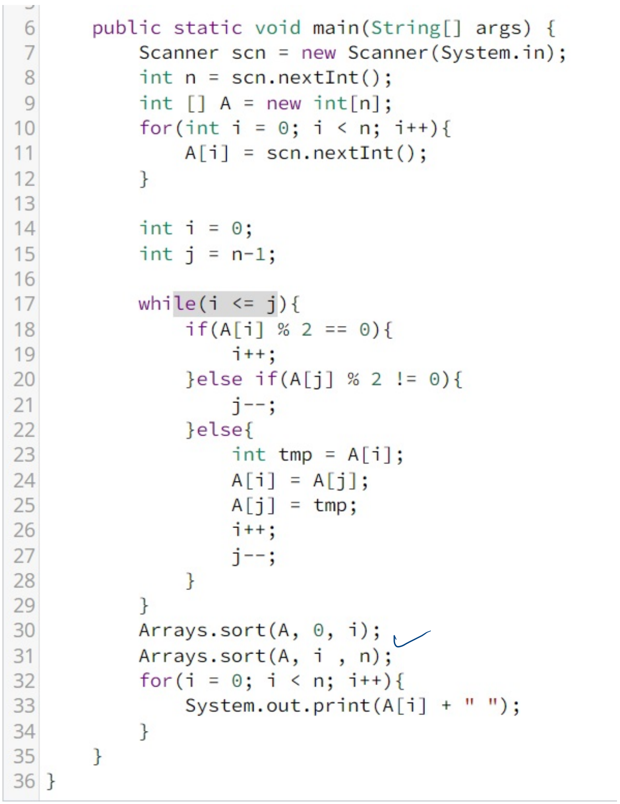
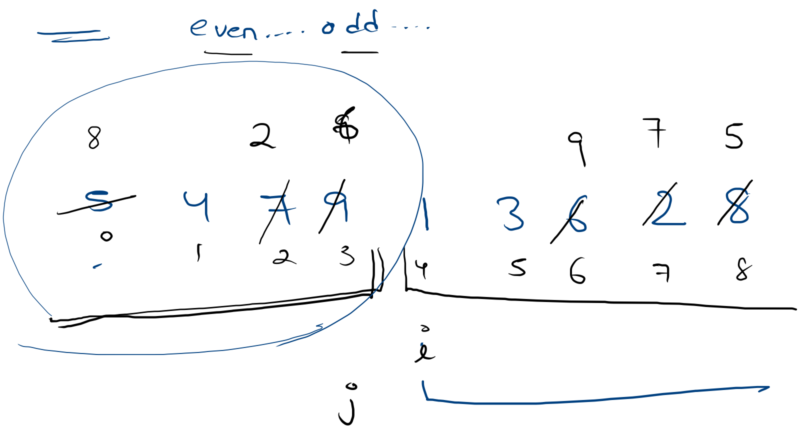


```java
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int [] A = new int[n];
        for(int i = 0; i < n; i++){
            A[i] = scn.nextInt();
        }

        int i = 0;
        int j = n-1;

        while(i <= j){
            if(A[i] % 2 == 0){
                i++;
            }else if(A[j] % 2 != 0){
                j--;
            }else{
                int tmp = A[i];
                A[i] = A[j];
                A[j] = tmp;
                i++;
                j--;
            }
        }
        Arrays.sort(A, 0, i);
        Arrays.sort(A, i , n);
        for(i = 0; i < n; i++){
            System.out.print(A[i] + " ");
        }
    }
}
```

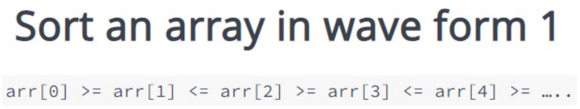


$0, i$

$[0, i)$

$[0, 4) \Rightarrow [0, 3]$

$[i, n) \Rightarrow [i, n-1]$

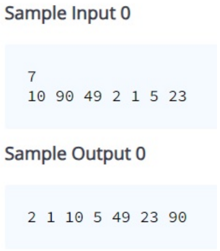


Sort an array in wave form 1

```
arr[0] >= arr[1] <= arr[2] >= arr[3] <= arr[4] >= …..
```

