

Bubble sort.
Selection sort.
Inbuilt sort

★ Bubble sort:-

each its move largest number at end.
each its move smallest number at beginning.

★ Selection sort:-

Arrays.sort ();

→ ascending order.
↳ default behaviour in Java.

gn. section Sol:-

5
9 4 3 8 7
0 1 2 3 4

→ 3 4 7 8 9

9 4 3 8 7

3 4 7 8 9

3 4 9 7

3 4 8 9 7

3 4 7 8 9

7

3 4 7 8 9

A 2 3 4 5 6 7 8 9 10

7 9

3 4 7 8 9

back

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] A = new int[n];
    for(int i = 0; i < n; i++){
        A[i] = scn.nextInt();
    }

    //9 4 3 8 7
    //insertion sort
    for(int i = 1; i < n; i++){
        for(int j = i-1; j >= 0; j--){
            if(A[j+1] < A[j]){
                int temp = A[j+1];
                A[j] = A[j+1];
                A[j+1] = temp;
            }
            else{
                break;
            }
        }
    }

    for(int i = 0; i < n; i++){
        System.out.print(A[i] + " ");
    }
}
```

Insertion sort with swap function.

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int [] A = new int[n];
    for(int i = 0; i < n; i++){
        A[i] = scn.nextInt();
    }
    //9 4 3 8 7
    //insertion sort
    for(int i = 1; i < n; i++){
        for(int j = i-1; j >= 0; j--){
            if(A[j+1] < A[j]){
                swap(A, j, j+1);
            }
            else{
                break;
            }
        }
    }
    for(int i = 0; i < n; i++){
        System.out.print(A[i] + " ");
    }
}
```

Handwritten notes showing indices and values:

Index	Value
0	9
1	3
2	9
3	8
4	7

Below the table, there are handwritten annotations:

- Under index 0, the value 9 is written, and a red arrow points from it to index 1.
- Under index 2, the value 9 is written, and a red circle is drawn around it.

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int [] A = new int[n];
    for(int i = 0; i < n; i++){
        A[i] = scn.nextInt();
    }
    //9 4 3 8 7
    //insertion sort
    for(int i = 1; i < n; i++){
        for(int j = i-1; j >= 0; j--){
            if(A[j+1] < A[j]){
                swap(A, j, j+1);
            }
            else{
                break;
            }
        }
    }
    for(int i = 0; i < n; i++){
        System.out.print(A[i] + " ");
    }
}
```

Decreasing order sorting using inbuilt method.

```
7 public class Solution {
8
9     public static void main(String[] args) {
10         Scanner scn = new Scanner(System.in);
11         int n = scn.nextInt();
12
13         Integer [] A = new Integer[n];
14
15         for(int i = 0; i < n; i++)
16             A[i] = scn.nextInt();
17
18         Arrays.sort(A, Collections.reverseOrder());
19
20         for(int i = 0; i < n; i++){
21             System.out.print(A[i] + " ");
22         }
23     }
24 }
25 }
```

sc. O.C.I. ✗ java.

Comparator:



used to compare 2 obj.

Comparator:

↳ compare function.

return →

(int)

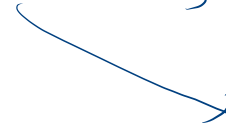
Integer a, Integer b



1 (positive) ~> swap



0



-1 (negative)

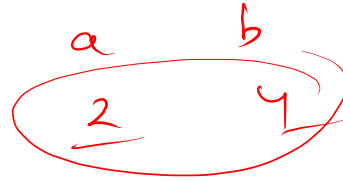
```
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner scn = new Scanner(System.in);  
        int n = scn.nextInt();  
  
        Integer [] A = new Integer[n];  
  
        for(int i = 0; i < n; i++)  
            A[i] = scn.nextInt();  
        //how you can create your own comparator  
        Comparator<Integer> myComp = new Comparator<Integer>(){  
            public int compare(Integer a, Integer b){  
                return b-a; //logic  
            }  
        };  
  
        Arrays.sort(A, myComp);  
  
        for(int i = 0; i < n; i++){  
            System.out.print(A[i] + " ");  
        }  
    }  
}
```

```
public int compare(Integer a, Integer b){  
    return b-a; //logic  
}
```

asc

a

b



$b-a \rightarrow$ desc.

$a-b \rightarrow$ incre.

$2-4$ -ve.

$a < b$.


```
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner scn = new Scanner(System.in);  
        int n = scn.nextInt();  
  
        Integer [] A = new Integer[n];  
  
        for(int i = 0; i < n; i++){  
            A[i] = scn.nextInt();  
  
            Arrays.sort(A, (a,b)-> b-a);  
  
            for(int i = 0; i < n; i++){  
                System.out.print(A[i] + " ");  
            }  
        }  
    }  
}
```