Experiment1.4

Student Name:Rajiv Paul **UID:20BCS1812**
Branch: CSE **Section/Group: 20BCS_WM_702(A)**
Semester: 5th **Date of Performance:15/09/2022**
Subject Name: Competitive Coding **Subject Code: 20CSP-314**

1. **Aim/Overview of the practical:**

To solve the following hacker rank problems based on Searching and Sorting.

2. Task to be done/ Which logistics used:

https://www.hackerrank.com/challenges/quicksort1/problem?isFullScreen=true

3. Steps for experiment/practical/Code:

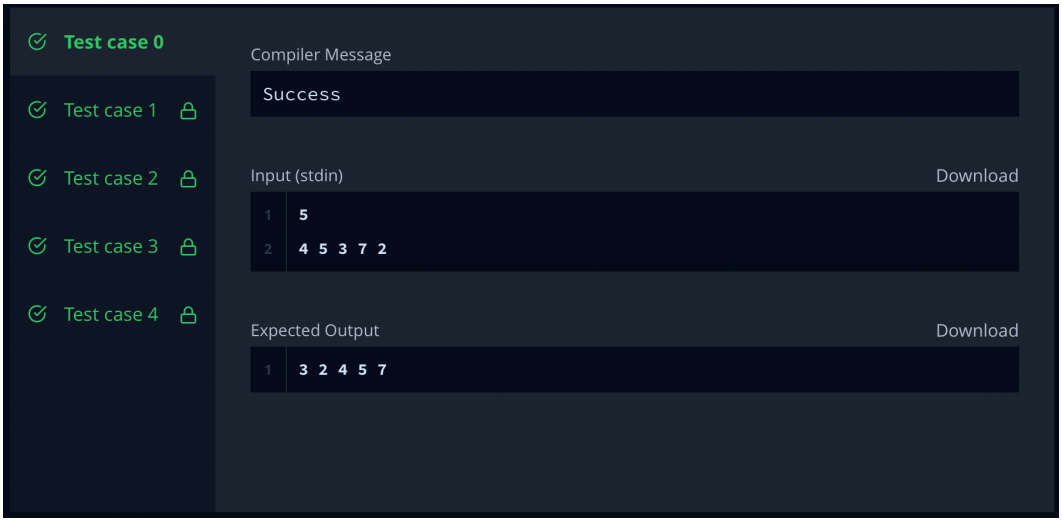
```
import java.util.*;

public class Solution {
   public static void main(String[] args) {
      Scanner I = new Scanner(System.in);
        int n = I.nextInt();
        int[] arr = new int[n];
        for(int i=0;i<n;i++){
          arr[i]=I.nextInt();
        }
        partition(arr);
        printArr(arr);
```

```
static void printArr(int[] arr) {
    for(int n: arr){
       System.out.print(n+" ");
    }
      System.out.println("");
}
static void partition(int[] arr) {
   int r=arr[0];
   int[] cp=Arrays.copyOf(arr, arr.length);
   int c=0;
   for(int i=1;i<arr.length;i++){
      if(cp[i]<=r){
         arr[c]=cp[i];
         c++;
      }
   }
   arr[c]=r;
   c++;
```

```
        for(int j=0;j<arr.length;j++){
            if(cp[j]>r){
                arr[c]=cp[j];
                c++;
            }
        }
    }
}
```

4. Result/Output/Writing Summary:



1. **Aim/Overview of the practical:**

To solve the following hacker rank problems based on Searching and Sorting..

2. Task to be done/ Which logistics used:

https://www.hackerrank.com/challenges/countingsort4/problem?isFullScreen=true

3. Steps for experiment/practical/Code:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {
    public static void main(String[] args) throws Exception {
        BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
        int n = Integer.parseInt(in.readLine());
        StringBuffer[] map = new StringBuffer[100];
        for(int i = 0; i < 100; i++) {
            map[i] = new StringBuffer();
        }
        for(int i = 0; i < n; i++) {
            StringTokenizer tok = new StringTokenizer(in.readLine());
            int v = Integer.parseInt(tok.nextToken());
            String s = tok.nextToken();
```

```
            map[v].append(i < n / 2 ? "-" : s).append(" ");
        }
        for(int i = 0; i < 100; i++) {
            System.out.print(map[i]);
        }
        System.out.println();
    }
}
```

4. Result/Output/Writing Summary: