**Program :**

import java.util.Scanner;

public class SelectionSort {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the number of elements : ");

        int n = sc.nextInt();

        int[] arr = new int[n];

        System.out.print("Enter the Elements : ");

        for(int i=0;i<n;i++) arr[i]= sc.nextInt();

        int swaps = selectnSort(arr);

        System.out.println("\nSorted array :");

        for (int k=0;k<arr.length;k++) {

            System.out.print("\t"+arr[k]);

        }

        System.out.println("\n\nNumber of swaps : "+ swaps);

    }

    private static int selectnSort(int[] arr) {

        int swaps=0;

        System.out.println("\n\*\*\*Selection Sort\*\*\* \nPasses :");

        for(int i=0;i<arr.length;i++){

            print(arr,i);

            int min = i;

            for(int j=i+1;j<arr.length;j++){

                if(arr[min]>arr[j]) min = j;

            }

            int temp = arr[i];

            arr[i] = arr [min];

            arr[min] = temp;

            swaps++;

        }

        return swaps;

    }

    private static void print(int[] arr, int i) {

        for (int k=0;k<arr.length;k++) {

            System.out.print("\t"+arr[k]);

            if(i-1 == k) System.out.print("   |");

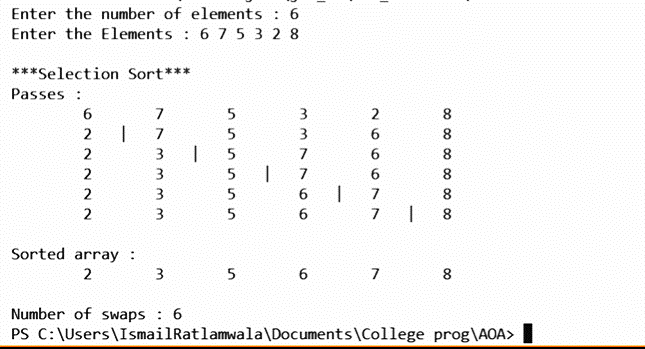
        }

        System.out.println();

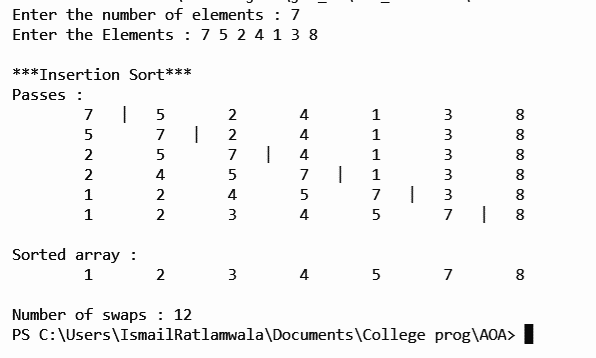
    }

}

**Output :**

****

**Output :**



**Program :**

import java.util.Scanner;

public class InsertionSort{

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the number of elements : ");

        int n = sc.nextInt();

        int[] arr = new int[n];

        System.out.print("Enter the Elements : ");

        for(int i=0;i<n;i++) arr[i]= sc.nextInt();

        int swaps = IntsertnSort(arr);

        System.out.println("\nSorted array :");

        for (int k=0;k<n;k++) {

            System.out.print("\t"+arr[k]);

        }

        System.out.println("\n\nNumber of swaps : "+ swaps);

        sc.close();

    }

    private static int IntsertnSort(int[] arr) {

        int swaps=0;

        System.out.println("\n\*\*\*Insertion Sort\*\*\* \nPasses :");

        for(int i=1;i<arr.length;i++){

            print(arr,i);

            int temp = arr[i];

            int j;

            for(j=i-1;j>=0;j--){

                if(arr[j]>temp){

                    arr[j+1] = arr[j];

                    swaps++;

                }

                else break;

            }

            arr[j+1]=temp;

        }

        return swaps;

    }

    private static void print(int[] arr, int i) {

        for (int k=0;k<arr.length;k++) {

            System.out.print("\t"+arr[k]);

            if(i-1 == k) System.out.print("   |");

