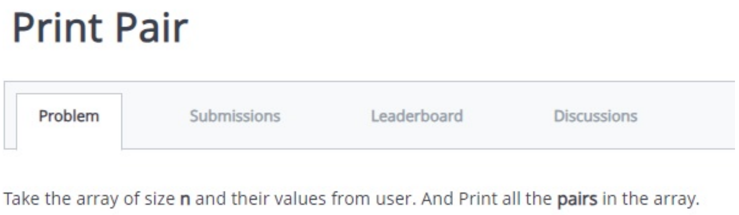
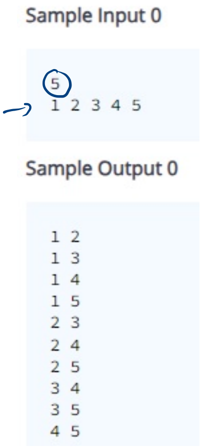


Print Pair

Take the array of size **n** and their values from user. And Print all the **pairs** in the array.

n=5

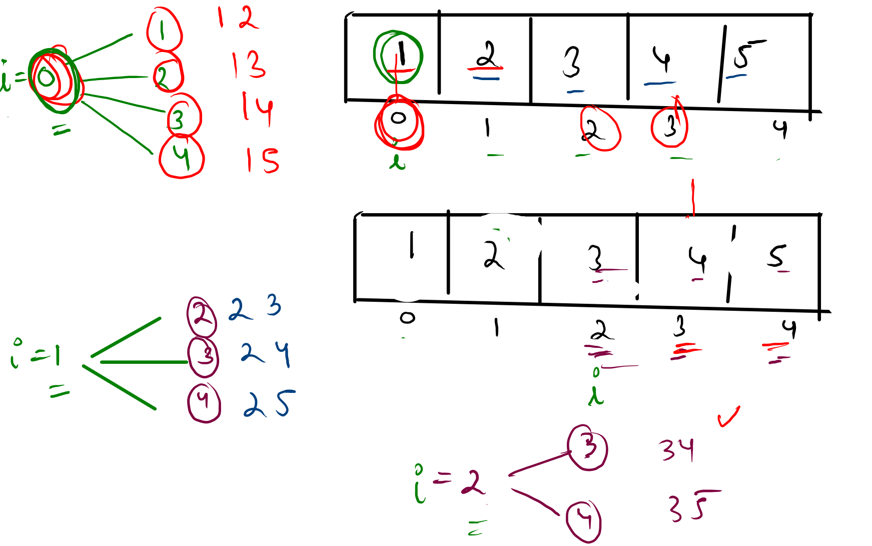


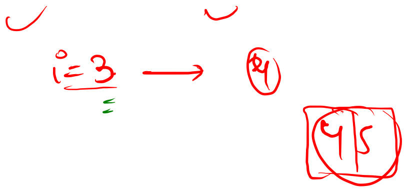
Sample Input 0

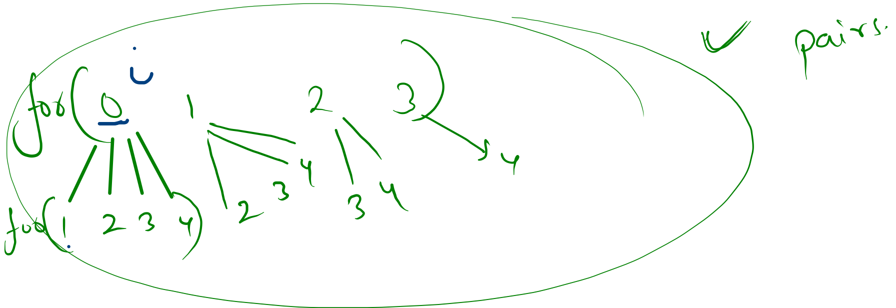
```
5
1 2 3 4 5
```

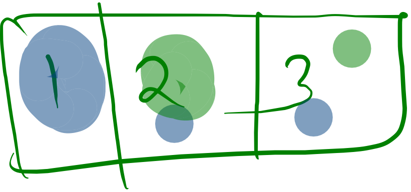
Sample Output 0

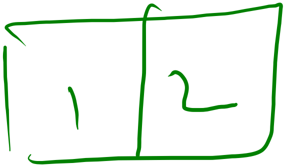
```
1 2
1 3
1 4
1 5
2 3
2 4
2 5
3 4
3 5
4 5
```

