Rajvaibhav Rahane
17u283 223045
SE-C Comp,Viit,Pune

Aim: To implement a Priority Queue .

CODE:

```
/*
*     @author    Rajvaibhav Rahane
*/
/*
*     Program to implement priority queue for a hospital.
*     places patients in lower priority -> first order.
*/
#include<iostream>
using namespace std;
class EmptyQueueException:public exception{
      public:
            const char*what()const throw(){
                  return "EmptyQueueException";
            }
};
struct Node{
      int priority;
      string name;
      Node *next;
};
void printNode(const Node * node){
      cout<<"Patient : "<<node->name<<" Priority : "<<node-
>priority<<endl;
}
class PriorityQueue{
      Node * front;
      public:PriorityQueue(){
            front=NULL;
      }
      void enQueue(Node * nn);
      Node * deQueue();
      void printQueue();
};
void PriorityQueue::enQueue(Node * nn){
            if(front==NULL || nn->priority<front->priority){
                  nn->next=front;
                  front=nn;
            }else{
                  Node *temp=front;
                  while(temp->next!=NULL && temp->next->priority<=nn-
>priority){
                        temp=temp->next;
                  }
                  nn->next=temp->next;
                  temp->next=nn;
            }
}
Node* PriorityQueue::deQueue(){
      if(!front)
```

```
            throw EmptyQueueException();
      Node* removedNode=front;
      front=front->next;
      removedNode->next=NULL;
      return removedNode;
}
void PriorityQueue::printQueue(){
      Node * temp=front;
      while(temp){
            printNode(temp);
            temp=temp->next;
      }
}
Node * createNode(){
      string patientName;int priority;
      cout<<"Enter Patient Name and Priority\t";
      cin>>patientName>>priority;
      Node * nn=new Node;
      nn->name=patientName;
      nn->priority=priority;
      nn->next=NULL;
      return nn;
}
void printMenu(){
      cout<<"1)Add Patient\t";
      cout<<"2)Treat Patient\t";
      cout<<"3)Display Queue\n";
      cout<<"4)Exit\tChoice : ";
}
int main(){
      int choice;
      PriorityQueue hospitalQueue;
      printMenu();
      do{
            cin>>choice;
            switch(choice){
                  case 1:{
                        hospitalQueue.enQueue(createNode());
                        break;
                  }
                  case 2:{
                        try{
                              Node *
removedNode=hospitalQueue.deQueue();
                              printNode(removedNode);
                              delete removedNode;
                        }catch(EmptyQueueException e){
                              cout<<"No Patients\n";
                        }
                        break;
                  }
                  case 3:{
                        hospitalQueue.printQueue();
                        break;
                  }
                  case 4:break;
```

```
            }
      }while(choice!=4);
      return 0;
}
```

Output: