**//BUBBLE SORT**

class BubbleSortArr {

public static void main(String[] arg){

int[] c ={8,4,3,5,6,7};

int flag =0;

for(int i=0;i<c.length-1;i++) {

flag=0;

for(int j=0;j<c.length-1-i;j++) {

if(c[j]>c[j+1]) {

flag=1;

c[j] = c[j]+c[j+1];

c[j+1] = c[j]-c[j+1];

c[j] = c[j]-c[j+1];

}

//loopCount++;

}

if(flag==0)

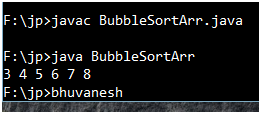
break;

}

for(int p=0;p<c.length;p++)

System.out.print(c[p]+" ");

}



**//SELECTION SORT**

import java.util.Scanner;

class SelectionSort {

public static void main(String[] arg){

int min=0;

Scanner ip = new Scanner(System.in);

System.out.println("enter the size of array");

int size = ip.nextInt();

int[] a = new int[size];

System.out.println("enter values for array");

for(int i=0;i<size;i++)

a[i] = ip.nextInt();

for(int i=0;i<size-1;i++) {

min=i;

for(int j=i+1;j<size;j++)

if(a[min]>a[j])

min=j;

if(i!=min) {

a[i] = a[i]+a[min];

a[min] = a[i]-a[min];

a[i] = a[i]-a[min];

}

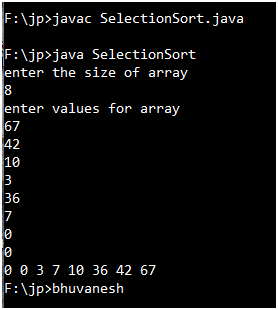
}

for(int p=0;p<a.length;p++)

System.out.print(a[p]+" ");

}

}



**//INSERTION SORT**

import java.util.Scanner;

class InsertionSort {

public static void main(String[] arg){

int value=0,hole=1,j=0,flag=0;

Scanner ip = new Scanner(System.in);

System.out.println("enter the size of array");

int size = ip.nextInt();

int[] a = new int[size];

System.out.println("enter values for array");

for(int i=0;i<size;i++)

a[i] = ip.nextInt();

for(int i=1;i<size;i++) {

value=a[i];

hole =i; flag=1;

j=i;

while(j>0 && a[j-1]>value) {

a[hole--] = a[--j];

flag=0;

}

if(flag==0)

a[j]=value;

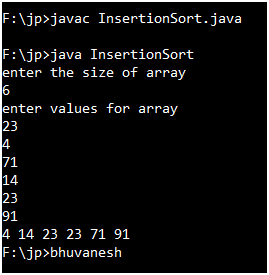
}

for(int p=0;p<a.length;p++)

System.out.print(a[p]+" ");

}

}



**//QUICK SORT**

class Qsort {

static int[] a ={1,2,1,6,8,5,3,4};

public static void main(String[] arg) {

quickSort(0,a.length-1);

for(int p=0;p<a.length;p++)

System.out.print(a[p]+" ");

}

public static void quickSort(int start, int end) {

if(start<end){

int pIndex = partition(start,end);

//System.out.println(pIndex);

quickSort(start,pIndex-1);

quickSort(pIndex+1,end);

}

}

public static int partition(int start, int end){

int pivot =a[end];

int pIndex=start, temp=0;

for(int i=start;i<=end;i++) {

if(a[i]<=pivot){

temp = a[i];

a[i] = a[pIndex];

a[pIndex] = temp;

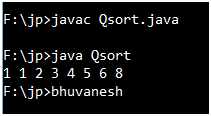
pIndex++;

}

}

return pIndex-1;

}



**//HEAP SORT**

import java.util.Scanner;

class HeapSort {

public static void main(String[] arg) {

Scanner ip = new Scanner(System.in);

System.out.println("enter the size of array");

int n = ip.nextInt();

int[] a = new int [n];

int[] b = new int [n];

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

a[i] = ip.nextInt();

int parent =0;

int lc = (2\*parent)+1;

int rc = (2\*parent)+2;

System.out.println("build heap into the array");

for(int i=0;i<n;i++) {

if((2\*i+2)<n) {

if(a[i]>a[2\*i+1]) {

a[i] = a[2\*i+1]+a[i];

a[2\*i+1] = a[i]-a[2\*i+1];

a[i] = a[i]-a[2\*i+1];

}

if(a[i]>a[2\*i+2]) {

a[i] = a[2\*i+2]+a[i];

a[2\*i+2] = a[i]-a[2\*i+2];

a[i] = a[i]-a[2\*i+2];

}}}

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

System.out.print(a[i]+" ");

System.out.println();

System.out.println("find the min heap");

int j=n-1;

for(int i=0;i<n;) {

if(a[parent]>a[lc]) {

a[parent] = a[parent]+a[lc];

a[lc] = a[parent]-a[lc];

a[parent] = a[parent]-a[lc];

}

if(a[parent]>a[rc]) {

a[parent] = a[parent]+a[rc];

a[rc] = a[parent]-a[rc];

a[parent] = a[parent]-a[rc];

}

b[i++]=a[parent];

a[parent]=a[j--];

if(j==0){

b[i]=a[j]; break;}

}

for(int p=0;p<a.length;p++)

System.out.print(b[p]+" ");

}

}

