Add One

Take an **array arr** of size N as input which represents a **large number**. **Add 1 (one)** to this large number and print the resultant array.

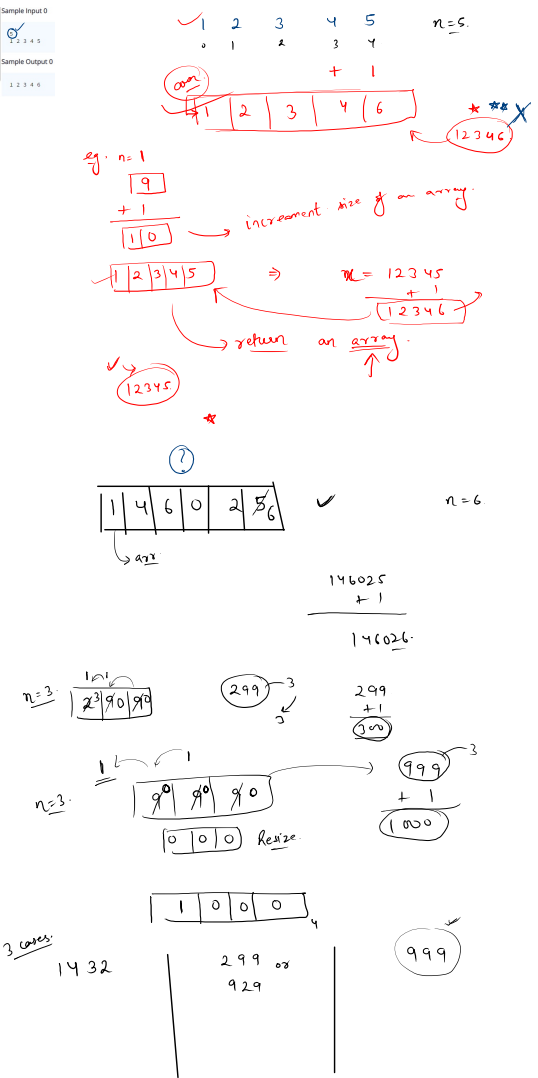
eg:- [4,2,3,6,5,8,7,1,5,3,9,6] In this case answer must be [4,2,3,6,5,8,7,1,5,3,9,7]

Sample Input 0

```
1 2 3 4 5
```

Sample Output 0

```
1 2 3 4 6
```



Add one.

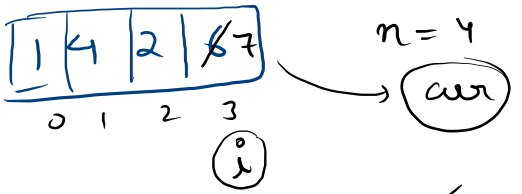
```java
import java.util.*;

public class Solution {
    public static void printArr(int [] arr){
        for(int e : arr){
            System.out.print(e+" ");
        }
    }

    public static int [] addOne(int [] arr){
        for(int i  = arr.length-1; i >= 0; i--){
            if(arr[i] != 9 ){
                arr[i]++;
                break;
            }
            else{
                arr[i] = 0;
            }
        }
        if(arr[0] == 0){
            //resize
            int [] ans = new int[arr.length+1];
            ans[0] = 1;
            return ans;
        }
        return arr;
    }
}
```

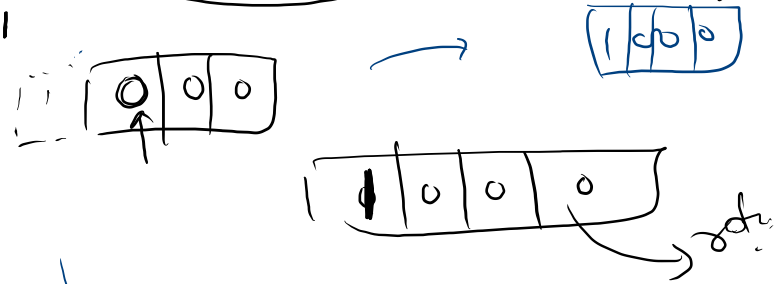
1427

1.

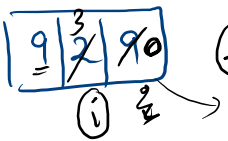


arr[3] != 9 ✓

3.



