IARE

IARE
INSTITUTE OF
AERONAUTICAL ENGINEERING
(An Autonomous Institute affiliated to JNTUH, Hyderabad)
Dundigal, Hyderabad - 500 043

LABORATORY WORK SHEET

Name of the Student : Mahek Tabeen

Class CSE-C Semester III

Course Code : ACSD08 Course Name : DS lab

Roll Number									
2	3	9	5	1	A	0	5	4	U

Name of the Course Faculty Dr. M. Lakshmi Prasad Faculty ID IARE 10862

Exercise Number : 04 Week Number : 04 Date : 7/11/24

DAY TO DAY EVALUATION:

Marks	Aim / Preparation	Algorithm / Procedure / Performance in the Lab	Source Code / Calculations and Graphs	Program Execution / Results and Error Analysis	Viva - Voce	Total
Max. Marks	4	4	4	4	4	20
Obtained	4	4	4	4	4	20

Signature of Faculty

START WRITING FROM HERE :

4. Divide & Conquer

4.1 Quick sort

```
Import java.util.Scanner;
public class Main {
public static void quickSort(int[] arrays,
                             int start, intend){
    if (static start < end){
        int p index = partition(array, stack, end);
        quick sort = (array, start, Pindex-1);
        quick sort = (array, pindex+1, end;
    }
}
public static int partition(int[] array. int start. intend)
{
    int pivet = array[end];
```

```
        int i = sort - 1;
        for (int j = start; j < end ; j++) {
            if (array [j] <= pivot) {
                i++;
                swap (array, i, j);
            }
        }
        swap (array, i+1 , end);
        return i+1;
    }
    Public static void main (String[] args){
        Scanner sc = new Scanner (System.in);
        int size = sc.nextInt();
        int [] array = new int [size];
        array[i] = sc.nextInt();
    }
    quickSort (array, 0, array.length-1);
    for (int i = 0; i < array.length ; i++) {
        System.out.print(array [i]);
        if (i < array.length - 1){
            System.out.print (", ");
        }
    }
}
```

Input: 10, 80, 30, 90, 40, 50, 70
Output: 10, 30, 40, 50, 70, 80, 90

4.2 Merge Sort

```
import java.util.Scanner;
Public class mergeSort {
    Public static void mergeSort(int[] array, int left,
                                 int right) {
        if (left < right) {
            int middle = left + right/2 ;
            mergeSort = {array. left. middle);
```

ROLL NUMBER :

```
        Merge sort ( array. middle +1. right );
        merge ( array. left. middle . right );
    }
}
Public static void mergesort (int [] array. int left , int
                                middle , int right ){
    int n1 = middle . left + 1;
    int n1 = right - middle;
    int [] left array = new int [n1];
    int [] right Array = new int [n2];
    for (int i = 0; i < n; i++) {
            left Array[i]. array [left + 1];
}
for (int j = 0; j < n2 ; j++) {
    right array [j]. array [middle + 1 + j];
}
int i = 0; j = 0;
int K = left;
while ( i < n : && j < n2 ) {
    if ( left Array [i] <= right Array[ j ]) {
        array [K] = left Array [i];
        i++;
    } else {
        array[K] = right Array [j];
        j++;
    }
K++;
}
mergesort (array, 0, array.length -1);
for (int i = 0; i < arr. length ; i++) {
        System. out. println (array [i]);
        if (i < arr.length -1) {
            System. Out. println (" , ");
        }
    }
}
```

ROLL NUMBER :

Input: 6, 3, 0, 5
Output: 0, 3, 5, 6

4.3 Shell sort

```
import java.util.Scanner;
Public class Main {
    Public static void main(String args[]){
        Scanner sc = new Scanner(System.in){
            int n = sc.nextInt();
            sc.nextline();
            String input = sc.nextline();
            string[] part = input.split(" , ");
            int arr[] = new int[n];
            for(int i=0 ; i<n ; i++){
                arr[i] = Integer.parseInt(parts[i]);
            }
            for(int i = 0; i<n-1 ; i++){
                for(int j = 0; i<n-1; i++){
                    if(arr[j] > arr[j+1]){
                        int temp = arr[j+1];
                        arr[j] = arr[j+1];
                        arr[j+1] = temp;
                    }
                }
            }
            for(int i=0; i<n; i++){
                if(i==n-1){
                    System.out.println(arr[i]);
                }else{
                    System.out.println(arr[i]+" , ");
                }
            }
        }
    }
```

Input: 4
6, 3, 0, 5

output: 0, 3, 5, 6.

4/4