1. QuickSort

```
Algorithm :: QuickSort(a[0,...n-1],low,high)
//Sorts the data in ascending order by following divide and conquer approach

//INPUT  :: An array a[0...n-1] of orderable elements
//OUTPUT :: An array a[0...n-1] or ordered   elements

   key= a[low]
    i = low + 1
    j = high

   while(true)
       while a[i]<key && i<high then
           i++
       while a[j]>key && j>low  then
         j--
      if i<j then
         swap a[i] and a[j]
      otherwise
         swap a[low] and a[j]
         return j
```

Program in java

```
+++++++++++++++
import java.util.*;

class QuickSortApp
{
      public static void main(String[] args)
      {
            Scanner scan = new Scanner(System.in);

            System.out.print("Enter the size of an array :: ");
            int n = scan.nextInt();

            int[] arr = new int[n];

            for (int i=0;i<=n-1;i++ )
            {
                  System.out.print("Enter the array element:: ");
                  arr[i] = scan.nextInt();
            }

            System.out.println("Array before Sorting :: "+Arrays.toString(arr));

            quickSort(arr,0,n-1);

            System.out.println("Array after Sorting :: "+Arrays.toString(arr));

      }

      public static void quickSort(int[] arr,int low,int high)
      {
            int j;
            if (low<high)
            {
                   j = partition(arr,low,high);
```

```
                quickSort(arr,low,j-1);
                quickSort(arr,j+1,high);
          }
     }

     public static int partition(int[] arr,int low,int high)
     {
          int key = arr[low];
          int i   = low+1 ;
          int j   = high;

          while (true)
          {
               while (arr[i]<key && i<high)
                    i++;

               while(arr[j]>key && j>low)
                    j--;

               if (i<j)
               {
                    //swap a[i] and a[j]
                    int temp = arr[i];
                    arr[i]   = arr[j];
                    arr[j]   = temp;
               }
               else
               {
                    //swap a[low] and a[j]
                    int temp = arr[low];
                    arr[low] = arr[j];
                    arr[j] = temp;
                    System.out.println(j);
                    return j;
               }
          }
     }
}

Output
Enter the size of an array :: 8

Enter the array element:: 55
Enter the array element:: 33
Enter the array element:: 11
Enter the array element:: 99
Enter the array element:: 88
Enter the array element:: 22
Enter the array element:: 44
Enter the array element:: 77

Array before Sorting :: [55, 33, 11, 99, 88, 22, 44, 77]
j= 4
j= 1
j= 2
j= 6
Array after Sorting :: [11, 22, 33, 44, 55, 77, 88, 99]
```