2



(2) times →

9 3 0 ✓

V. imp.

1. 1 4 2 6
2. 9 2 9
3. 9 9 9

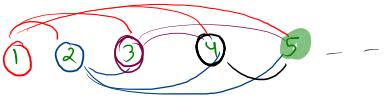
Print Pair

nested loops

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

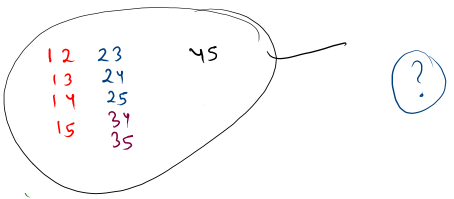
Sample Input 0

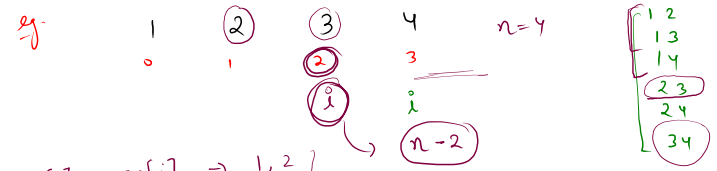
```
5
1 2 3 4 5
```



Sample Output 0

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





arr[i] , arr[j] =>

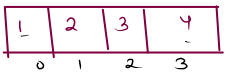
1, 2
1, 3
1, 4
2, 3
2, 4
3, 4

```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 [] arr = new int[n];
        for(int i = 0; i<n; i++){
            arr[i] = scn.nextInt();
        }

        for(int i = 0; i<=n-2; i++){
            for(int j = i+1; j<n; j++){
                System.out.println(arr[i]+" "+arr[j]);
            }
        }
    }
}
```



n=4

i = 0 ≤ 2

i = 0 1 2 3

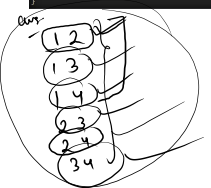
i

j

3 ≤ 2

1 ≤ 2

3 < 4



Find The Pair 4

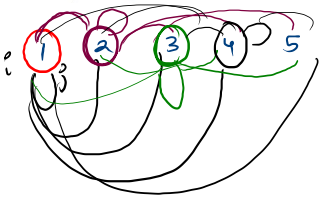
Given condition is that the array contains all the unique elements. Then take the sum as an integer input and print all the permutations of the pairs that add up to the given sum k.

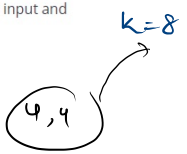
Sample Input 0

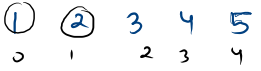
```
5
1 2 3 4 5
8
```

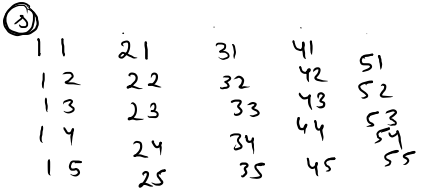
Sample Output 0

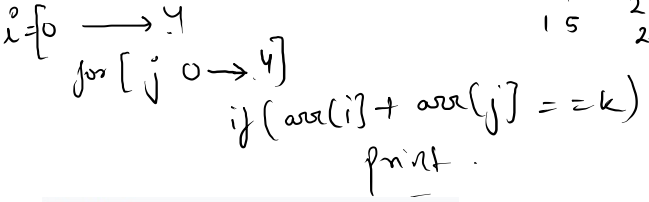
```
3 5
4 4
5 3
```









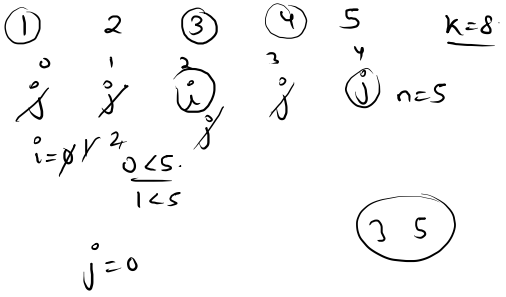



```java
import java.io.*;
import java.util.*;

public class Solution {

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

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



Greater Than Me

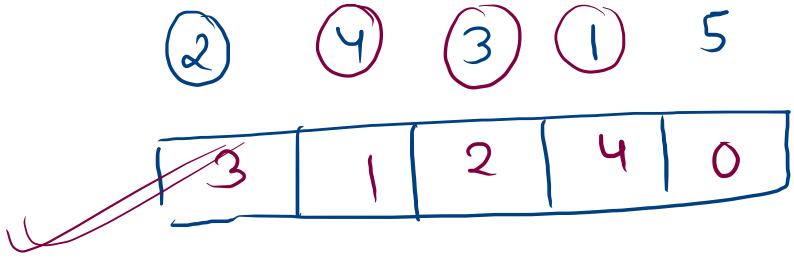
Given an array then for each index print the count of the elements which are strictly greater than the element present at that index.

Sample Input 0

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

Sample Output 0

```
4 3 2 1 0
```



1. Use extra arr.