**Practicle no 4: Implementation of program based on dequeue.**

**Name:-Bindiya Chaudhari**

**Roll no:-30**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include"iostream.h"

#include"conio.h"

class DQUEUE\_30

{

int \*A,size,rear,front;

public:

DQUEUE\_30(int);

void ADD\_REAR(int);

int DEL\_FRONT();

int DEL\_REAR();

void LIST\_QUE();

};

DQUEUE\_30::DQUEUE\_30(int par)

{

size=par;

A=new int[size+1];

rear=front=0;

}

void DQUEUE\_30::ADD\_REAR(int ele)

{

if (rear==size)

cout<<endl<<"Queue is full";

else

{

if(front==0)

front=1;

rear=rear+1;

A[rear]=ele;

}

}

int DQUEUE\_30::DEL\_FRONT()

{

if(front==0)

{

cout<<endl<<"Queue is empty";

return NULL;

}

else

{

int ele=A[front];

if(front==rear)

rear=front=0;

else

front=front+1;

return ele;

}

}

int DQUEUE\_30::DEL\_REAR()

{

if(front==0)

{

cout<<endl<<"Que is empty";

return NULL;

}

else

{

int ele=A[rear];

if(rear==front)

front=rear=0;

else

rear=rear-1;

return ele;

}

}

void DQUEUE\_30::LIST\_QUE()

{

if(front==0)

cout<<endl<<"Queue is empty";

else

{

cout<<endl<<"Queue elements are:";

for(int i=front;i<=rear;i++)

cout<<A[i]<<" ";

}

}

void MENU()

{

int n,opt,ele;

cout<<endl<<"Enter the size of queue:";

cin>>n;

DQUEUE\_30 obj(n);

do

{

cout<<endl<<"1. ADD @ REAR";

cout<<endl<<"2. DEL from FRONT";

cout<<endl<<"3. DEL from REAR";

cout<<endl<<"4. EXIT";

cout<<endl<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

cout<<endl<<"Enter your choice:";

cin>>opt;

switch(opt)

{

case 1:

cout<<endl<<"Enter the elements:";

cin>>ele;

obj.ADD\_REAR(ele);

obj.LIST\_QUE();

break;

case 2:

ele=obj.DEL\_FRONT();

if(ele)

cout<<endl<<"deleted ele="<<ele;

obj.LIST\_QUE();

break;

case 3:

ele=obj.DEL\_REAR();

if(ele)

cout<<endl<<"Deleted ele="<<ele;

obj.LIST\_QUE();

break;

case 4:

return;

default:

cout<<endl<"Invalid choice";

}

}while(1);

}

void main()

{

clrscr();

MENU();

getch();

}