

```

import java.util.*;
class SelSort {
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int[] a=new int[n];
        for(int i=0;i<n;i++) a[i]=sc.nextInt();
        for(int i=0;i<n-1;i++){
            int min=i;
            for(int j=i+1;j<n;j++) if(a[j]<a[min]) min=j;
            int t=a[min]; a[min]=a[i]; a[i]=t;
        }
        for(int x:a) System.out.print(x+" ");
        sc.close();
    }
}

```

// Q1(b) Insertion Sort

```

import java.util.*;
class InsSort {
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int[] a=new int[n];
        for(int i=0;i<n;i++) a[i]=sc.nextInt();
        for(int i=1;i<n;i++){
            int key=a[i], j=i-1;
            while(j>=0 && a[j]>key){a[j+1]=a[j]; j--;}
            a[j+1]=key;
        }
        for(int x:a) System.out.print(x+" ");
        sc.close();
    }
}

```

// Q1(c) Bubble Sort

```

import java.util.*;
class BubSort {
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int[] a=new int[n];
        for(int i=0;i<n;i++) a[i]=sc.nextInt();
        for(int i=0;i<n-1;i++)
            for(int j=0;j<n-i-1;j++)
                if(a[j]>a[j+1]){
                    int t=a[j];a[j]=a[j+1];a[j+1]=t;
                }
        for(int x:a) System.out.print(x+" ");
        sc.close();
    }
}

```

// Q1(d) Merge Sort

```

import java.util.*;

```

```

class MergeSort {
    static void merge(int[] a,int l,int m,int r){
        int n1=m-l+1,n2=r-m;
        int[] L=new int[n1];
        int[] R=new int[n2];
        for(int i=0;i<n1;i++) L[i]=a[l+i];
        for(int i=0;i<n2;i++) R[i]=a[m+1+i];
        int i=0,j=0,k=l;
        while(i<n1 && j<n2) a[k++]=(L[i]<=R[j]?L[i++]:R[j++]);
        while(i<n1) a[k++]=L[i++];
        while(j<n2) a[k++]=R[j++];
    }
    static void mergesort(int[] a,int l,int r){
        if(l<r){
            int m=(l+r)/2;
            mergesort(a,l,m);
            mergesort(a,m+1,r);
            merge(a,l,m,r);
        }
    }
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int[] a=new int[n];
        for(int i=0;i<n;i++) a[i]=sc.nextInt();
        mergesort(a,0,n-1);
        for(int x:a) System.out.print(x+" ");
        sc.close();
    }
}

```

```

// Q1(e) Quick Sort
import java.util.*;
class QuickSort {
    static int partition(int[] a,int l,int r){
        int p=a[r],i=l-1;
        for(int j=l;j<r;j++) if(a[j]<p){
            i++; int t=a[i]; a[i]=a[j]; a[j]=t;
        }
        int t=a[i+1]; a[i+1]=a[r]; a[r]=t;
        return i+1;
    }
    static void quicksort(int[] a,int l,int r){
        if(l<r){
            int pi=partition(a,l,r);
            quicksort(a,l,pi-1);
            quicksort(a,pi+1,r);
        }
    }
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int[] a=new int[n];
        for(int i=0;i<n;i++) a[i]=sc.nextInt();
        quicksort(a,0,n-1);
    }
}

```

```

        for(int x:a) System.out.print(x+" ");
        sc.close();
    }
}

// Q2 Improved Selection Sort (Min & Max)
import java.util.*;
class ImprovedSel {
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int[] a=new int[n];
        for(int i=0;i<n;i++) a[i]=sc.nextInt();
        int l=0,r=n-1;
        while(l<r){
            int min=l,max=r;
            for(int i=l;i<=r;i++){
                if(a[i]<a[min]) min=i;
                if(a[i]>a[max]) max=i;
            }
            int t=a[l];a[l]=a[min];a[min]=t;
            if(max==l) max=min;
            t=a[r];a[r]=a[max];a[max]=t;
            l++; r--;
        }
        for(int x:a) System.out.print(x+" ");
        sc.close();
    }
}

```