0 value : 0
Position : 1 value : 0
Position : 2 value : 0
Position : 3 value : 0
Position : 4 value : 50

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

The removed Element : 50 from Position : 4
The Elements in Queue are :
Position : 0 value : 0
Position : 1 value : 0
Position : 2 value : 0
Position : 3 value : 0
Position : 4 value : 0

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

Sorry UnderFlow
The Elements in Queue are :
Position : 0 value : 0
Position : 1 value : 0
Position : 2 value : 0
Position : 3 value : 0
Position : 4 value : 0
```

Queue

Practical 04

Abhishek Gupta 2019450017

```
0.Exit
01.Push an Element
02.Pop an Element
03.Display
Enter Your Choice :  3
The Elements in Queue are :
Position : 0 value : 0
Position : 1 value : 0
Position : 2 value : 0
Position : 3 value : 0
Position : 4 value : 0
0.Exit
01.Push an Element
02.Pop an Element
03.Display
Enter Your Choice :  0

Process returned 0 (0x0)   execution time : 218.308 s
Press any key to continue.
```

Queue

Abhishek Gupta 2019450017

4.2 Doubly_Queue_17.cpp

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

class DoublyQueue
{
public:
int choice,value,lfront,lrear,rrear,rfront,size,rElement,count;
int arr[100];

DoublyQueue()
{
    lfront=lrear=-1;
    for(int i=0;i<100;i++)
    {
        arr[i]=0;
    }

    cout<<"Enter Size of Queue (Less than 100) : ";
    cin>>size;
    cout<<endl;

    rrear=rfront=size;
}

void get()
{
    do
            {
```

Practical 04

Abhishek Gupta 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;
```

Queue

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

}

bool isEmptyr()
{
    if(rrear == size && rfront == size)
    {
        return true;
    }
    else
    {
        return false;
    }
}

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

bool isEmptyl()
{
    if(lrear == -1 && lfront == -1)
```

Queue

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

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

void lpush()
{

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

Queue

```
    else if (lrear == -1 and lfront == -1)
    {
        cout<<lrear<<" ";
        lrear=0;
        lfront=0;
        arr[lrear]=value;
    }
    else
    {
        cout<<lrear<<" ";
    lrear=lrear+1;
    arr[lrear]=value;
    }
    display();
    cout<<endl;
}

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

Queue

```
    else
    {
        cout<<"Sorry UnderFlow "<<endl;
    }
    display();
    cout<<endl;
}

void rpush()
{

    cout<<"Enter value : ";
    cin>>value;
    cout<<endl;
    if(isFullr())
    {
        cout<<"Sorry OverFlow "<<endl;
    }
    else if (rrear == size and rfront == size)
    {
        rrear=size-1;
        rfront=size-1;
        arr[rrear]=value;
    }
    else
    {
    rrear=rrear-1;
    arr[rrear]=value;
    }
    display();
    cout<<endl;
}
```

Queue

Practical 04

Abhishek Gupta 2019450017

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

Queue

```
};

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

Queue

Practical 04

Abhishek Gupta 2019450017

Output :

```
C:\Users\gupta\Desktop\Queue>dq.exe
Enter Size of Queue (Less than 100) : 3

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

-1 The Elements in Queue are :
Position : 0 value : 10
Position : 1 value : 0
Position : 2 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

0 The Elements in Queue are :
Position : 0 value : 10
Position : 1 value : 20
Position : 2 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 : 200
```

Practical 04

Abhishek Gupta 2019450017

```
The Elements in Queue are :
Position : 0 value : 10
Position : 1 value : 20
Position : 2 value : 200

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 Queue are :
Position : 0 value : 0
Position : 1 value : 20
Position : 2 value : 200

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 Queue are :
Position : 0 value : 0
Position : 1 value : 0
Position : 2 value : 200

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 : 1010
```

Queue

Practical 04

Abhishek Gupta 2019450017

```
The Elements in Queue are :
Position : 0 value : 0
Position : 1 value : 1010
Position : 2 value : 200

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 : 200 from Position : 2
The Elements in Queue are :
Position : 0 value : 0
Position : 1 value : 1010
Position : 2 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 :  5
The Elements in Queue are :
Position : 0 value : 0
Position : 1 value : 1010
Position : 2 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 :  0

C:\Users\gupta\Desktop\Queue>
```

