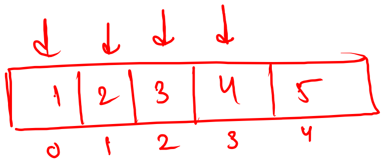
Ques Print Pairs

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

    solve(arr, n);
}

public static void solve(int[] arr, int n) {
    for (int i = 0; i < n; i++) {
        for (int j = i + 1; j < n; j++) {
            System.out.println(arr[i] + " " + arr[j]);
        }
    }
}
```



$i=0,\ 1 \rightarrow 2, 3, 4, 5$

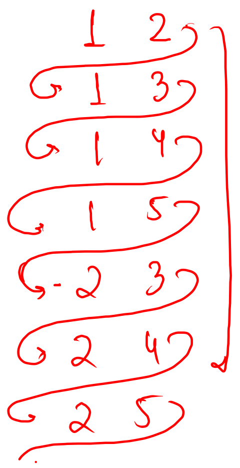
$i=1,\ 2 \rightarrow 3, 4, 5$

$i=2,\ 3 \rightarrow 4, 5$

$i=3,\ 4 \rightarrow 5$

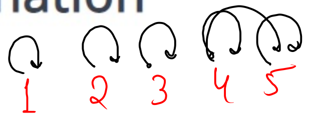
$S.C = O(1)$

$T.C = O(n*n)$



Find all Combination

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



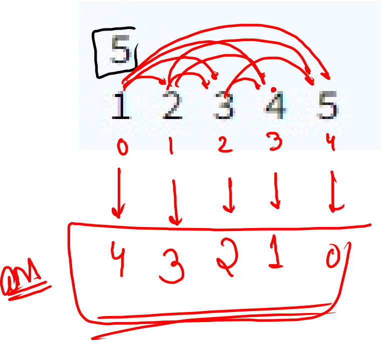
1 1	22	33	$\boxed{44}$	55
1 2	23	34	45	
1 3	24	$\boxed{35}$		
1 4	25			
1 5				

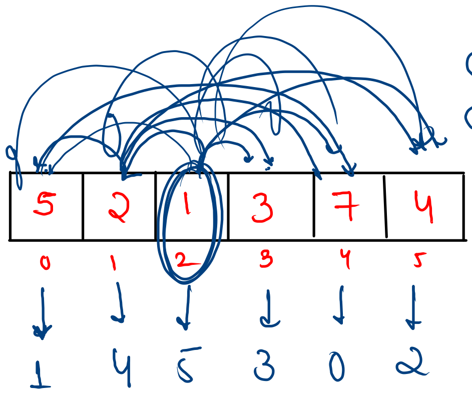
```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    int target = scn.nextInt();

    solve(arr, n, target);
}

public static void solve(int[] arr, int n, int target) {
    for (int i = 0; i < n; i++) {
        for (int j = i; j < n; j++) {
            int num1 = arr[i];
            int num2 = arr[j];
            if (num1 + num2 == target) {
                System.out.println(num1 + " " + num2);
            }
        }
    }
}
```

Greater Than Me





count= ~~0~~ ~~1~~ ~~2~~ ~~3~~ 4

count= ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

count= ~~0~~ ~~1~~ ~~2~~ 3

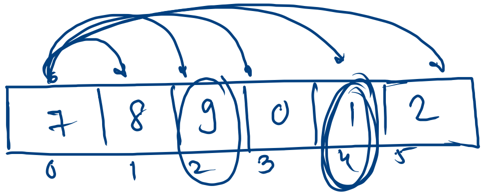
count= 0

count = ~~0~~ ~~1~~ 2

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

    solve(arr, n);
}

public static void solve(int[] arr, int n) {
    for (int i = 0; i < n; i++) {
        int count = 0;
        for (int j = 0; j < n; j++) {
            int num1 = arr[i];
            int num2 = arr[j];
            if (num1 < num2) {
                count++;
            }
        }
        System.out.print(count + " ");
    }
}
```



