Reach Target

→ A → sorted

2+2=4

tar = 4

-1 1 2 3 4 5

0 1 2 3 4 5

i

j

i < j

sum = A[i]+A[j]

s == tar
↳ print
i++ j--

s=5 ↓

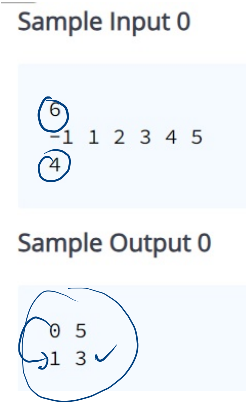
s=4

s = 4

s > tar
j--

s < tar
i++

i j

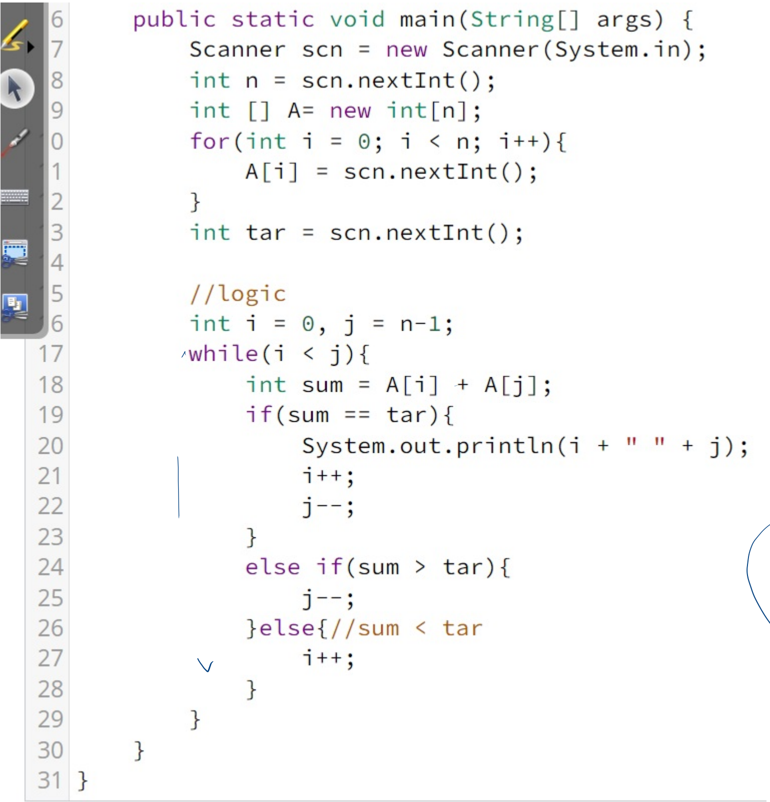


Sample Input 0

```
6
-1 1 2 3 4 5
4
```

Sample Output 0

```
0 5
1 3
```



```
    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 tar = scn.nextInt();

        //logic
        int i = 0, j = n-1;
        while(i < j){
            int sum = A[i] + A[j];
            if(sum == tar){
                System.out.println(i + " " + j);
                i++;
                j--;
            }
            else if(sum > tar){
                j--;
            }else{//sum < tar
                i++;
            }
        }
    }
}
```