        }

        System.out.println();

    }

}

**Program :**

import java.util.Scanner;

public class QuickSort {

    static int swaps = 0;

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the number of elements : ");

        int n = sc.nextInt();

        int[] arr = new int[n];

        System.out.print("Enter the Elements : ");

        for(int i=0;i<n;i++) arr[i]= sc.nextInt();

        System.out.println("Passes :");

        quickSort(arr, 0, arr.length-1);

        System.out.println("\nSorted array :");

        for (int k=0;k<n;k++) {

            System.out.print("\t"+arr[k]);

        }

        System.out.println("\n\nNumber of swaps : "+ swaps);

        sc.close();

    }

    private static void quickSort(int[] arr, int lb, int ub) {

        if(lb<ub){

            for(int i=0;i<arr.length;i++) System.out.print(arr[i]+" ");

            System.out.println();

            int piv = partition(arr, lb, ub);

            quickSort(arr, lb, piv-1);

            quickSort(arr, piv+1, ub);

        }

    }

    private static int partition(int[] arr, int lb, int ub) {

        int start=lb, end=ub, pivot=arr[lb];

        while(start<end){

            while(start<ub && arr[start]<=pivot) start++;

            while(arr[end]>pivot) end--;

            if(start<end) swap(arr,start,end);

        }

        swap(arr, lb, end);

        return end;

    }

    private static void swap(int[] arr, int a, int b) {

        int temp = arr[a];

        arr[a] = arr[b];

        arr[b] = temp;

        swaps++;

    }

}

**Program :**

import java.util.Scanner;

public class MergeSort {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the number of elements : ");

        int n = sc.nextInt();

        int[] arr = new int[n];

        System.out.print("Enter the Elements : ");

        for(int i=0;i<n;i++) arr[i]= sc.nextInt();

        System.out.println("Passes :");

        mergeSort(arr, 0, arr.length-1);

        System.out.println("\nSorted array :");

        for (int k=0;k<n;k++) {

            System.out.print("\t"+arr[k]);

        }

    }

    private static void mergeSort(int[] arr, int lb, int ub) {

        if (lb < ub) {

            for (int l = lb; l <= ub; l++)

                System.out.print(arr[l] + " ");

            System.out.println();

            int mid = (lb + ub) / 2;

            mergeSort(arr, lb, mid);

            mergeSort(arr, mid + 1, ub);

            merge(arr, lb, mid, ub);

        }

    }

    private static void merge(int arr[], int lb, int mid, int ub) {

        int i = lb, j = mid + 1, k = 0;

        int[] tempArr = new int[ub-lb+1];

        while (i <= mid && j <= ub) {

            if (arr[i] < arr[j]) {

                tempArr[k] = arr[i];

                i++;

            } else {

                tempArr[k] = arr[j];

                j++;

            }

            k++;

        }

        if (i > mid) {

            while (j <= ub) {

                tempArr[k] = arr[j];

                j++;

                k++;

            }

        } else {

            while (i <= mid) {

                tempArr[k] = arr[i];

                i++;

                k++;

            }

        }

        for (k = lb; k <= ub; k++) { //copy tempArr to main arr

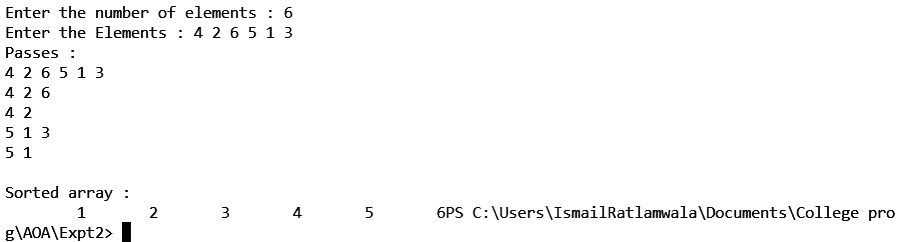
            arr[k] = tempArr[k-lb];

        }

    }

}

**Output :**



**Output :**

