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**.

Input Format

First line contains an integer number **n** representing size of array.

Second line contains **n** integer numbers representing elements of the array.

Third line contains an integer number **k**

Constraints

```
1 <= n <= 100000

0 <= array[index] <= 100000

0 <= k <= 1000000
```

Output Format

Print all pair that gives the **sum** equals to the given number **k**.

Sample Input 0

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

Sample Output 0

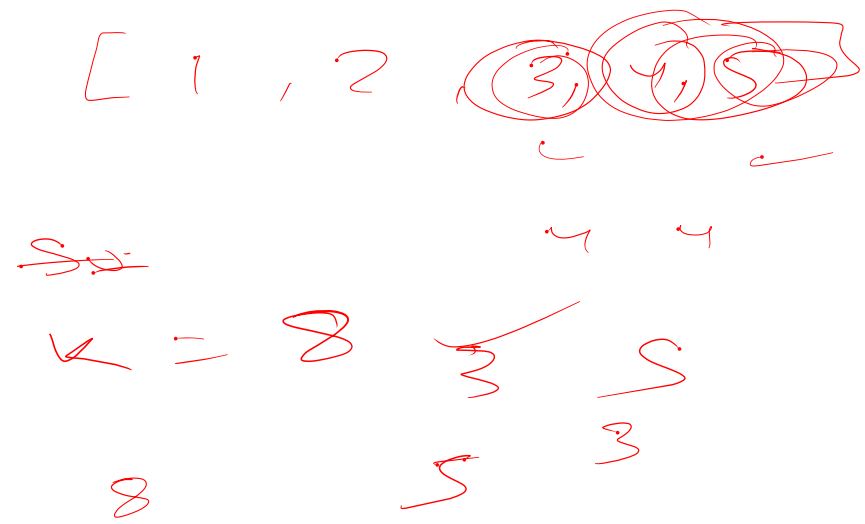
```
3 5
4 4
5 3
```

Explanation 0

Since arr[2]+arr[4] = 8

arr[3]+arr[3] = 8

arr[4]+arr[2] = 8



Submitted Code

Language: Java 15

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

public class Solution {

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

        int k=sc.nextInt();
        findPair(arr,k);
    }
    public static void findPair(int arr[] , int k){
        for(int i=0;i<arr.length;i++){
            for(int j=0;j<arr.length;j++){
                if(arr[i]+arr[j]==k)System.out.println(arr[i]+" "+arr[j]);
            }
        }
    }
}
```

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

Input Format

First line contains an integer number **n** representing size of array.

Second line contains **n** numbers representing elements of the array and all element will be **unique**.

Constraints

```
1 <= n <= 100000

0 <= array[index] <= 100000
```

Output Format

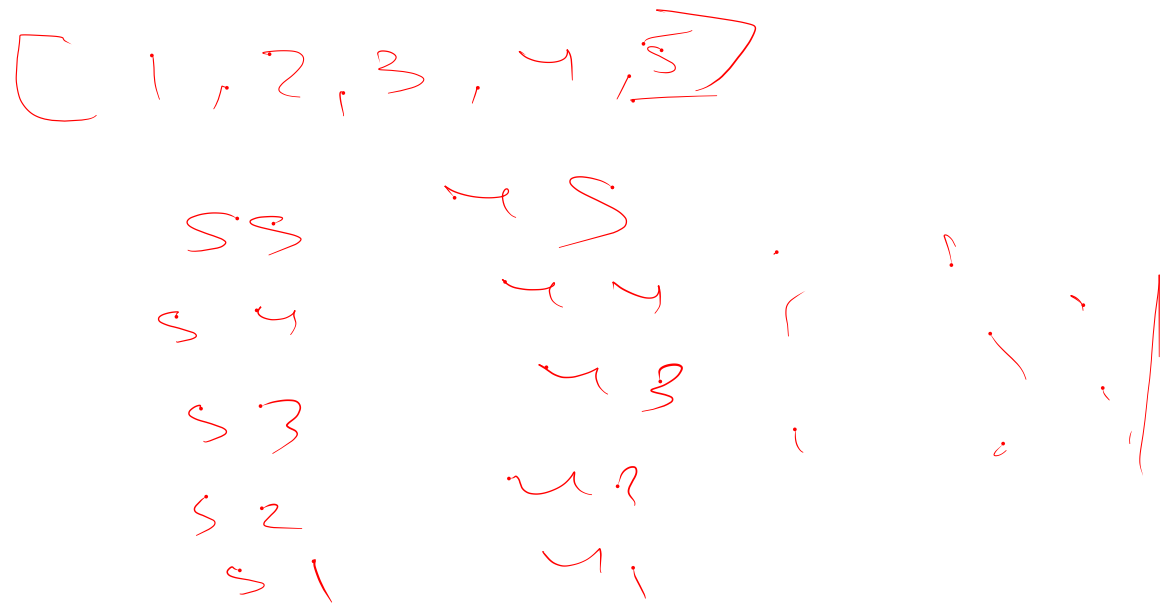
Print all pairs in differnt line.

Sample Input 0

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

Sample Output 0

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



Submitted Code

Language: Java 15

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

public class Solution {

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

        reversePair(arr);
    }
    public static void reversePair(int arr[] ){
        for(int i=arr.length-1;i>=0;i--){
            for(int j=arr.length-1;j>=0;j--){
                System.out.println(arr[i]+" "+arr[j]);
            }
        }
    }
}
```