Queue

4.3 Circular_Queue_17.cpp

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

class CircularQueue
{
public:
int choice,value,fr,rear,size,rElement;
int arr[100];

CircularQueue()
{
    rear=fr=-1;
    for(int i=0;i<100;i++)
    {
        arr[i]=0;
    }

    cout<<"Enter Size of CircularQueue (Less than 100) : ";
    cin>>size;
    cout<<endl;
}

void get()
{
    do
            {
                cout<<"0.Exit\n01.Push an Element\n02.Pop an
Element\n03.Display\n";
                cout<<"Enter Your Choice : "<<" ";
```

```
                cin>>choice;
                switch(choice)
                {
                    case 0:
                    break;

                    case 1:
                        push();
                        break;

                    case 2:
                        pop();
                        break;

                    case 3:
                        display();
                        break;

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

}

bool isEmpty()
{
    if(rear == -1 && fr == -1)
    {
        return true;
    }
    else
```

Queue

```
    {
        return false;
    }
}

bool isFull()
{
    if((rear+1)%size == fr)
    {
        return true;
    }
    else
    {
        return false;
    }
}

void push()
{

    cout<<"Enter value : ";
    cin>>value;
    cout<<endl;
    if(isFull())
    {
        cout<<"Sorry OverFlow "<<endl;
    }
    else if (rear == -1 and fr == -1)
    {
        rear=0;
        fr=0;
        arr[rear]=value;
```

```
    }
    else
    {
    rear=(rear+1)%size;
    arr[rear]=value;
    }
    display();
    cout<<endl;
}

void pop()
{
    cout<<endl;
    if(isEmpty())
    {
        cout<<"Sorry UnderFlow "<<endl;
    }
    else if (fr == rear)
    {
        rElement=arr[fr];
        arr[fr]=0;
        cout<<"The removed Element : "<<rElement<<" from
Position : "<<fr<<endl;
        fr=-1;
        rear=-1;
    }
    else
    {
        rElement=arr[fr];
        arr[fr]=0;
        cout<<"The removed Element : "<<rElement<<" from
Position : "<<fr<<endl;
```

Queue

Practical 04

Abhishek Gupta 2019450017

```
        fr=(fr+1)%size;
    }
    display();
    cout<<endl;
}

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

};

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

Queue

Practical 04

Abhishek Gupta 2019450017

Output :

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

The Elements in CircularQueue 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
02.Pop an Element
03.Display
Enter Your Choice :  1
Enter value : 20

The Elements in CircularQueue 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
02.Pop an Element
03.Display
Enter Your Choice :  1
Enter value : 30

The Elements in CircularQueue are :
Position : 0 value : 10
Position : 1 value : 20
Position : 2 value : 30
Position : 3 value : 0
Position : 4 value : 0
```

Practical 04

Abhishek Gupta 2019450017

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

The removed Element : 10 from Position : 0
The Elements in CircularQueue are :
Position : 0 value : 0
Position : 1 value : 20
Position : 2 value : 30
Position : 3 value : 0
Position : 4 value : 0

0.Exit
01.Push an Element
02.Pop an Element
03.Display
Enter Your Choice :  1
Enter value : 40

The Elements in CircularQueue are :
Position : 0 value : 0
Position : 1 value : 20
Position : 2 value : 30
Position : 3 value : 40
Position : 4 value : 0

0.Exit
01.Push an Element
02.Pop an Element
03.Display
Enter Your Choice :  1
Enter value : 50

The Elements in CircularQueue are :
Position : 0 value : 0
Position : 1 value : 20
Position : 2 value : 30
Position : 3 value : 40
Position : 4 value : 50
```

Queue

Practical 04

Abhishek Gupta 2019450017

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

The removed Element : 20 from Position : 1
The Elements in CircularQueue are :
Position : 0 value : 0
Position : 1 value : 0
Position : 2 value : 30
Position : 3 value : 40
Position : 4 value : 50