→ wave

2 ≥ 1 ≤ 10 ≥ 5 ≤ 49 ≥ 23 ≤ 90



Sample Input 0

```
7
10 90 49 2 1 5 23
```

Sample Output 0

```
2 1 10 5 49 23 90
```

n=7

10 90 49 2 1 5 23

(2 1 10 5 49 23 90)

1 2 5 10 23 49 90

logic.

1. Sort → ✓

2. alt. swap

0≤5
2≤5
4≤5

6≤5 ✗

n=7

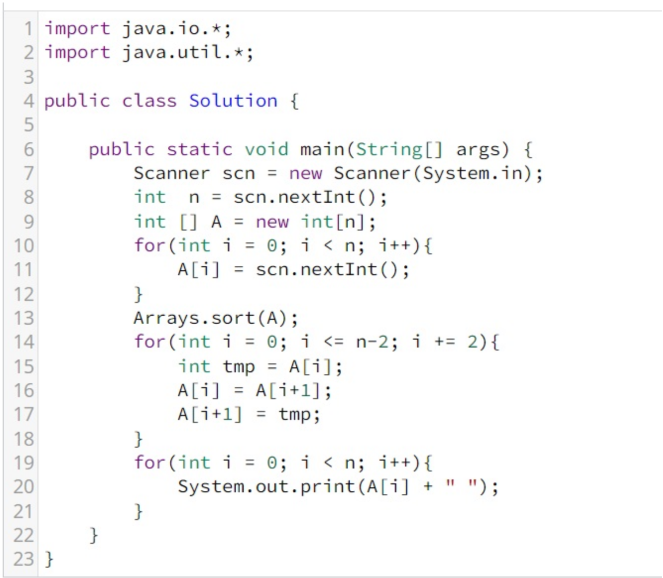
10 90 49 2 1 5 23

2 1 10 5 49 23

~~1~~ ~~2~~ ~~5~~ ~~10~~ ~~23~~ ~~49~~ 90

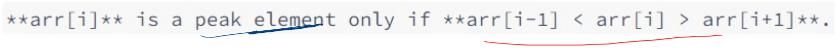
0 1 2 3 4 5 6

i i

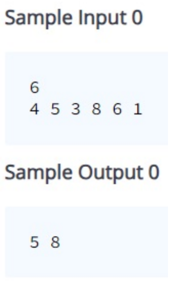


```java
import java.io.*;
import java.util.*;

public class Solution {

    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();
        }
        Arrays.sort(A);
        for(int i = 0; i <= n-2; i += 2){
            int tmp = A[i];
            A[i] = A[i+1];
            A[i+1] = tmp;
        }
        for(int i = 0; i < n; i++){
            System.out.print(A[i] + " ");
        }
    }
}
```