tar == 4

-1	1	2	3	4	5
0	1	2	3	4	5

0 < 5 ✓

1 < 4 ✓

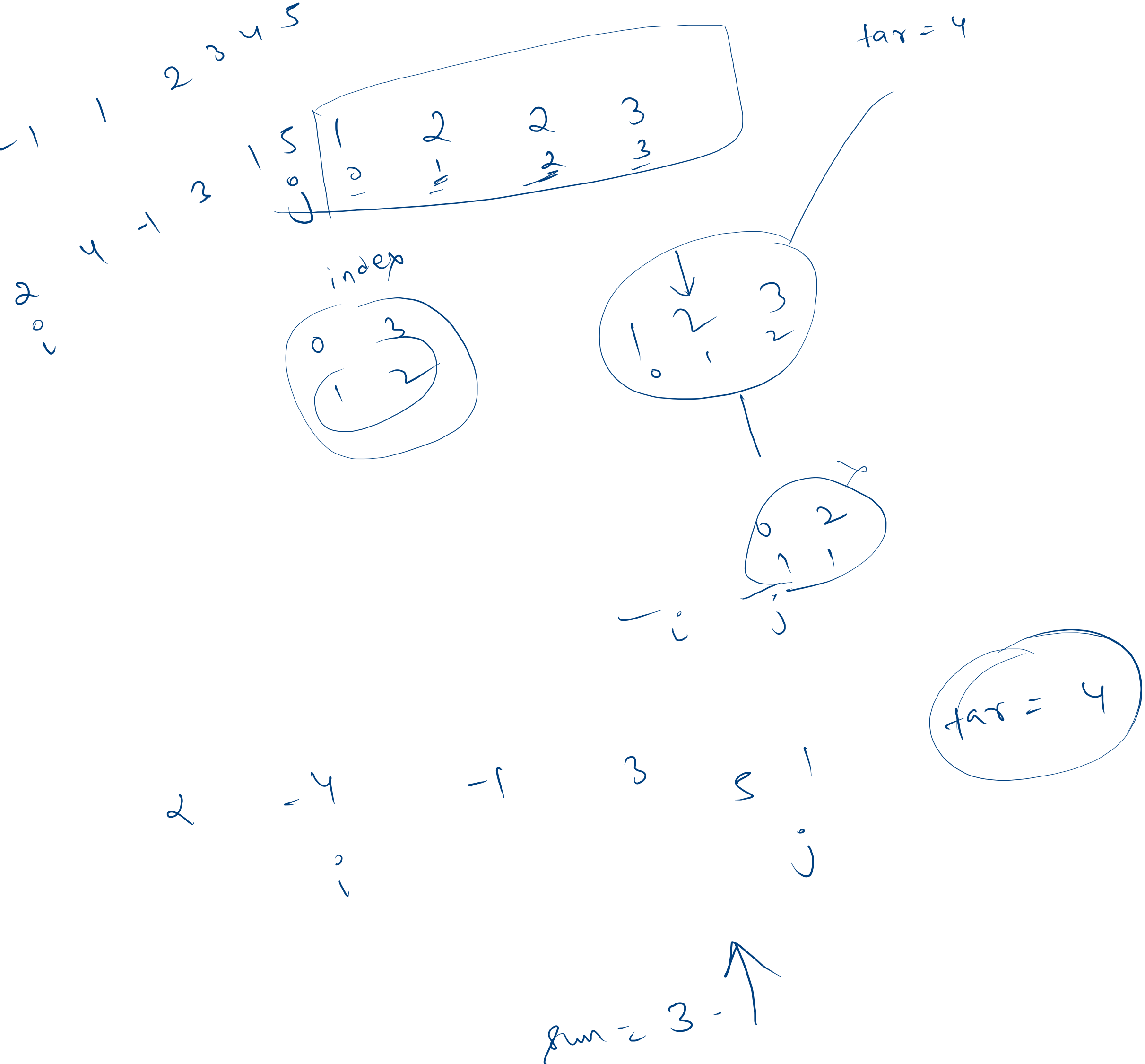
1 < 3 ✓

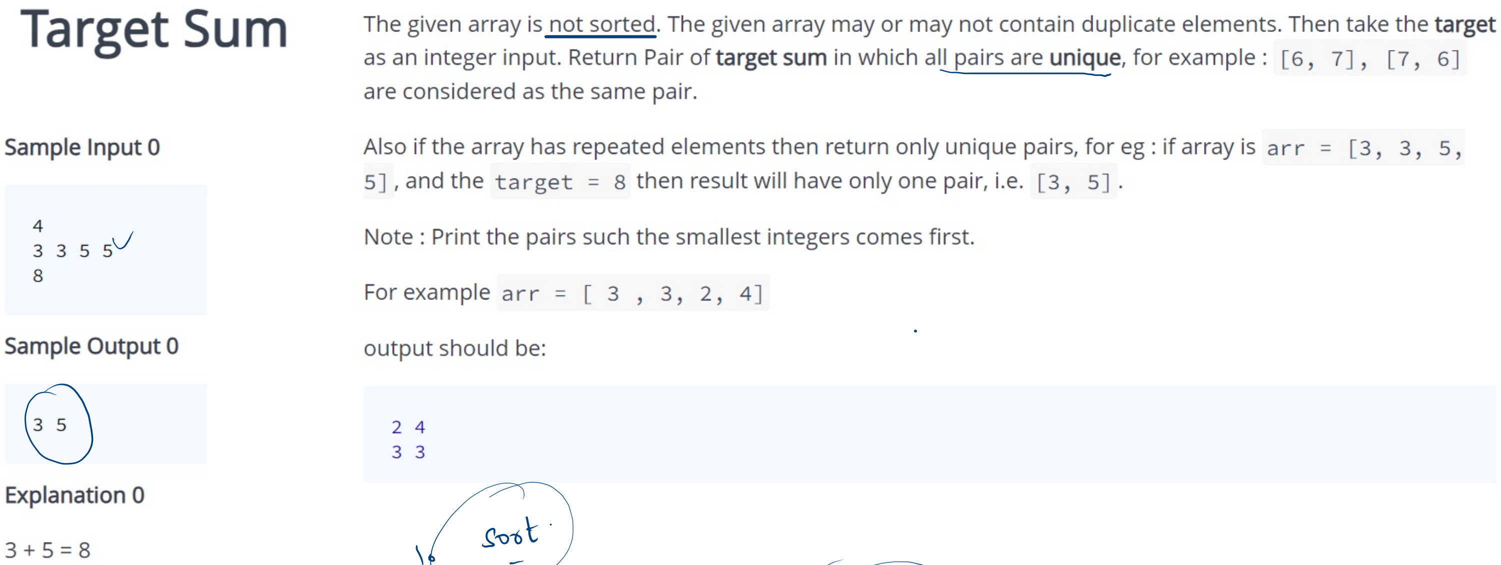
sum = -1 + 5 = 4

sum = 5

sum = 4

0	5
1	3


-1 1 2 3 4 5
tar = 4
1 2 2 3
0 1 2 3
2 4 -1 3 1 5
i j
index
0 3
1 2
1 2 3
0 1 2
0 2
1 1
-1 i j
tar = 4
2 -4 -1 3 5 1
i j
sum = 3



Target Sum

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.

