Name: Vijay Misal
Div: C Batch: C3
Roll No: 233073
PRN No: 22320079

Practical No: 9

Title: a) The internship is offered to students based on rank obtained in second year of graduation. Create suitable non-linear data structure to identify next topper student for internship. (Create max-heap).

b) Sort the student data in ascending order of grades

Code:

```
// a) The internship is offered to students based on rank obtained in
second year of graduation. Create
// suitable non-linear data structure to identify next topper student for
internship. (Create max-heap).
// b) Sort the student data in ascending order of grades.

import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of students: ");
        int n = sc.nextInt();
        MaxHeap heap = new MaxHeap(n);
        for (int i = 0; i < n; i++) {
            System.out.println("Enter the name of student " + (i + 1) + ":
");
            String name = sc.next();
            System.out.println("Enter the rank of student " + (i + 1) + ":
");
            int rank = sc.nextInt();
            heap.insert(name, rank);
        }
        System.out.println("The next topper student for internship is: ");
        heap.peek();
        System.out.println("The students sorted in ascending order of
grades are: ");
        heap.sort();
        sc.close();
    }
```

```
}

class MaxHeap {
    private Student[] heap;
    private int size;
    private int maxSize;

    public MaxHeap(int maxSize) {
        this.maxSize = maxSize;
        this.size = 0;
        heap = new Student[this.maxSize + 1];
        heap[0] = new Student("", Integer.MAX_VALUE);
    }

    private int parent(int pos) {
        return pos / 2;
    }

    private int leftChild(int pos) {
        return 2 * pos;
    }

    private int rightChild(int pos) {
        return 2 * pos + 1;
    }

    private boolean isLeaf(int pos) {
        return pos >= (size / 2) && pos <= size;
    }

    private void swap(int fpos, int spos) {
        Student tmp;
        tmp = heap[fpos];
        heap[fpos] = heap[spos];
        heap[spos] = tmp;
    }

    private void maxHeapify(int pos) {
        if (!isLeaf(pos)) {
            if (heap[pos].rank < heap[leftChild(pos)].rank ||
heap[pos].rank < heap[rightChild(pos)].rank) {
                if (heap[leftChild(pos)].rank >
heap[rightChild(pos)].rank) {
                    swap(pos, leftChild(pos));
                    maxHeapify(leftChild(pos));
                } else {
                    swap(pos, rightChild(pos));
                    maxHeapify(rightChild(pos));
```

```
                }
            }
        }
    }

    public void insert(String name, int rank) {
        heap[++size] = new Student(name, rank);
        int current = size;
        while (heap[current].rank > heap[parent(current)].rank) {
            swap(current, parent(current));
            current = parent(current);
        }
    }

    public void peek() {
        System.out.println(heap[1].name);
    }

    public void sort() {
        for (int i = size / 2; i >= 1; i--) {
            maxHeapify(i);
        }
        for (int i = size; i > 1; i--) {
            swap(1, i);
            size--;
            maxHeapify(1);
        }
        for (int i = 1; i <= size; i++) {
            System.out.println(heap[i].name + " " + heap[i].rank);
        }
    }
}

class Student {
    String name;
    int rank;

    public Student(String name, int rank) {
        this.name = name;
        this.rank = rank;
    }
}
```

Output:

Enter the number of students:

5

Enter the name of student 1:

vijay

Enter the rank of student 1:

2

Enter the name of student 2:

shubham

Enter the rank of student 2:

3

Enter the name of student 3:

kanhaiya

Enter the rank of student 3:

5

Enter the name of student 4:

siddhant

Enter the rank of student 4:

4

Enter the name of student 5:

vedant

Enter the rank of student 5:

1

The next topper student for internship is:

kanhaiya

The students sorted in ascending order of grades are:

vedant 1