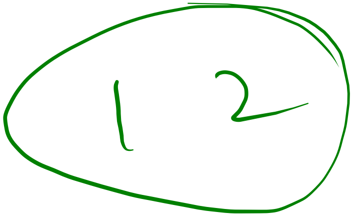








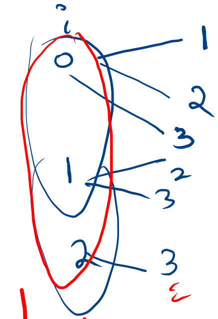


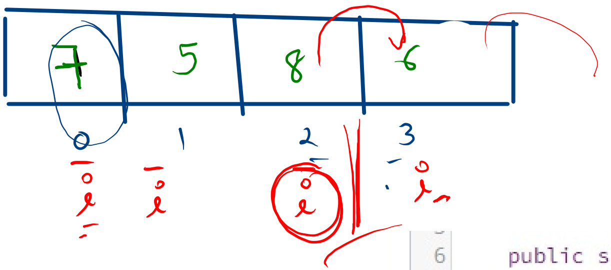


1 2

1 3

2 3





n=4

7 5

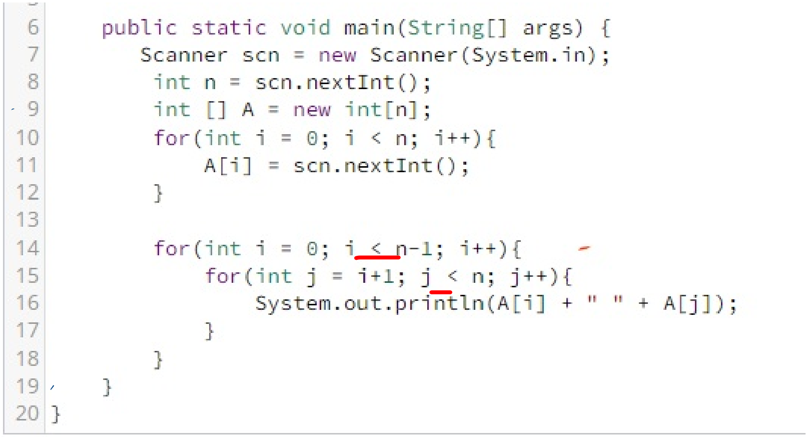
7 8

7 6

5 8

5 6

8 6

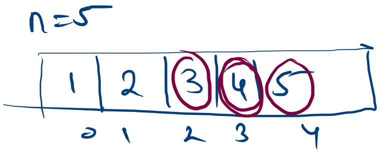


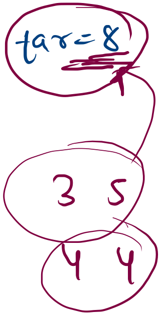
```
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int [] A = new int[n];
        for(int i = 0; i < n; i++){
            A[i] = scn.nextInt();
        }

        for(int i = 0; i < n-1; i++){
            for(int j = i+1; j < n; j++){
                System.out.println(A[i] + " " + A[j]);
            }
        }
    }
}
```

Find all Combination

Problem Submissions Leaderboard Discussions

Given condition is that the array contains all the unique elements. Then take the sum as an integer input and print all the combinations of the pairs that add up to the given sum.



