

// Question : Program to implement a descending priority queue using singly linked list.

// Author : Sharun E Rajeev

```
import java.io.*;
```

```
class PriorityQ {
    private int maxSize;
    private int[] qArray;
    private int n;

    public PriorityQ(int size) {
        maxSize = size;
        qArray = new int[maxSize];
        n = 0;
    }

    public void insert(int value) {
        int i;
        if (isFull()) {
            System.out.println("No more insertions possible.");
        }
        else if (n == 0) {
            qArray[n++] = value;
        } else {
            for (i = n - 1; i >= 0; i--) {
                if (value > qArray[i])
                    qArray[i + 1] = qArray[i];
                else
                    break;
            }
            qArray[i + 1] = value;
            n++;
        }
    }

    public int remove() {
        if (isEmpty()) {
            System.out.println("No more deletions possible.");
            return 0;
        } else {
            return qArray[--n];
        }
    }

    public int peekMin() {
        return qArray[n - 1];
    }

    public boolean isEmpty() {
        return n == 0;
    }

    public boolean isFull() {
```

```

        return n == maxSize;
    }

    public void display() {
        for (int i = 0; i < n; i++) {
            System.out.print(qArray[i] + " ");
        }
    }
}

class LabB4 {
    public static void main(String[] args) throws IOException {
        PriorityQ pQ = new PriorityQ(20);
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
        int choice;
        do {
            System.out.println("\nPriority Queue Manager");
            System.out.println("1. Insert");
            System.out.println("2. Remove");
            System.out.println("3. Display Minimum element");
            System.out.println("4. Display");
            System.out.println("5. Exit");
            choice = Integer.parseInt(br.readLine());
            switch (choice) {
                case 1:
                    System.out.println("Enter data to be inserted.");
                    pQ.insert(Integer.parseInt(br.readLine()));
                    break;
                case 2:
                    System.out.println("Removed element is " + pQ.remove());
                    break;
                case 3:
                    System.out.println(pQ.peekMin());
                    break;
                case 4:
                    pQ.display();
                    break;
                case 5:
                    System.out.println("See you later!");
                    break;
                default: System.out.println("wrong choice");
            }
        } while (choice != 5);
    }
}

```