Lab-6

Name:Aniket Verma

Reg no.209301578

Q1

Code:

#include<iostream>

using namespace std;

struct node

{

int data;

node\* next;

};

node\* front=NULL;

node\* rear=NULL;

void Enqueue(int x)

{

node\* temp=new node;

temp->data=x;

temp->next=0;

if(front==0&&rear==0)

{

front=rear=temp;

}

else

{

rear->next=temp;

rear=temp;

}

}

int Dequeue()

{ if(front==0)

{

cout<<"The Queue is empty\n";

return -1;

}

else if(front==rear)

{

int x=front->data;

node\* temp=front;

front=rear=0;

delete temp;

return x;

}

else

{node\* temp=front;

front=front->next;

int x=temp->data;

delete temp;

return x;

}

}

int main()

{int x,n;

do

{cout<<"What operation you want to perform\n1-Enqueue\n2-Dequeue\n0-Exit\n";

cin>>n;

switch(n)

{

case 1:cout<<"Enter the element you want to add\n";

cin>>x;

Enqueue(x);

break;

case 2: x=Dequeue();

if(x!=-1)

cout<<"The element dequeued is:"<<x<<endl;

}

}

while(n);

}

Output:



Q-2

Code:

#include<iostream>

using namespace std;

struct node

{

int data;

node\* next;

};

node\* front=NULL;

node\* rear=NULL;

void Enqueue(int x)

{

node\* temp=new node;

temp->data=x;

temp->next=0;

if(front==0&&rear==0)

{

front=rear=temp;

rear->next=front;

}

else

{ temp->next=front;

rear->next=temp;

rear=temp;

}

}

int Dequeue()

{ if(front==0)

{

cout<<"The Queue is empty\n";

return -1;

}

else if(front==rear)

{

int x=front->data;

node\* temp=front;

front=rear=0;

delete temp;

return x;

}

else

{node\* temp=front;

front=front->next;

rear->next=front;

int x=temp->data;

delete temp;

return x;

}

}

int main()

{int x,n;

do

{cout<<"What operation you want to perform\n1-Enqueue\n2-Dequeue\n0-Exit\n";

cin>>n;

switch(n)

{

case 1:cout<<"Enter the element you want to add\n";

cin>>x;

Enqueue(x);

break;

case 2: x=Dequeue();

if(x!=-1)

cout<<"The element dequeued is:"<<x<<endl;

}

}

while(n);

}

Output:

