**COMP 203 Lab 9**

**Linked List Implementation of Priority Queue**

**SOLUTION**

class Node { //2pt

char data;

int priority;

Node next;

public Node(char d, int p) {//3pt

data = d;

priority = p;

next = null;

}

}

class PriorityQueue {//5pt

private Node head;

private int size;

public PriorityQueue() {//5pt

head = null;

size = 0;

}

public void insert(char addedData, int priority) {//20pt

Node newNode = new Node(addedData, priority);

if (head == null || priority < head.priority) {

newNode.next = head;

head = newNode;

} else {

Node current = head;

while (current.next != null && priority >= current.next.priority) {

current = current.next;

}

newNode.next = current.next;

current.next = newNode;

}

size++;

}

public char removeMin() {//20pt

if (isEmpty()) {

throw new IllegalStateException("Priority Queue is empty. Cannot remove minimum element.");

}

Node temp = head;

char removedData = temp.data;

head = head.next;

temp.next = null; // Disconnect the removed node

size--;

return removedData;

}

public char min() {//10pt

if (isEmpty()) {

throw new IllegalStateException("Priority Queue is empty. Cannot get minimum element.");

}

return head.data;

}

public int size() {//5pt

return size;

}

public void printQueue() {//10pt

Node current = head;

while (current != null) {

System.out.print("(" + current.data + "," + current.priority + ")");

if (current.next != null) {

System.out.print("->");

}

current = current.next;

}

System.out.println("NULL");

}

public boolean isEmpty() {

return size == 0;

}

}

public class Main {

public static void main(String[] args) {

// Creating and testing the given Priority Queue //10pt

PriorityQueue pq = new PriorityQueue();

pq.insert('A', 1);

pq.insert('G', 2);

pq.insert('U', 3);

pq.insert('C', 4);

pq.insert('E', 5); //2pt

System.out.print("Priority Queue created: ");

pq.printQueue();//2pt

System.out.println("Removed element with highest priority: " + pq.removeMin());//2pt

System.out.println("Minimum element in the Priority Queue: " + pq.min());//2pt

System.out.println("Size of the Priority Queue: " + pq.size());//2pt

}

}