

```

15 //Creating own comparator
16 Comparator<Integer> myComp = new Comparator<Integer>(){
17     public int compare(Integer a, Integer b){
18         return b-a;
19     }
20 };
21

```

logic  $\rightarrow$  compare.

- 0
- +ve  $\rightarrow$  swapping
- ve

$\begin{matrix} 5 & 7 \\ \cancel{7} & \cancel{5} \\ a & b \end{matrix}$ 
 $\dots$ 
 $\begin{matrix} 5 & 7 \end{matrix}$

$\checkmark$   
 $\rightarrow$  asc.  $\rightarrow$  a-b  
 $7-5 = (2) +ve$

$\begin{matrix} 7 & 5 \\ a & b \end{matrix}$

desc  $\rightarrow$  b-a       $5-7 = (-2)$

---

$\begin{matrix} 7 & 5 \\ \cancel{7} & \cancel{5} \\ a & b \end{matrix}$ 
 $7-5 = 2$

Summary.

$\hookrightarrow$  return a-b  $\rightarrow$  asc.  
 return b-a  $\rightarrow$  desc.

Arrays.sort ( A, start, end )

↳ range sorting.  
(inbuilt).

9	2	1	6	4	3	8	9	4
0	1	2	3	4	5	6	7	8

(Elements 1, 6, 4, 3 from index 2 to 5 are highlighted in blue and grouped with a bracket below.)

Arrays.sort ( A, 2, 6 )

$\Rightarrow [2, 6) \Rightarrow [2, 5]$

$\rightarrow$

9	2	1	3	4	6	8	9	4
0	1	2	3	4	5	6	7	8

(Elements 3, 4, 6, 8 from index 2 to 5 are enclosed in a green box.)

Arrays.sort ( A, (2,6) )

$[2,6) \Rightarrow [2,5]$

Range     Sorting

9     2  
0     1

4	3	1	6	8	4
2	3	4	5	6	7

9   2   1   3   4   6 } 8   4

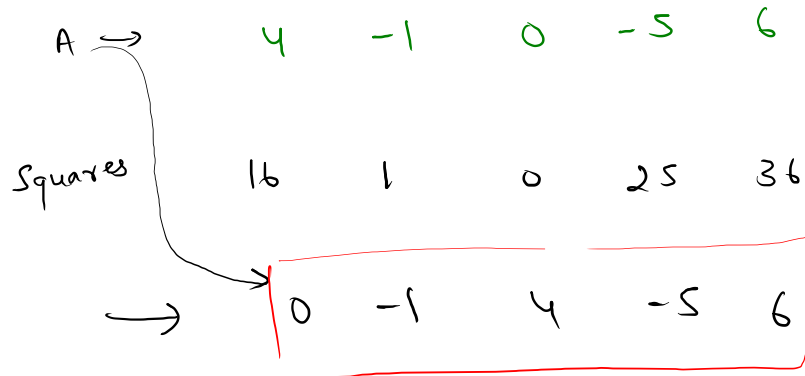
# Sort the array according to their Square of each element

Sample Input 0

```
5
4 -1 0 -5 6
```

Sample Output 0

```
0 -1 4 -5 6
```



asc

✓  $a - b$

$a * a - b * b$

eg.

$-4$        $2$   
 $a$        $b$

$16 - 4 = 1$

1150

$a$	$b$
$\cancel{-4} 2$	$\cancel{2} -4$

$$\underline{asc.} \rightarrow a - b$$
$$-4 - 2 = -6$$

$$16 - 4 = \textcircled{12}$$

+ve

asc. sg.  $\rightarrow a * a - b * b.$

eg.  $\rightarrow$  desc. (sg):

2	-4
<del>-4</del>	<del>2</del>
a	b

$$\begin{array}{cc} -4 & 2 \\ a & b \end{array}$$

$$b-a \Rightarrow 2 - (-4) = 2+4 = \textcircled{6} \begin{matrix} +ve \\ - \end{matrix}$$

desc. 2 - 4

sort desc: (sg).

a	b
-4	2

$$\underline{b \times b - a \times a}$$

$$2 \times 2 = 16$$

$$= 4 - 16 = -12$$

$$\begin{cases} \text{asc.} & a-b \\ \text{desc.} & b-a \end{cases}$$

$$\begin{array}{cc} a & b \\ -2 & 2 \end{array}$$

asc.  $\checkmark \{-2, 2\}$   $a-b = -2-2 = -4$

desc  $\{2, -2\}$   $b-a = 2 - (-2) = 4$

$\xrightarrow{a \leftrightarrow b}$   $\{2, -2\}$  <sub>actual</sub>

asc. (sq)  $\underline{\underline{a^2 - b^2}} = 4 - 4 = 0$

$\swarrow \quad \searrow$   
 $a-b$   
 $b-a$

```
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int n = scn.nextInt();
9         Integer [] A = new Integer[n];
10        for(int i = 0; i < n; i++){
11            A[i] = scn.nextInt();
12        }
13        //sort
14        Comparator<Integer> myComp = new Comparator<Integer>(){
15            public int compare(Integer a, Integer b){
16                return a*a - b*b;
17            }
18        };
19        Arrays.sort(A, myComp);
20        //print
21        for(int i = 0; i < n; i++){
22            System.out.print(A[i] + " ");
23        }
24    }
25 }
```