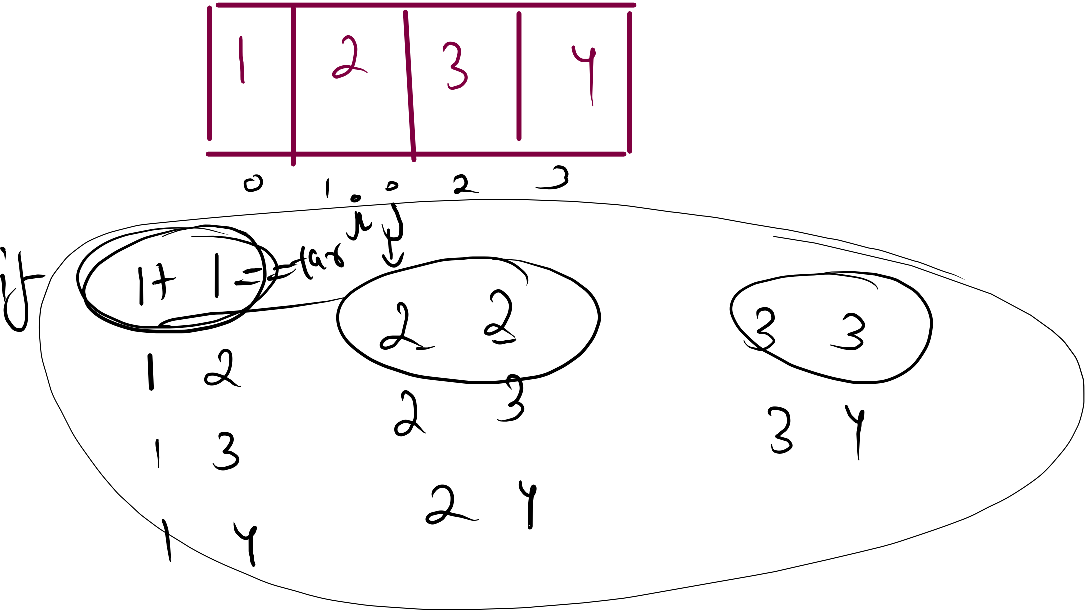


Sample Input 0

```
n 5
  1 2 3 4 5
  8 tar
```

Sample Output 0

```
3 5
4 4
```

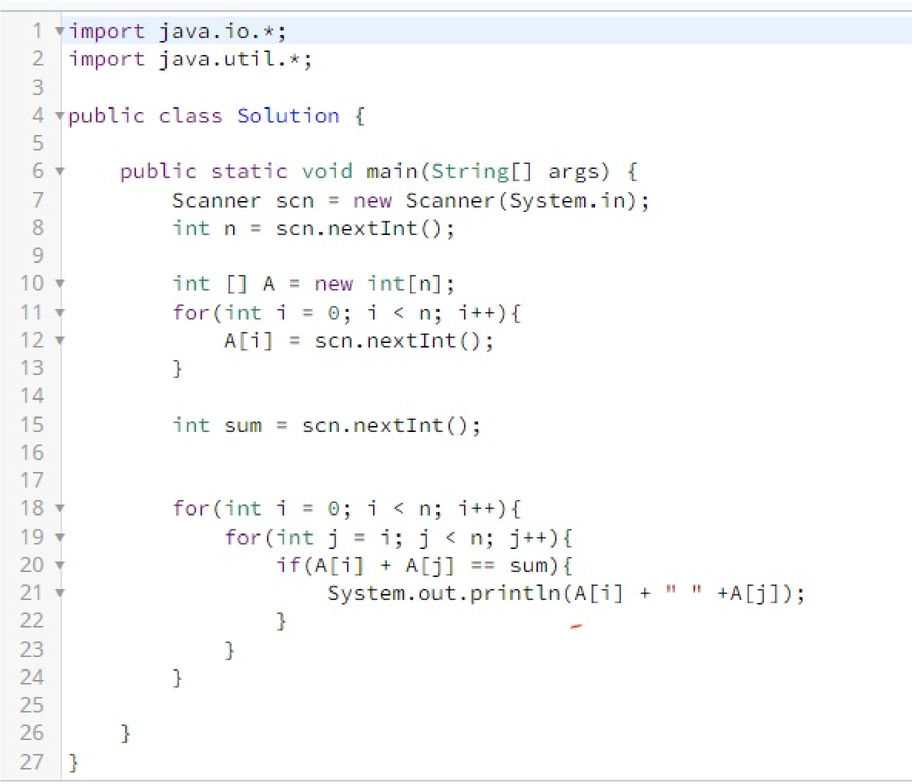


j = i

2 2

2 3

2 4



```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 sum = scn.nextInt();


        for(int i = 0; i < n; i++){
            for(int j = i; j < n; j++){
                if(A[i] + A[j] == sum){
                    System.out.println(A[i] + " " +A[j]);
                }
            }
        }

    }
}
```

1 2 3 4 5

HW.

1	2	3	4	5

O/P 1.

1 1
1 2
1 3
1 4
1 5

2 2
2 3
2 4
2 5

3 3
3 4
3 5

8
4 4
4 5

5 5

✓

✓ O/P 2.