Take the array of size **n** and their values from user. And Find **Pairs** whose **sum is odd.**

Input Format

First line contains an integer number **n** representing size of array.

Second line contains **n** numbers representing elements of the array and all element will be **unique**.

Constraints

```
1 <= n <= 100000

0 <= array[index] <= 100000
```

Output Format

Print the required pairs in different lines.

Sample Input 0

```
3
1 2 3
```

Sample Output 0

```
1 2
2 1
2 3
3 2
```

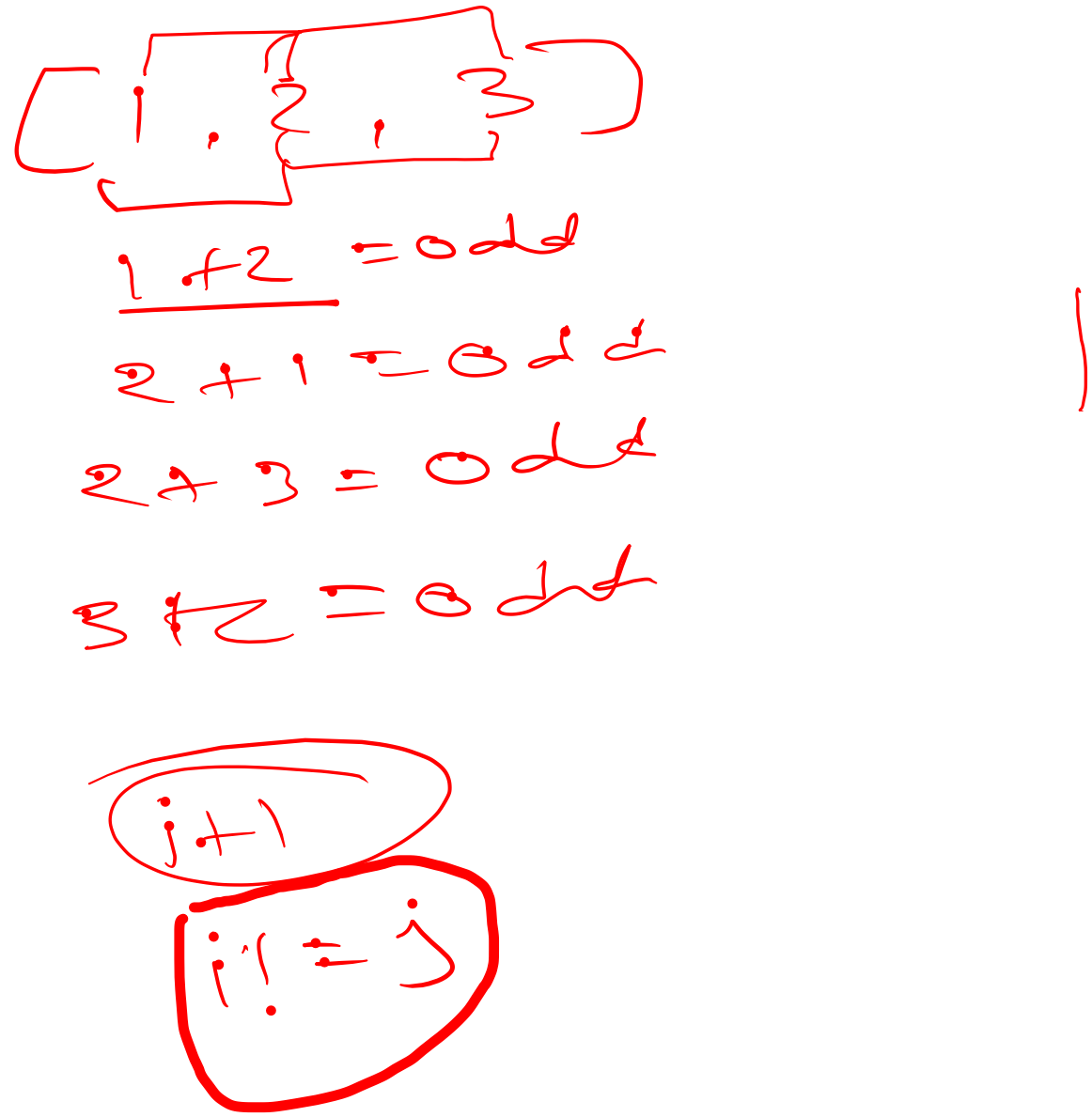
Explanation 0

arr[0]+arr[1] = 1+2 = odd

arr[1]+arr[0] = 2+1 = odd

arr[1]+arr[3] = 2+3 = odd

arr[0]+arr[1] = 1+2 = odd

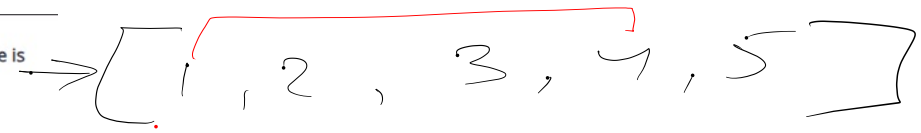


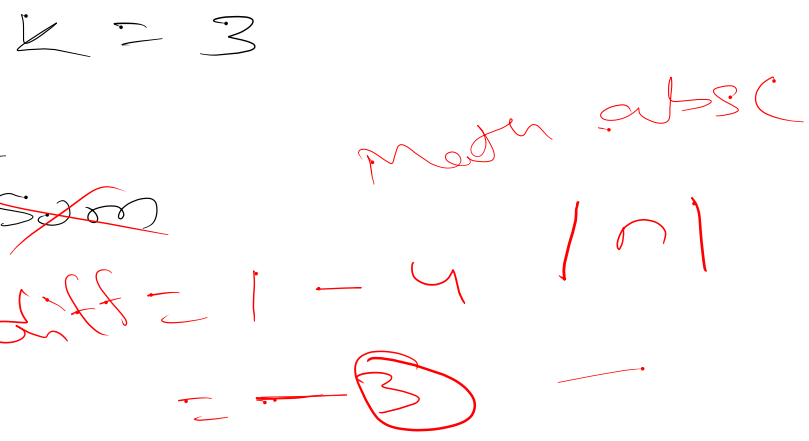
Submitted Code

Language: Java 15

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

public class Solution {

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


        oddPair(arr);
    }
    public static void oddPair(int arr[] ){
        for(int i=0;i<arr.length;i++){
            for(int j=0;j<arr.length;j++){
                if(i!=j && (arr[i]+arr[j])%2!=0)System.out.println(arr[i]+" "+arr[j]);
            }
        }
    }
}
```

Take the **array** and **k** as an integer input. 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 whose **absolute difference is k**.





Input Format

First line contains an integer number **n** representing size of array.

Second line contains **n** integer numbers representing elements of the array.

Third line contains an integer number **k**

Constraints

```
1 <= n <= 100000

0 <= array[index] <= 100000

0 <= k <= 1000000
```

Output Format

Print the required permutation in different line.

Sample Input 0

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

Sample Output 0

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

Explanation 0

|arr[0]-arr[3]| = |1-4| = 3

|arr[1]-arr[4]| = |2-5| = 3

|arr[3]-arr[1]| = |4-1| = 3

|arr[4]-arr[1]| = |5-2| = 3

Submitted Code

Language: Java 15

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

public class Solution {

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

        int k=sc.nextInt();
        findDiffer(arr,k);
    }
    public static void findDiffer(int arr[] , int k){
        for(int i=0;i<arr.length;i++){
            for(int j=0;j<arr.length;j++){
                if(Math.abs(arr[i]-arr[j])==k)System.out.println(arr[i]+" "+arr[j]);
            }
        }
    }
}
```