**Name – Surve Sachin Jitendra**

**Roll No. – 127**

**Assignment no. – 1.4**

**Title – Implementation of programs based on De-Que using array.**

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

#include<iostream.h>

#include<conio.h>

class DEQ\_127

{

private:

int \*A,front,rear,size;

public:

DEQ\_127(int par);

void ADD\_END();

void ADD\_BEG();

int DEL\_END();

int DEL\_BEG();

void LIST\_ALL();

};

DEQ\_127::DEQ\_127(int par)

{

front=0,rear=0;

size=par;

A=new int[size+1];

}

void DEQ\_127::ADD\_END()

{

if(front==1 && rear==size)

{

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

return;

}

int ele;

if(front==0)

{

front=1;

}

rear=rear+1;

cout<<"Enter the element you want to add: ";

cin>>ele;

A[rear]=ele;

}

void DEQ\_127::ADD\_BEG()

{

int ele;

if(front==1 && rear==size)

{

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

return;

}

if(front==0)

{

front=1;

}

if(front>1)

{

front=front-1;

}

else

{

int i;

i=rear;

while(i>0)

{

A[i+1]=A[i];

i--;

}

}

cout<<"Enter the element you want to add: ";

cin>>ele;

A[front]=ele;

rear=rear+1;

}

int DEQ\_127::DEL\_END()

{

if(front==0)

{

return NULL;

}

int ele=A[rear];

if(front==rear)

{

front=0;

}

rear=rear-1;

return ele;

}

int DEQ\_127::DEL\_BEG()

{

if(front==0)

{

return NULL;

}

int ele=A[front];

int i=front;

while(i<rear)

{

A[i]=A[i+1];

i++;

}

if(front==rear)

{

front=0;

}

rear=rear-1;

return ele;

}

void DEQ\_127::LIST\_ALL()

{

if(front==0)

{

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

return;

}

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

{

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

}

}

void MENU()

{

int opt,size,ele;

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

cin>>size;

DEQ\_127 obj(size);

do

{

cout<<"\n1. Add element at end.";

cout<<"\n2. Add element at begining.";

cout<<"\n3. Remove element from end.";

cout<<"\n4. Remove element from begining.";

cout<<"\n5. List all";

cout<<"\n6. Exit";

cout<<"\nEnter your choice: ";

cin>>opt;

switch(opt)

{

case 1: obj.ADD\_END();

break;

case 2: obj.ADD\_BEG();

break;

case 3: ele=obj.DEL\_END();

if(ele==NULL)

{

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

}

else

{

cout<<"The deleted element is "<<ele<<endl;

}

break;

case 4: ele=obj.DEL\_BEG();

if(ele==NULL)

{

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

}

else

{

cout<<"The deleted element is "<<ele<<endl;

}

break;

case 5: obj.LIST\_ALL();

break;

case 6:

return;

default:

cout<<"Invalid option."<<endl;

}

}while(1);

}

void main()

{

clrscr();

MENU();

getch();

}