RIPHAH
INTERNATIONAL UNIVERSITY

Name: **Muhammad Fahad**

SAP ID: **29006**

Course: **BSCS**

Section: **BSCS-3**

Subject: **DSA**

Submitted To: **Ma'am Zarmina**

Submission Date: **06/07/2022**

Priority queues using Stacks

```
#include<iostream>
using namespace std;

class Stacks
{
int *arr;
int capacity;
int pos;
public:
Stacks()
{
capacity = 10;
arr = new int[capacity];
pos = -1;
for (int i = 0; i < capacity; i++)
{
arr[i] = 0;
}
}
Stacks(int c)
{
capacity = c;
arr = new int[capacity];
pos = -1;
```

```
for (int i = 0; i < capacity; i++)
{
arr[i] = 0;
}
}
bool isEmpty()
{
if (pos<0)
{
return true;
}
return false;
}
void push(int data)
{
if (pos >= capacity - 1)
cout << "Stack Full" << endl;
else
{
pos++;
arr[pos] = data;
}
}
int pop()
{
int x = 0, y = 0;

if (pos <= -1)
{
```

```
return -1;
}
else
{
x = arr[pos];
arr[pos] = 0;
pos--;
}
return x;
}
void display()
{
for (int i = 0; i < capacity; i++)
{
cout << endl << arr[i];
}
}
~Stacks();

};

class PQueue
{
private:
Stacks* s;
Stacks* pro;
int capacity;
int priority;
int rare, front;
```

```
public:
PQueue()
{
capacity = 10;
rare = 0;
front = 0;
}
PQueue(int c)
{
capacity = c;
s = new Stacks(c);
pro = new Stacks(c);
rare = 0;
front = 0;
}
bool isFull()
{
if (rare == capacity)
return true;
return false;
}
bool isEmpty()
{
if (front == capacity)
return true;
return false;
}
void Enqueue(int a,int p)
{
```

```
if (!isFull())
{
if (rare == 0)
{
s->push(a);
pro->push(p);
}
else
{
Stacks* ss = new Stacks(capacity);
Stacks* proo = new Stacks(capacity);
int x = pro->pop();
proo->push(x);
ss->push(s->pop());
if (x>=p)
{
pro->push(proo->pop());
s->push(ss->pop());
s->push(a);
pro->push(p);
}
else
{
while (!(x >= p))
{
x = pro->pop();
if (x == -1)
{
break;
```

```
}
if (x >= p)
{
pro->push(x);
break;
}
proo->push(x);
ss->push(s->pop());
}
pro->push(p);
s->push(a);
int y = proo->pop();
pro->push(y);
s->push(ss->pop());
while (!(y<0))
{
y = proo->pop();
if (y == -1)
{
break;
}
pro->push(y);
s->push(ss->pop());
}
}
}
rare++;
}
else
```

```
{
cout << "The  queue is already full." << endl;
}
}
int Dequeue()
{
int x;

if (!isEmpty())
{
if (front == 0)
{
front++;
x = s->pop();
}
else
{
x = s->pop();
front++;
}
return x;
}
else
{
cout << "The  queue is empty." << endl;
}
}
void display()
{
```

```
int x;
for (int i = 0; i < capacity; i++)
{
x = Dequeue();
cout << "Data: " << x <<"  Priority: " << pro->pop() << endl;
rare--;
}

}
};
int main()
{

int x;
int y;
int q;

cout << "Please enter the capacity of the queue: ";
cin >> q;

PQueue* q1 = new PQueue(q);

for (int i = 0; i < q; i++)
{
cout << "Please enter " << i+1 << " value in queue: ";
cin >> x;
cout << "Please enter it's priority: ";
cin >> y;
q1->Enqueue(x,y);
```

```
}


q1->display();
}
```

Program Execution