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

LABORATORY WORK BOOK

Name of the Student : N · Ravi Chandrika

Roll Number									
•	3	9	5	1	A	6	7	8	2

Class CSD - B Semester 3rd Semester

Course Code : ACSD11 Course Name DS laboratory

Name of the Course Faculty Mr · A · Srikanth Faculty ID : IARE10652

Exercise Number : 3 Week Number : 3 Date 24-09-24

S. No.	Exercise Number	EXERCISE NAME	MARKS AWARDED					
			Aim/ Preparation	Algorithm / Procedure / Performance in the Lab	Source Code Calculations and Graphs	Program Execution/ Results and Error Analysis	Viva - Voce	Total
			4	4	4	4	4	20
1	3·1	Bubble Soort						
2	3·2	Selection Soort						
3	3·3	Insertion Soat						
4								
5								
6								
7								
8								
9								
10								
11								
12								

N · Ravichandrika
Signature of the Student

Signature of the Faculty

3·1 Aim: Bubble sort is a simplest sorting algorithm that works by repeatdly swapping the adjacent elements if they are in wrong order. This algorithm is not suitable for large data sets.

code:

```
import java.util.*;
public class Main{
        Static void bubblesort(int []arr){
            for(int i=0;i<arr.length-1;i++){
                for(int j=0; j<arr.length-1-i;j++){
                    if(arr[j]>arr[j+1]){
                        int temp=arr[j];
                        arr[j]=arr[j+1];
                        arr[j+1] = temp;
                    }
                }
            }
    }
    Static void printarray(int[] arr){
        for(int i=0;i<arr.length;i++){
            system.out.print(arr[i]);
            if(i<arr.length-1){
                System.out.print(", ");
            }
    }
}
public static void main(String args[]){
    Scanner sc = new Scanner(system.in);
    int n= sc.nextInt();
    String m= sc.nextLine();
    String[] input = sc.nextLine().split(" ");
    int[] arr = new int[n];
    for(int i=0; i<n; i++){
        arr[i] = Integer.parseInt (input[i]):
    }
   bubblesort(arr);
   printarray(arr);
  }
}.
```

Input: 4
6 3 0 5

Output: 0,3,5,6.

8.2 Aim: Selection sort is an simple and efficient sorting algorithm that works by repeatedly selecting the smallest element from the unsorted position of the list and moving it to the sorted portion of the list.

Code:

```
import java.util.Scanner;
public class Main{
   public static void main(String args[]){
         Scanner scanner = new Scanner(System.in);
         int size = scanner.nextInt();
```

```
    scanner.nextLine();
    String input = scanner.nextLine();
    String[] inputArr = input.split(" ");
    int[] arr = new int[size];
    for(int i=0; i<size ; i++){
        arr[i] = Integer.parseInt(inputArr[i]);

    }
   for (int i=0; i<size -1; i++){
      int minIndex = i;
      for (int j=i+1 ; j< size; j++){
          if(arr[j] < arr[minIndex]) {
              minIndex = j;
          }
      }
      int temp = arr[minIndex];
      arr[minIndex] = arr[i];
      arr[i] = temp;
   }
   for (int i=0; i<size; i++) {
      if (i! =0) {
         System.out.print(",");
      }
      System.out.print(arr[i]);
   }
  }
}
```

Input: 4

6 3 0 5

Output: 0,3,5,6.

Aim: Insertion Sort

```
import java.util.Scanner;
public class Main{.
    public static void main (String args[]) {
        Scanner scanner = new Scanner (System.in);
        int size = scanner.nextInt();
        scanner.nextLine();
        String input = scanner.nextLine();
        String] inputArr = input.Split(" ");
        int[] arr = new int[size];
        for (int i=0 ; i<size ; i++) {
            arr[i] = Integer.ParseInt(inputArr[i]);
        }
        for (int i=1; i<size; i++){
            int key = arr[i];
            int j= i-1;
            while (j>=0 && arr[j] > key) {
                arr[j+1] = arr[j];
                j = j-1;
            }
            arr[j+1] = key;
        }
        for (int i=0 ; i< size ; i++) {
```

```
        if(i!=0){
              system.out.print(" ,");

        }
        System.out.print (arr[i]);
    }
  }.
}.
```

Input: 4

6 3 0 5.

Output: 0, 3, 5, 6.