**INPUT**

**Priority Queue**

#include <iostream>

using namespace std;

class Node

{

public:

int Priority;

int roll\_number;

Node \*next;

Node \*previous;

Node(int p, int r)

{

Priority = p;

roll\_number = r;

next = NULL;

previous = NULL; } };

class d\_link\_list

{

private:

Node \*first;

Node \*last;

public:

d\_link\_list()

{

first = last = NULL; }

void insert(int p, int r)

{

Node \*newNode = new Node(p, r);

Node \*curr = first;

Node \*prev = curr;

if(curr == NULL)

{

first = last = newNode;

delete curr;

delete prev; }

else

{

while(curr != NULL && curr->Priority <= p)

{

prev = curr;

curr = curr->next; }

prev->next = newNode;

newNode->previous = prev;

newNode->next = curr; } }

void pop()

{

Node \*tmp = first;

while(tmp->next->next != NULL)

{

tmp = tmp->next; }

tmp->next = NULL;

last = tmp; }

void display()

{

Node \*tmp = first;

while(tmp != NULL)

{

cout << "Priority :" << tmp->Priority << endl;

cout << "Roll Number :" << tmp->roll\_number << endl;

tmp = tmp->next; } } };

int main()

{

d\_link\_list d;

int choice;

int priority,roll\_no;

cout << "\t\tPriority Queue Implementation using Linked List" << endl << endl;

do{

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*Menu Priority Queue\*\*\*\*\*\*\*";

cout<<"\n1)To add a student";

cout<<"\n2)To remove the student";

cout<<"\n3)Display all student with Priority";

cout<<"\n4)Exit";

cout<<"\n######################################";

cout<<"\n Enter your choice";

cin>>choice;

switch(choice)

{

case 1:

cout<<"\n Enter Roll no with Priority";

cin>>roll\_no>>priority;

d.insert(priority, roll\_no);

cout<<"\n inserted";

break;

case 2:

d.pop();

break;

case 3:

d.display();

break;

case 4:

break;

default:

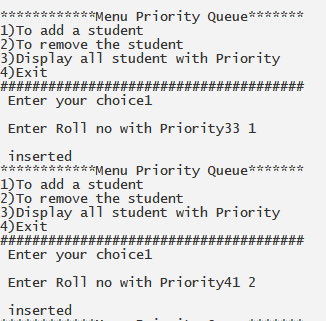
cout<<"\n Please enter proper value"; } }

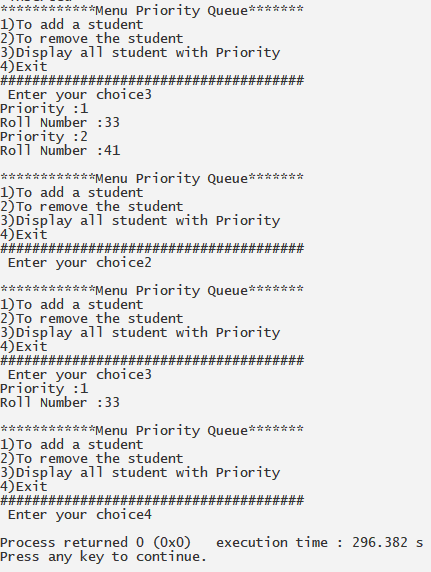
while(choice!=4);

return 0;

}

**OUTPUT**





**INPUT**

**DOUBLE ENDED QUQUE**

#include<iostream>

using namespace std;

class node

{

public:

int data;

node\* next;

node\* prev; };

class Dequey{

public:

node\* newnode;

node\* temp;

node\* head;

node\* tail;

int i,n;

void create\_node(){

newnode = new node();

cout<<"\n Enter data to be inseted:";

cin>>newnode->data;

newnode->next = NULL;

newnode->prev = NULL; }

void init(){

head = NULL;

tail = NULL;

cout<<"Enter Data of First NODE"<<endl;

create\_node();

head = newnode;

tail = newnode; }

void insertion\_end(){

create\_node();

newnode->prev = tail;

tail->next = newnode;

tail = tail->next; }

void deletion\_beg(){

head = head->next; }

void deletion\_end(){

tail = tail->prev;

tail->next = NULL; }

void display()

{

temp = head;

if(temp!=tail){

while(temp != tail){

cout<<temp->data<<" ";

temp = temp->next; } }

else{

cout<<temp->data<<" ";

temp = temp->next; } }

void insertion\_beg(); };

void Dequey::insertion\_beg(){

create\_node();

newnode->next = head;

head->prev= newnode;

head = head->prev; }

int main(){

Dequey d;

int choice=0,flag=0;

d.init();

do{

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*DQUEUE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<endl<<"1. Insertion at Beg \n2. Insertion at End \n3. Deletion at Beg \n4. Deletion at End \n5. Display all \n 6.Exit"<<endl;

cout<<"\n################################\n";

cout<<"\n Enter your choice:";

cin>>choice;

switch(choice){

case 1:

d.insertion\_beg();

break;

case 2:

d.insertion\_end();

break;

case 3:

d.deletion\_beg();

break;

case 4:

d.deletion\_end();

break;

case 5:

d.display();

break;

case 6:

break;

default:

cout<<"\n Please enter a valid choice";

}

}while(choice !=6);

d.display();

return 0;

}

**OUTPUT**