$$\begin{array}{r} -2 \qquad 4 \\ \hline \cancel{4} \qquad \cancel{-2} \\ a \qquad b \end{array}$$

return  $\xrightarrow{\text{compare}}$

(+ve)

asc.  
desc.

$$a - b$$

$$4 + 2 = (6)$$

4	-2
a	b

$$-2 - 4 = (-6)$$

$\rightarrow b - a$

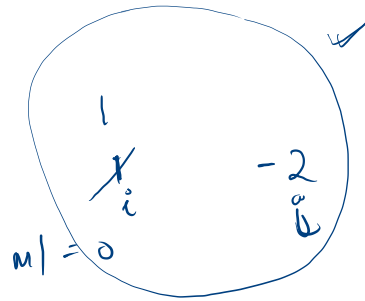


Adivya

```
1 import java.util.*;
2
3 public class Solution {
4
5     public static void main(String[] args) {
6         Scanner scn = new Scanner(System.in);
7         int n = scn.nextInt();
8         int []A = new int[n];
9         for(int i = 0; i < n; i++){
10             A[i] = scn.nextInt();
11         }
12         for(int i = 0; i < n-1; i++){
13             int minIdx = i;
14             for(int j = i+1; j < n; j++){
15                 if(A[j] * A[j] < A[minIdx] * A[minIdx]){
16                     minIdx = j;
17                 }
18             }
19             int tmp = A[i];
20             A[i] = A[minIdx];
21             A[minIdx] = tmp;
22         }
23         for(int i = 0; i < n; i++){
24             System.out.print(A[i] + " ");
25         }
26     }
27 }
```

-2 1 0 -1  
i j k j

$min = \cancel{1} \cancel{1} 0$



$\begin{pmatrix} 2 \\ 1 \end{pmatrix}$

$\frac{-2}{2} = 1$

1 4

# Sort Array By Parity

Given an integer array `nums[]`, move all the **even** integers at the beginning of the array followed by all the **odd** integers in non- decreasing order.

Sample Input 0

```
4
3 1 2 4
```

Sample Output 0

```
2 4 1 3
```

even.... odd----

5 2 3 6 8 4 7 1



2 4 6 8 1 3 5 7

~~4~~ 8 3 7 1 5  
~~5~~ 2 ~~3~~ 6 ~~4~~ ~~1~~ ~~7~~  
0 1 2 3 4 5 6 7

even.... odd----



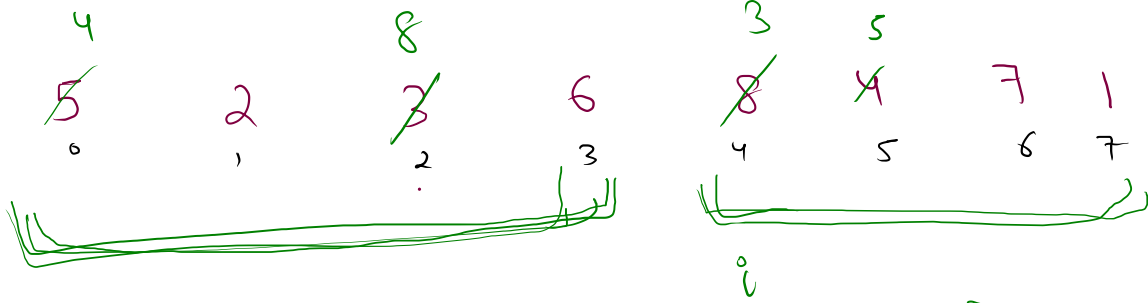
while (i <= j)

i  
↓  
else

4 2 8 6 | 3 7 1 5  
j i

2 4 6 8 1 3 5 7

ans.



$$i \leq j$$

[ , ]

j

```
if ( A[i] → even )
{
    i++
}
```

```
elseif ( A[j] → odd )
{
    j--
}
```

```
else { swap ( i, j )
      i++
      j--
    }
```