Peak Elements



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



Sample Input 0

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

Sample Output 0

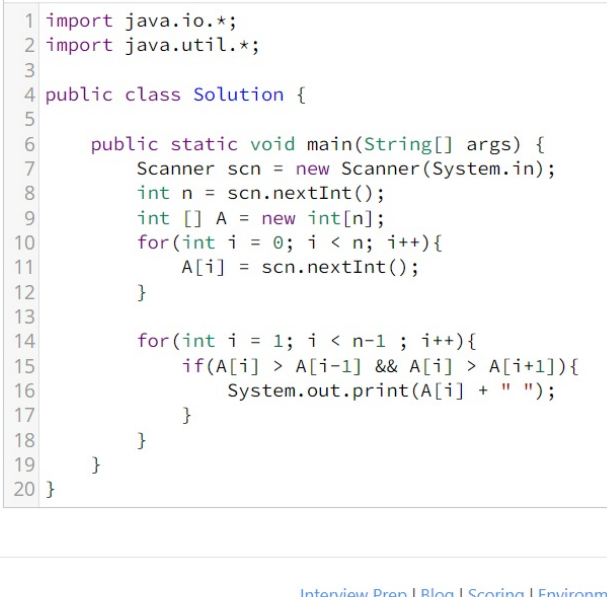
```
5 8
```

n=6

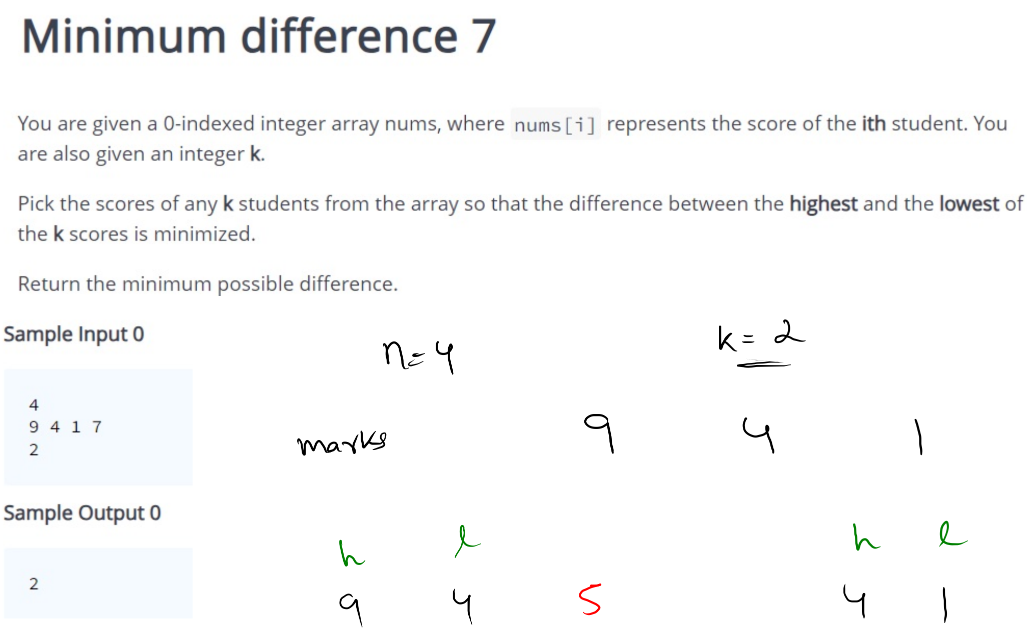
4	5	3	8	6	1
0	1	2	3	4	5
	i				

$if (A[i] > A[i-1] \;\&\&\; A[i] > A[i+1])$

↳ $A[i]$



```java
import java.io.*;
import java.util.*;

public class Solution {

    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();
        }

        for(int i = 1; i < n-1 ; i++){
            if(A[i] > A[i-1] && A[i] > A[i+1]){
                System.out.print(A[i] + " ");
            }
        }
    }
}
```



Minimum difference 7

You are given a 0-indexed integer array nums, where `nums[i]` represents the score of the **ith** student. You are also given an integer **k**.

Pick the scores of any **k** students from the array so that the difference between the **highest** and the **lowest** of the **k** scores is minimized.

Return the minimum possible difference.