1 1
1 2
1 3
1 4
1 5

2 1
2 2
2 3
2 4
2 5

3 1
3 2
3 3
3 4
3 5

4 1
4 2
4 3
4 4
4 5

5 1
5 2
5 3
5 4
5 5

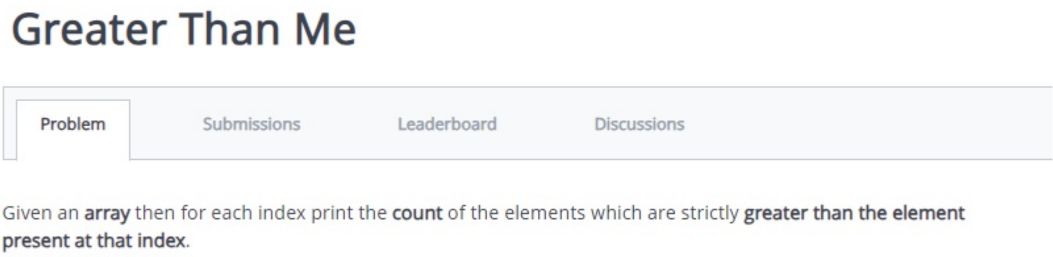
O/P 3

1 2
1 3
1 4
1 5

2 3
2 4
2 5

3 4
3 5

4 5



Greater Than Me

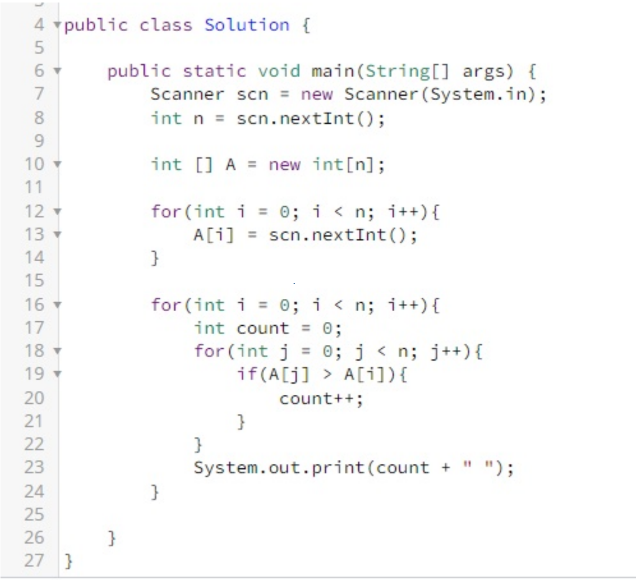
Given an **array** then for each index print the **count** of the elements which are strictly **greater than the element present at that index**.

