

## Reach Target

find all the required pair

### Problem Statement

Take the target as an integer input. Then print the **indices** of the two numbers such that they add to the target. Note that the array is sorted here. Use Two pointer, answer must be unique.

eg. Sorted.

1	3
0	1

o/p →

$$1 + 7 = 8$$

0	4
---	---

$$3 + 5 = 8$$

1	2
---	---

✓  
5  
2

6  
3

7  
4

tar = 8

?



1	3	5	6	7
0	1	2	3	4
	j	i		

tar = 8

$i < j$

sum

$A[i] + A[j] == \text{tar}$

print  
 $i++ j--$

$\text{sum} > \text{tar}$

$j--$

1	3	5	6	7
0	1	2	3	4
	(1)	j		

(8)

$1 + 7$

$\text{sum} == \text{tar}$

print  $i++ j--$

(sum) tar

$3 + 6 = 9 \downarrow \uparrow ?$   
(7)  $\uparrow$

0	4
1	2



## Target Sum

find the pair of target sum in array

### Problem Statement

The given array is not sorted. The given array may or may not contain duplicate elements. Then take the **target** as an integer input. Return Pair of **target sum** in which all pairs are **unique**, for example : [6, 7], [7, 6] are considered as the same pair.

Also if the array has repeated elements then return only unique pairs, for eg : if array is arr = [3, 3, 5, 5], and the target = 8 then result will have only one pair, i.e. [3, 5].

**Note** : Print the pairs such the smallest integers comes first.

1. sort

### Test Case 1

Input:

4  
3 3 5 5  
8

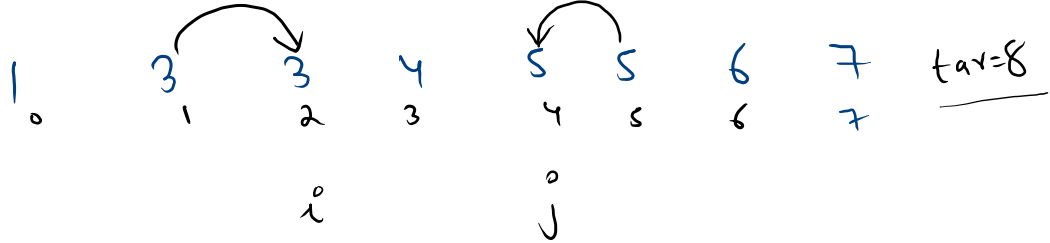
Output:

3 5

Explanation:

3 + 5 = 8

s == tar  
print ek  
it + j--



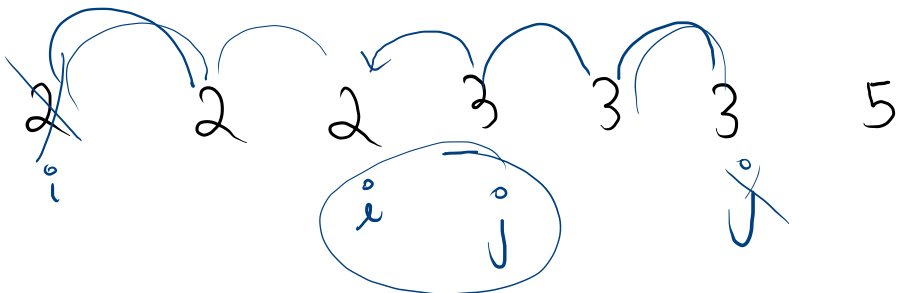
1 7

3 5

X (3 5)

$$\text{tar} = 5$$

1



✓  $s = \text{tar}$

$1 + s = 5$  ?

$s > \text{tar}$

$1 + 3 =$   
 $s < \text{tar}$

$1$        $2$        $2$   $2$        $3$   $3$   $3$        $5$   
 $0$        $1$        $2$   $3$        $4$        $5$   $6$        $7$   
 $i$                                      $j$

tar = 9

tar = 5.