Sample Input 0

```
4
9 4 1 7
2
```

Sample Output 0

```
2
```

n=4 k=2

marks 9 4 1 7

h 9	l 4	5	h 4	l 1	3
h 9	l 1	8	l 4	h 7	3
h 9	l 7	$\boxed{2}$	l 1	h 7	6

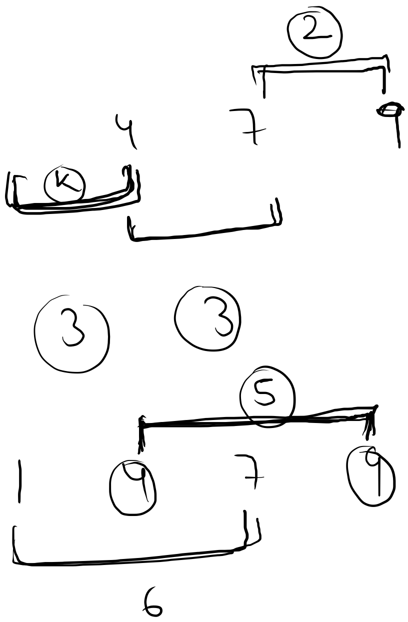
n=4

k=2

9 4 1 7

1. sort

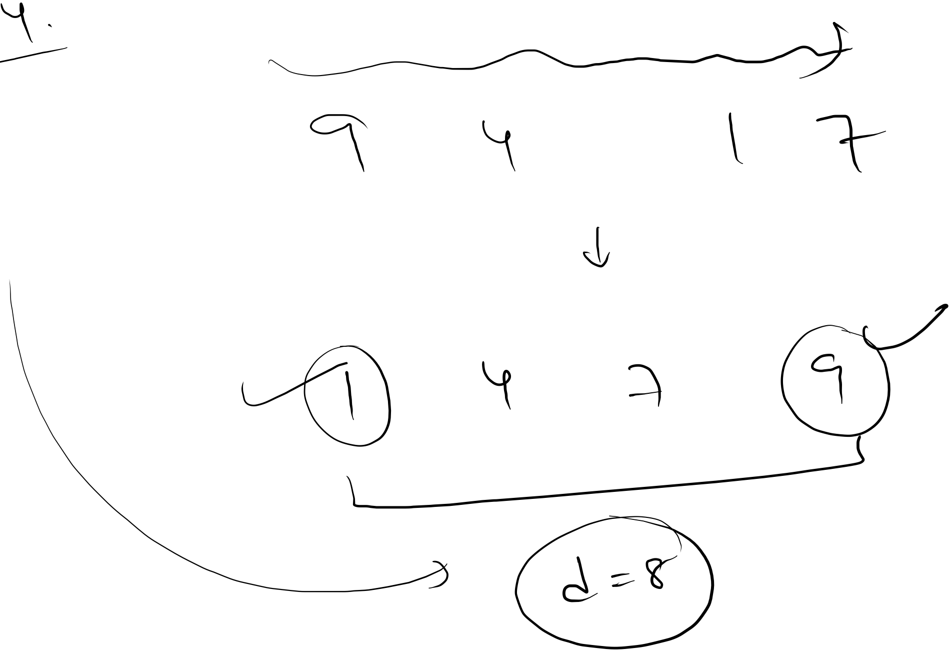
2. all contiguos window k



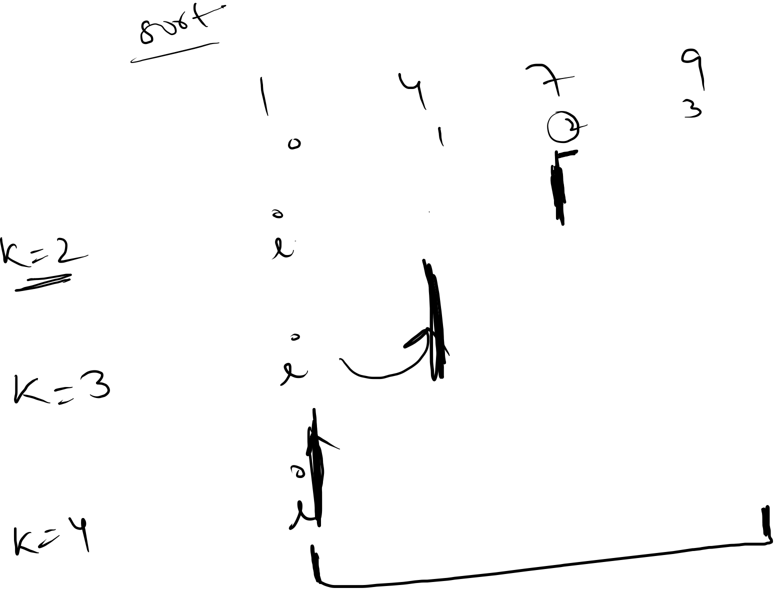
ans=2

ans=5

k=4.