2 4 6 8    1 3 5 7

---



```

4 public class Solution {
5     public static void main(String[] args) {
6         Scanner scn = new Scanner(System.in);
7         int n = scn.nextInt();
8         Integer [] A = new Integer[n];
9         for(int i = 0; i < n; i++){
10             A[i] = scn.nextInt();
11         }
12         int i = 0;
13         int j = n-1;
14         while(i <= j){
15             if(A[i] % 2 == 0){
16                 i++;
17             }
18             else if(A[j] % 2 != 0){
19                 j--;
20             }
21             else{
22                 int tmp = A[i];
23                 A[i] = A[j];
24                 A[j] = tmp;
25                 i++;
26                 j--;
27             }
28         }
29         Arrays.sort(A,0,i);
30         Arrays.sort(A,i,n);
31         //print
32         for( i = 0; i < n; i++){
33             System.out.print(A[i] + " ");
34         }
35     }
36 }
37

```

2 1 1 7 1 3 5 2 4 2 6 7

# Sort an array in wave form 1

$arr[0] \geq arr[1] \leq arr[2] \geq arr[3] \leq arr[4] \geq \dots$

$$A[0] \geq A[1] \leq A[2] \geq A[3] \leq A[4] \geq A[5] \dots$$

Sample Input 0

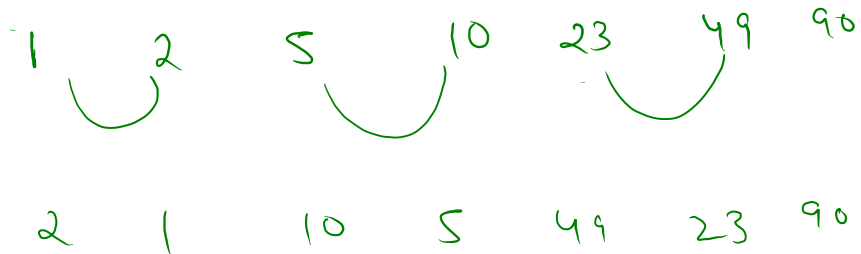
7  
10 90 49 2 1 5 23

Sample Output 0

2 1 10 5 49 23 90

try. code

\*\*\*  
logic → 2 step.  
1. → sort → asc.  
2. → swap alternate pairs.



```

1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int n = scn.nextInt();
9         int [] A = new int[n];
10        for(int i = 0; i < n; i++){
11            A[i] = scn.nextInt();
12        }
13        //logic
14        //1. Sort
15        Arrays.sort(A);
16        //2. Swap alternate
17
18        for(int i = 0; i < n-1; i += 2){
19            int tmp = A[i];
20            A[i] = A[i+1];
21            A[i+1] = tmp;
22        }
23        //print
24        for(int i = 0; i < n; i++){
25            System.out.print(A[i] + " ");
26        }
27    }
28 }

```

## Sample Input 0

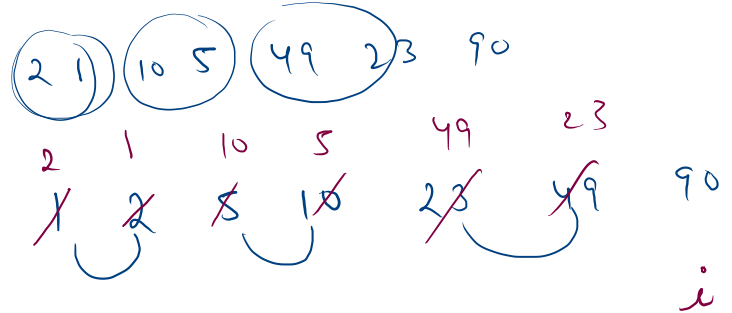
```

7
10 90 49 2 1 5 23

```

↓

2 1 10 5 49 23 90



0 < 6

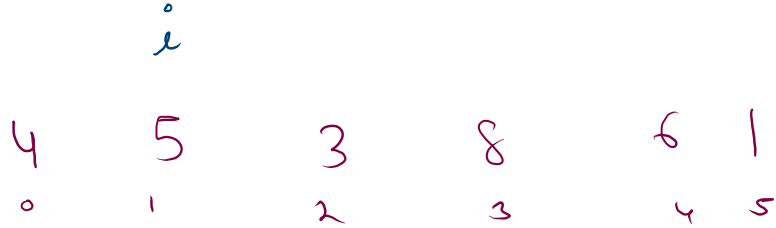
2 < 6

4 < 6

6 < 6

## Peak Elements

`**arr[i]**` is a peak element only if `**arr[i-1] < arr[i] > arr[i+1]**`.



$A[i] > A[i-1]$  &  $A[i] > A[i+1]$   
print

Sample Input 0

```
6
4 5 3 8 6 1
```

Sample Output 0

```
5 8
```

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int n = scn.nextInt();
9         int [] A = new int[n];
10
11         for(int i = 0; i < n; i++){
12             A[i] = scn.nextInt();
13         }
14
15         for(int i = 1; i < n-1; i++){
16             if(A[i] > A[i-1] && A[i] > A[i+1]){
17                 System.out.print(A[i] + " ");
18             }
19         }
20     }
21 }
```