c= ~~0~~ ~~1~~ ~~2~~ 3

c= ~~0~~ ~~1~~ 2

c= ~~0~~ 1

c= ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

c= ~~0~~ ~~1~~ ~~2~~ 3 4

i=0,(7) 7>7, 8>7, 9>7

i=1,(8) 7>8, 8>8, 9>8, 0>8, 1>8, 2>8

i=2,(9) 7>9, 8>9, 9>9, 0>9 - - - -

i=3,(0) 7>0, 8>0, 9>0, 0>0, 1>0, 2>0

i=4 (1), 7>1, 8>1, 9>1, 0>1, 1>1, 2>1

i=5(2), 7>2, 8>2, 9>2, 0>2, 1>2, 2>2

$\boxed{2, 1, 0, 5, 4, 3}$

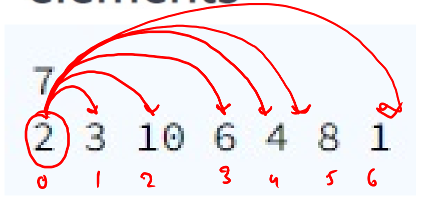
Greater At Right

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

    solve(arr, n);
}

public static void solve(int[] arr, int n) {
    for (int i = 0; i < n; i++) {
        int count = 0;
        for (int j = i; j < n; j++) {
            int num1 = arr[i];
            int num2 = arr[j];
            if (num1 < num2) {
                count++;
            }
        }
        System.out.print(count + " ");
    }
}
```

maximum difference between the two elements

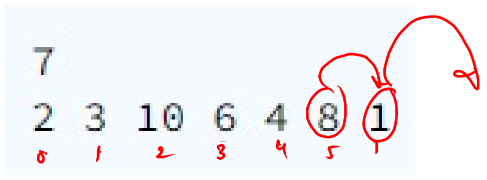


diff = ~~1~~ 8

Conditions
↳ larger element should be at right
↳ smaller element should be at left
↳ diff. of both no. should be maximum

```
public static void solve(int[] arr, int n) {
    int ans = 0;
    for (int i = 0; i < n; i++) {
        for (int j = i; j < n; j++) {
            int num1 = arr[i];
            int num2 = arr[j];
            if ( num2 > num1 ) {
                int diff = num2 - num1;
                if ( diff > ans ) {
                    ans = diff;
                }
            }
        }
    }
    System.out.println(ans);
}
```

ans = ~~0~~ 8



i=1 (3), diff = ~~7~~ ~~3~~ ~~1~~ 5

i=2 (10), ______

i=3 (6), diff = 2

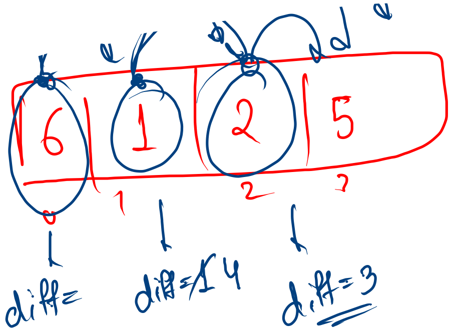
i=4 (4), diff = 4

i=5 (8), ______

i=6 (1) ______ ✗

```
public static void solve(int[] arr, int n) {
    int ans = 0;
    for (int i = 0; i < n; i++) {
        for (int j = i; j < n; j++) {
            int num1 = arr[i];
            int num2 = arr[j];
            if ( num2 > num1 ) {
                int diff = num2 - num1;
                if ( diff > ans ) {
                    ans = diff;
                }
            }
        }
    }
    System.out.println(ans);
}
```

ans = ~~0~~ ~~1~~ 4



1>6, 2>6, 5>6

2>1, 5>1

5>2

Find Duplicate 3

arr 1, 2, 3, 4, 5, 1

Arrays.sort(arr);

⇒ 1, 1, 2, 3, 4, 5

i

```
for ( ______ )
        return true
return false
```

$$\begin{tikzcd} \underline{\underline{O(n)}} \arrow[r, bend left] \arrow[r, phantom, "<"] & \underline{\underline{O(n^2)}} \end{tikzcd}$$