n=4



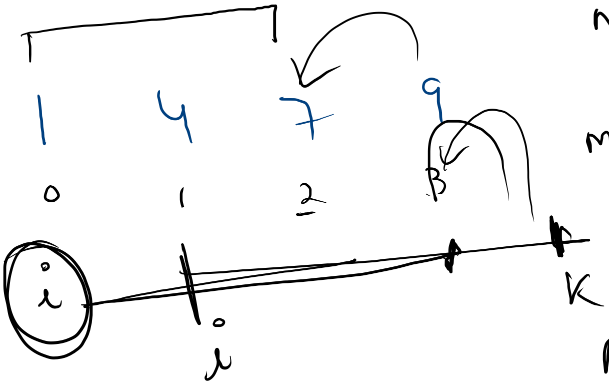
i ... < n-k

i ... <= n-k   4-3=1

i ... <= n-k   4-4=0

$k=3$

$n=4$



$\max = A[n-k]$

$\underline{\max = A[i+k-1]}$

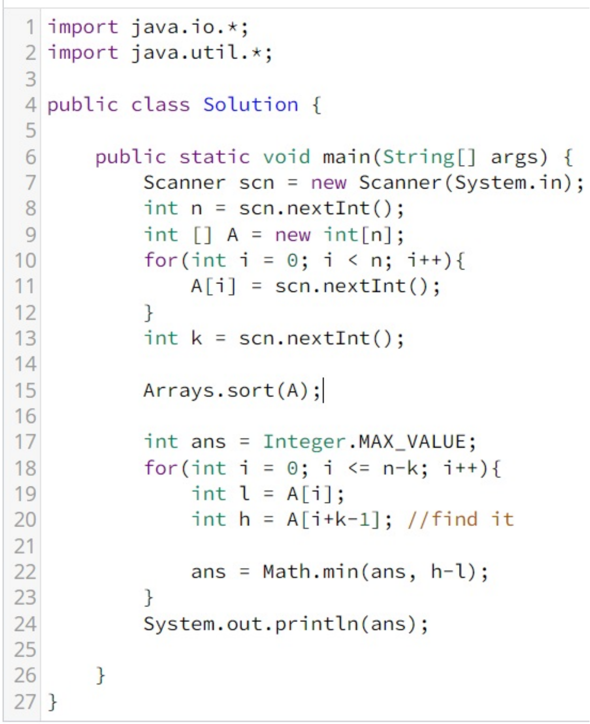
$A[0+3-1]$

$A[2]$

$A[1+3-1]$

$A[3]$

$\underline{i+k-1}$



```java
import java.io.*;
import java.util.*;

public class Solution {

    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();
        }
        int k = scn.nextInt();

        Arrays.sort(A);

        int ans = Integer.MAX_VALUE;
        for(int i = 0; i <= n-k; i++){
            int l = A[i];
            int h = A[i+k-1]; //find it

            ans = Math.min(ans, h-l);
        }
        System.out.println(ans);

    }
}
```