n= 5

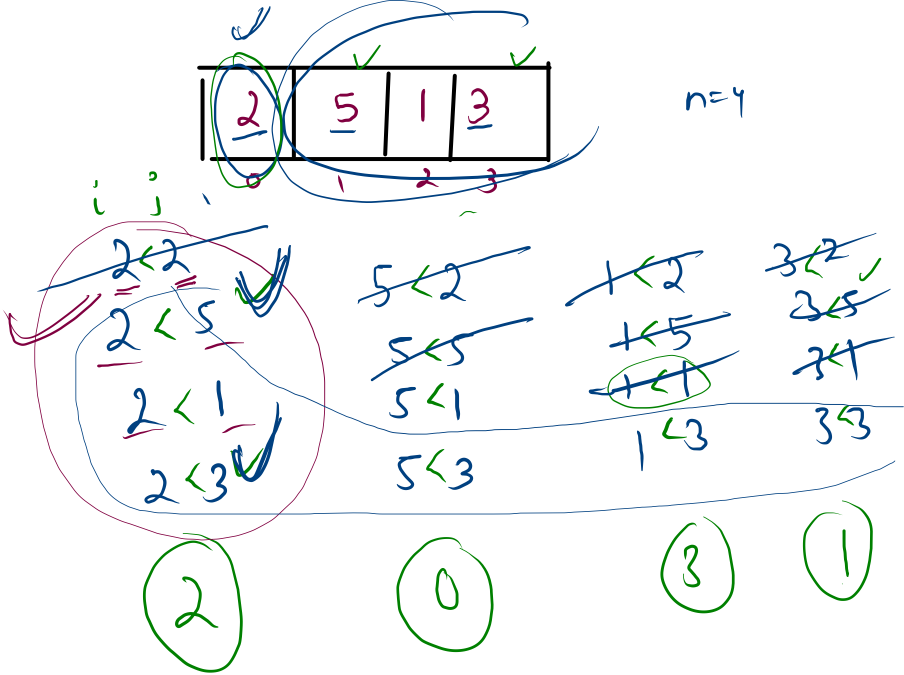
3	7	4	5	6
0	1	2	3	4

o/p→ 4 0 3 2 1

how many no. are there which is strictly greater than current element



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

    }
}
```



i=0
j ≠ i+1

i=0
j=0

i=0
j=i

Greater than me Right

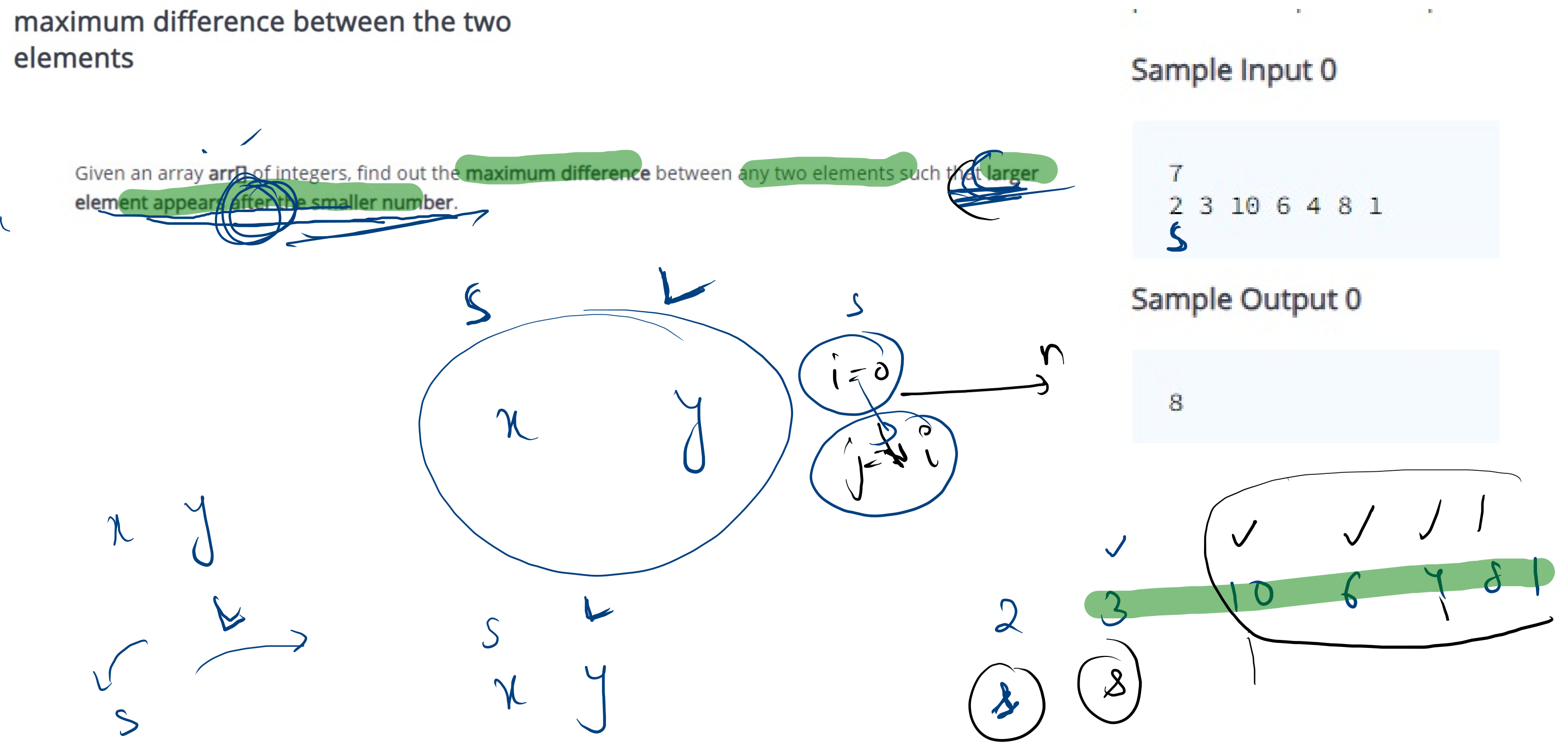
S L

L - S = max

6 5 7 1 3

ag. → 1 1 0 1 0

prev. 1 2 0 4 3



maximum difference between the two elements

Given an array **arr[]** of integers, find out the **maximum difference** between any two elements such that larger element appears after the smaller number.

Sample Input 0