1      (2)      2      2      (3)      3      3      5

2      3

tar = 4

2<sub>0</sub>      (2)<sub>1</sub>      2<sub>2</sub>      2<sub>3</sub>      (2)<sub>4</sub>

2      2

tar = 4

2<sub>0</sub>

No answer

tar = 5

1

2

2

2

j

3

i

3

3

5

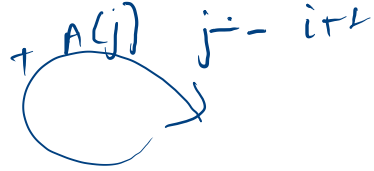
1.

s == t

A[i] == A[i+1] i++

A[j] == A[j-1] j--

else System.out.println(A[i])



i < j

2.

s > t

j--

3 2<sup>x</sup>

3.

s < t

i++



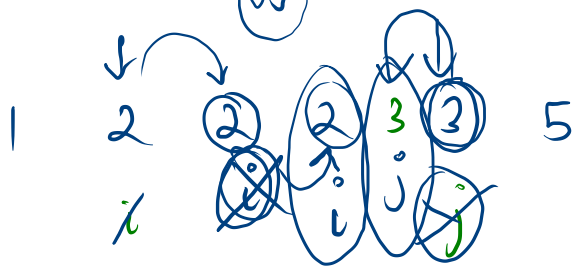
```

17 int j = n-1;
18
19 while(i < j){
20     int sum = A[i] + A[j];
21     if(sum == tar){
22         // found
23         if(A[i] == A[i+1]){
24             i++;
25         }
26         else if(A[j] == A[j-1]){
27             j--;
28         }
29         else{
30             System.out.println(A[i] + " " + A[j]);
31             i++;
32             j--;
33         }
34     }
35     else if(sum > tar){
36         j--;
37     }
38 }

```

5

1 2 2 2 3 3 5  
i j



$$1 + 5 =$$

$$1 + 3 < \text{tar}$$

2 3

$$2 + 3 = 5$$

## Count boat

Count boat

### Problem Statement

You are given an array people where people[i] is the weight of the ith person, and an infinite number of boats where each boat can carry a maximum weight of limit. Each boat carries at most two people at the same time, provided the sum of the weight of those people is at most limit.

Return the minimum number of boats to carry every given person.

Input Format

$$wt \leq limit$$

$$limit = 3$$

$$ans = 5$$

eg.

people

1 1 1 2 2 3 3  
0 1 2 3 4 5 6

$$pot = \frac{A}{6}$$

at max  
1 → 2 people  
boat

sort

$$A[i] + A[j] > limit \rightarrow count++ \quad j--$$

$$A[i] + A[j] \leq limit$$

3

i++  
j--  
count++

count++

1 1 1 2 2 3 3

fit 2 people  $\rightarrow$   $\left. \begin{array}{l} i++ \\ j-- \end{array} \right\}$  count++

fit heavy ele  $\rightarrow j--$

```

1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         Scanner scn = new Scanner(System.in);
5         int n = scn.nextInt();
6         int [] A = new int[n];
7         for(int i = 0; i < n; i++){
8             A[i] = scn.nextInt();
9         }
10        int limit = scn.nextInt();
11        Arrays.sort(A);
12        int i = 0;
13        int j = n-1;
14        int count = 0;
15        while(i <= j){
16            if(A[i] + A[j] <= limit){
17                i++;
18                j--;
19            }
20            else{
21                j--;
22            }
23            count++;
24        }
25        System.out.println(count);
26    }
27 }
28

```

limit=3

1	1	1	2	2	3	3
0	1	2	3	4	5	6

$1 + 1 \leq \text{limit}$

i

j

count = 0

2

$2 \leq 2$  ✓

$1 + 3 \leq 3$

$2 + 1 \leq 3$

$2 + 1 \leq 3$

3

3

12

12

1