Insertion sort

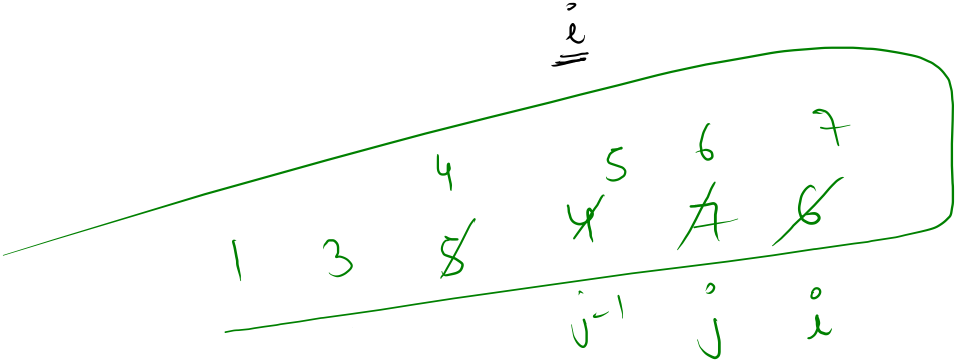
★ find correct position for current ele.

~~3~~ 1 ~~5~~ ~~1~~ 3 5

~~5~~ ~~3~~ ~~1~~ 4 7 6

~~if~~ ↳ $A[j] < A[j-1]$

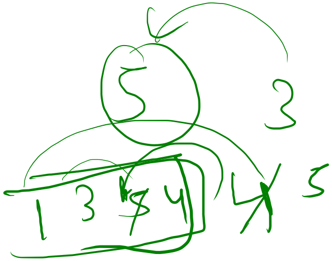
↳ swap.



else

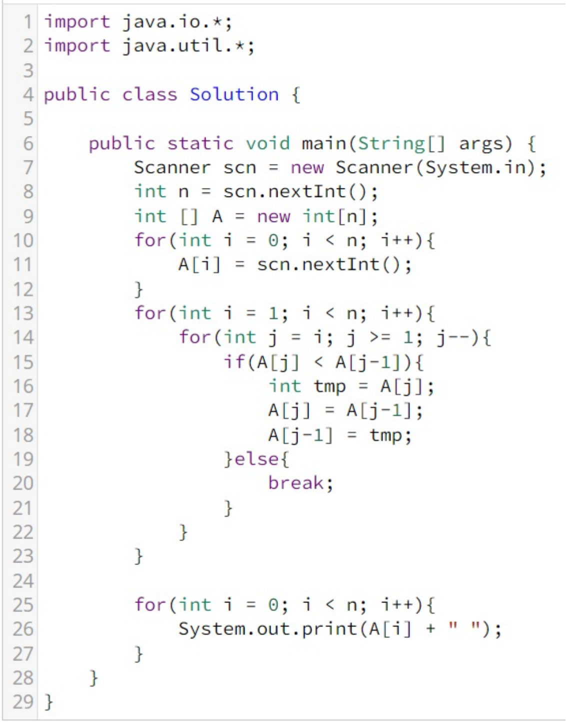
↳ Stop

break

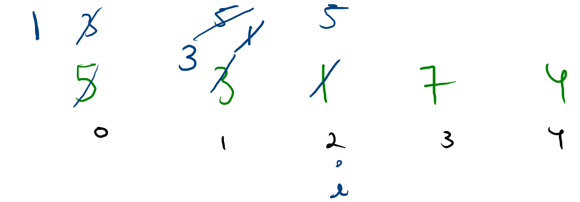


6 < 7

~~6 < 5~~



```
import java.io.*;
import java.util.*;

public class Solution {

    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();
        }
        for(int i = 1; i < n; i++){
            for(int j = i; j >= 1; j--){
                if(A[j] < A[j-1]){
                    int tmp = A[j];
                    A[j] = A[j-1];
                    A[j-1] = tmp;
                }else{
                    break;
                }
            }
        }

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



$i=1 \rightarrow 1$

$i=2 \rightarrow 2$

$i=3 \rightarrow 3$

$i=n-1 \rightarrow n-1$

$1+2+3+\dots+n-1$

$$\frac{n(n+1)}{2} = \frac{(n-1)(n)}{2}$$

$O(n^2)$