6.4.Program that Implements Double ended Queue using Linked list.

#include <conio.h> #include <iostream> using namespace std;

class queue

{ private:

public:

};

int data; queue \*next; queue \*prv;

void add(int); void display(); int remove(); void addf(int);

int remover(void);

queue \*front; queue \*rear;

void queue::add(int num)

{

if((front==NULL)&&(rear==NULL))

{

front=rear=new queue; front->data=num; front->next=NULL; front->prv=NULL;

}

else

{

rear->next=new queue; rear->next->data=num; rear->next->next=NULL; rear->next->prv=rear; rear=rear->next;

}

}

void queue::display()

{

queue \*q=front; if((front==NULL)&&(rear==NULL))

{

cout<<"\nStack empty";

}

else

{

while(q!=NULL)

{

cout<<q->data<<" "; q=q->next;

}

}

}

int queue::remove()

{

int num; queue \*temp;

if((front==NULL)&&(rear==NULL))

{

return -1;

}

else

{

temp=front; num=front->data; front=front->next; front->prv=NULL; delete(temp);

}

return num;

}

void queue::addf(int num)

{

queue \*temp; if((front==NULL)&&(rear==NULL))

{

front=rear=new queue; front->data=num; front->next=NULL; front->prv=NULL;

}

else

{

temp=front; front=new queue; front->data=num; front->prv=NULL; front->next=temp;

front->next->prv=front;

}

}

int queue::remover(void)

{

queue \*temp; int num;

if((front==NULL)&&(rear==NULL))

{

return -1;

}

else

{

temp=rear; num=rear->data; rear=rear->prv; rear->next=NULL; delete(temp); return num;

}}

int main()

{

queue z; char c='y'; int num,op; while(c=='y')

{ cout<<"\n1.Add"; cout<<"\n2.Display";

cout<<"\n3.Remove"; cout<<"\n4.Add from Front"; cout<<"\n5. Remove from Rear"; cout<<"\n Enter an option: "; cin>>op;

switch(op)

{ case 1:

cout<<"\nEnter value: "; cin>>num;

z.add(num);

cout<<"\n\tDo you want to continue: "; break;

case 2:

case 3:

case 4:

case 5:

}

cout<<"Queue values:\t"; z.display();

cout<<"\n\tDo you want to continue: "; break;

num=z.remove();

cout<< num <<" removed";

cout<< "Do you want to continue "; break;

cout<<"Enter number: "; cin>>num;

z.addf(num);

cout<<"Do you want to continue"; break;

num=z.remover();

cout<< num <<" removed";

cout<< "Do you want to continue "; break;

c=getch();

}

}