0.Exit
01.Push an Element
02.Pop an Element
03.Display
Enter Your Choice :  1
Enter value : 60

The Elements in CircularQueue are :
Position : 0 value : 60
Position : 1 value : 0
Position : 2 value : 30
Position : 3 value : 40
Position : 4 value : 50

0.Exit
01.Push an Element
02.Pop an Element
03.Display
Enter Your Choice :  1
Enter value : 70

The Elements in CircularQueue are :
Position : 0 value : 60
Position : 1 value : 70
Position : 2 value : 30
Position : 3 value : 40
Position : 4 value : 50
```

Queue

Practical 04

Abhishek Gupta 2019450017

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

The removed Element : 30 from Position : 2
The Elements in CircularQueue are :
Position : 0 value : 60
Position : 1 value : 70
Position : 2 value : 0
Position : 3 value : 40
Position : 4 value : 50

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

The removed Element : 40 from Position : 3
The Elements in CircularQueue are :
Position : 0 value : 60
Position : 1 value : 70
Position : 2 value : 0
Position : 3 value : 0
Position : 4 value : 50

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

The removed Element : 50 from Position : 4
The Elements in CircularQueue are :
Position : 0 value : 60
Position : 1 value : 70
Position : 2 value : 0
Position : 3 value : 0
Position : 4 value : 0

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

The removed Element : 60 from Position : 0
```

Queue

Abhishek Gupta 2019450017

4.4 Queue Using LinkedList

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

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

class QueueLink
{
public:
int choice,value;

void get()
{
    do
            {
                cout<<"0.Exit\n1.Push\n2.Pop\n3.display\n";
                cout<<"Enter Your Choice : "<<" ";
                cin>>choice;
                switch(choice)
                {
                    case 0:
                    break;

                    case 1:
                        push();
```

```
                        break;

                    case 2:
                        pop();
                        break;

                    case 11:
                        display();
                        break;

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

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

Queue

```
                p->next=NULL;
                rear=rear->next;

            }
            display();
}

void pop()
{
        cout<<"Delete Fisrt element "<<endl;
        if(front == NULL && rear == NULL)
        {
            cout<<"Empty List"<<endl<<endl;
        }
        else if (rear == front)
        {
            p=front;
            list=list->next;
            front=front->next;
            delete p;
        }
        else
        {
            p=front;
            list=list->next;
            front=front->next;
            delete p;
        }
        display();
}

void display()
```

Practical 04

Abhishek Gupta 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()
{
    QueueLink s;
    s.get();
    return 0;
}
```

Queue

Practical 04

Abhishek Gupta 2019450017

Output :

```
C:\Users\gupta\Desktop\Queue>g++ queuelink.cpp -o ql.exe

C:\Users\gupta\Desktop\Queue>ql.exe
0.Exit
1.Push
2.Pop
3.display
Enter Your Choice :  1
Enter the value : 10
The List is : 10|----->

0.Exit
1.Push
2.Pop
3.display
Enter Your Choice :  1
Enter the value : 20
The List is : 10|----->20|----->

0.Exit
1.Push
2.Pop
3.display
Enter Your Choice :  1
Enter the value : 30
The List is : 10|----->20|----->30|----->

0.Exit
1.Push
2.Pop
3.display
Enter Your Choice :  1
Enter the value : 40
The List is : 10|----->20|----->30|----->40|----->

0.Exit
1.Push
2.Pop
3.display
Enter Your Choice :  2
Delete Fisrt element
The List is : 20|----->30|----->40|----->
```

Practical 04

Abhishek Gupta 2019450017

```
0.Exit
1.Push
2.Pop
3.display
Enter Your Choice :  2
Delete Fisrt element
The List is : 30|----->40|----->

0.Exit
1.Push
2.Pop
3.display
Enter Your Choice :  2
Delete Fisrt element
The List is : 40|----->

0.Exit
1.Push
2.Pop
3.display
Enter Your Choice :  2
Delete Fisrt element

List is Empty

0.Exit
1.Push
2.Pop
3.display
Enter Your Choice :  0

C:\Users\gupta\Desktop\Queue>
```