**SORTING**

**BUBBLE SORT**

1 import java.util.\*;  
 2 public class BubbleSort{   
 3 public static void main(String[] args) {   
 4 Scanner sc = new Scanner(System.in);  
 5 System.out.print("Enter the size of array :");  
 6 int size = sc.nextInt();  
 7 int arr[] =new int[size];  
 8 for(int i = 0; i<size; i++){  
 9 System.out.print("\nEnter element no "+(i+1)+" :");  
10 arr[i] = sc.nextInt();  
11 }   
12 System.out.println("Array Before Sorting");   
13 for(int i=0; i < arr.length; i++){   
14 System.out.print(arr[i] + " ");   
15 }   
16 System.out.print("\n");   
17 bubble\_Sort(arr);   
18 System.out.println("Array After Sorting");   
19 for(int i=0; i < arr.length; i++){   
20 System.out.print(arr[i] + " ");   
21 }   
22   
23 }   
24 static void bubble\_Sort(int[] arr) {   
25 int len = arr.length;   
26 int temp = 0;   
27 for(int i=0; i < len; i++){   
28 for(int j=1; j < (len-i); j++){   
29 if(arr[j-1] > arr[j]){   
30 temp = arr[j-1];   
31 arr[j-1] = arr[j];   
32 arr[j] = temp;   
33 }   
34   
35 }   
36 }   
37   
38 }  
39   
40 }

**SELECTION SORT**

1 import java.util.\*;  
 2 class SelectionSort{  
 3 public static void main(String args[]){  
 4 Scanner sc = new Scanner(System.in);  
 5 System.out.print("Enter size of an array : ");  
 6 int l = sc.nextInt();  
 7 int arr[] = new int[l];  
 8 System.out.println("Enter elements of array");  
 9 for(int i=0; i<l ; i++){  
10 System.out.print((i+1)+" : ");  
11 arr[i] = sc.nextInt();  
12 }  
13 selectionSort(arr);  
14 for(int i=0; i<l ; i++){  
15 System.out.print(arr[i]+" ");  
16 }  
17 }  
18 public static void selectionSort(int[] arr){  
19 int l = arr.length;  
20 for(int i=0; i<l-1 ; i++){  
21 int a = min(i,l,arr);  
22 int temp = arr[a];   
23 arr[a] = arr[i];   
24 arr[i] = temp;  
25 }  
26 }  
27 public static int min(int n , int l , int[] arr){  
28 int a = n;   
29 for (int j = n+1; j < l; j++)   
30 if (arr[j] < arr[a])   
31 a = j;  
32 return a;  
33 }  
34 }

**QUICK SORT**

1 import java.util.\*;  
 2 class Quicksort{  
 3 static int c = 0;  
 4 public static void main(String args[]){  
 5 Scanner sc =new Scanner(System.in);  
 6 int beg = 0;  
 7 int end;  
 8 System.out.print(“Enter size of an array : “);  
 9 int l = sc.nextInt();  
10 end = l-1;  
11 int arr[] = new int[l];  
12 System.out.println(“Enter elements of array”);  
13   
14 for(int i=0; i<l ; i++){  
15 System.out.print((i+1)+” : “);  
16 arr[i] = sc.nextInt();  
17 }  
18   
19 System.out.println(“Entered elements without sorting”);  
20 for(int i=0; i<l ; i++){  
21 System.out.print(arr[i]+” “);  
22 }  
23   
24 System.out.println(“Entered elements after sorting”);  
25 quick\_sort(arr,beg,end);  
26 for(int i=0; i<arr.length ; i++){  
27 System.out.print(arr[i]+” “);  
28 }   
29   
30   
31 }  
32   
33 public static void quick\_sort(int []arr,int beg,int end){  
34 if(beg<end){  
35 int num = partition(arr,beg,end);  
36 quick\_sort(arr,beg,num-1);  
37 quick\_sort(arr,num+1,end);   
38 }   
39   
40   
41 }  
42   
43 public static int partition(int arr[], int beg, int end){  
44 int loc = beg;  
45 int l = beg;  
46 int r = end;  
47 int flag = 0;  
48 while(flag==0){  
49 while(arr[loc]<=arr[r] && loc!=r){  
50 r=r-1;  
51 }  
52 if(loc==r){  
53   
54 flag=100;  
55 }  
56 else if(arr[loc]>arr[r]){  
57 int temp = arr[loc];  
58 arr[loc] = arr[r];  
59 arr[r] = temp;  
60 loc=r;  
61 }  
62   
63 if(flag==0){  
64 while(arr[loc]>=arr[l] && loc!=l){  
65 l=l+1;  
66 }  
67 if(loc==l){  
68   
69 flag=100;  
70 }  
71 else if(arr[loc]<arr[l]){  
72 int temp = arr[loc];  
73 arr[loc] = arr[l];  
74 arr[l] = temp;  
75 loc=l;  
76 }  
77 }  
78 }  
79 return loc;  
80   
81 }  
82 }  
83

**BINAY 19CSU370**

**IOT-A**