```
7
2 3 10 6 4 8 1
```

Sample Output 0

```
8
```

4   2   3   5

8   ✓

ans = ~~1~~ 3

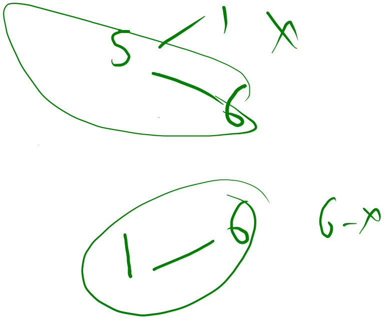
2 4 5 1 6

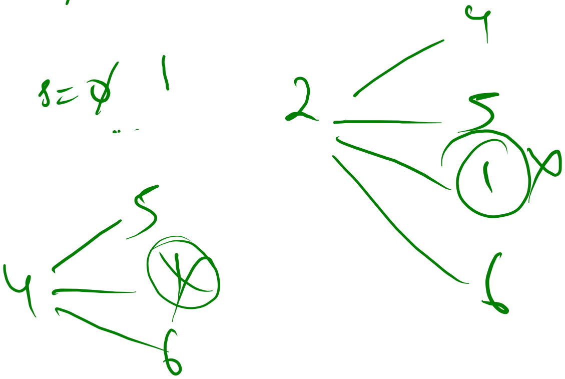
0 1 2 3 4

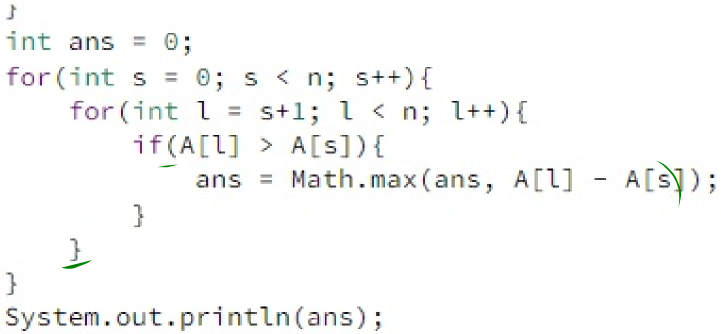
s l

ans = ~~0~~ ~~2~~ ~~3~~ 4

s = ~~0~~ 1

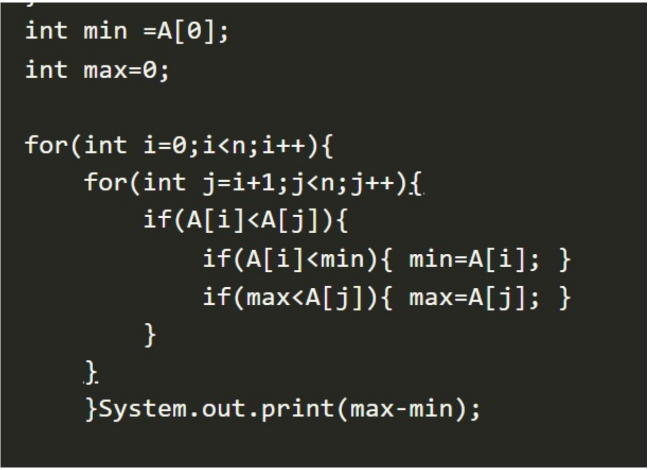






```
}
int ans = 0;
for(int s = 0; s < n; s++){
    for(int l = s+1; l < n; l++){
        if(A[l] > A[s]){
            ans = Math.max(ans, A[l] - A[s]);
        }
    }
}
System.out.println(ans);
```

2 4 5 1 6



```
int min =A[0];
int max=0;

for(int i=0;i<n;i++){
    for(int j=i+1;j<n;j++){
        if(A[i]<A[j]){
            if(A[i]<min){ min=A[i]; }
            if(max<A[j]){ max=A[j]; }
        }
    }
    }System.out.print(max-min);
```