For example `arr = [ 3 , 3, 2, 4]`

output should be:

```
2 4
3 3
```

Sample Input 0

```
4
3 3 5 5✓
8
```

Sample Output 0

```
3 5
```

Explanation 0

3 + 5 = 8

1. sort

tar = 8

3 5

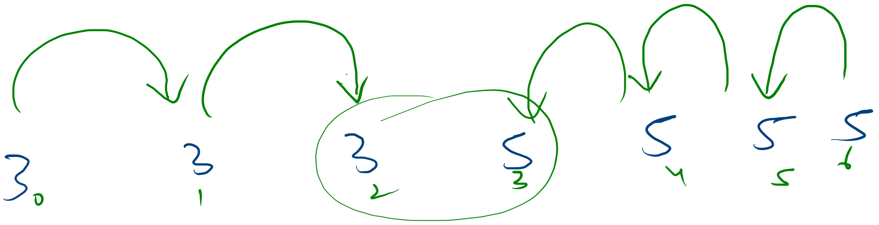
3 5

3_0 3_1 5_2 5_3

i j

s = 8

tar = 8



j

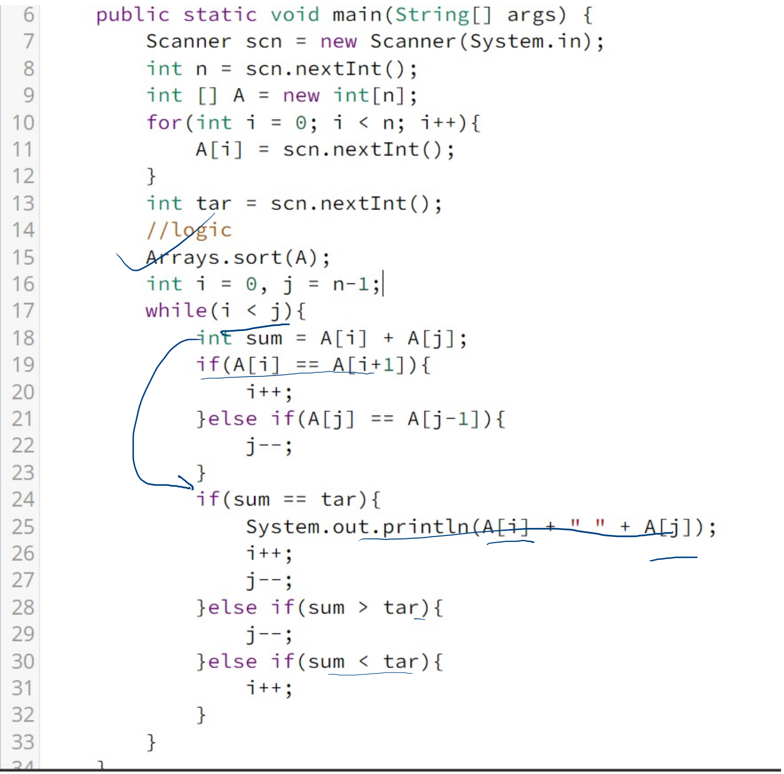
8 == tar

i

stop

$i < j$

3 5



```
    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 tar = scn.nextInt();
        //logic
        Arrays.sort(A);
        int i = 0, j = n-1;
        while(i < j){
            int sum = A[i] + A[j];
            if(A[i] == A[i+1]){
                i++;
            }else if(A[j] == A[j-1]){
                j--;
            }
            if(sum == tar){
                System.out.println(A[i] + " " + A[j]);
                i++;
                j--;
            }else if(sum > tar){
                j--;
            }else if(sum < tar){
                i++;
            }
        }
    }
```

tar =8

1	2	3	3	3	5	5	5	5	6
0	1	2	3	4	5	6	7	8	9

i

j

8

2 6

1 2 3 3 3 5 5 5 5 6

0 1 2 3 4 5 6 7 8 9

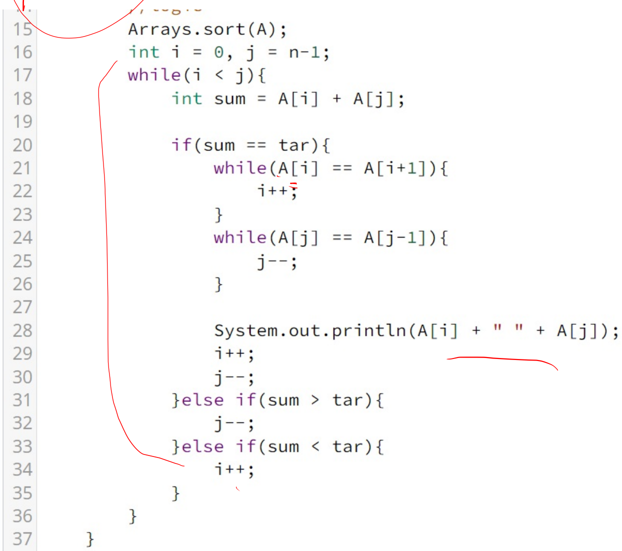
tar = 8

j

2 6

3 5

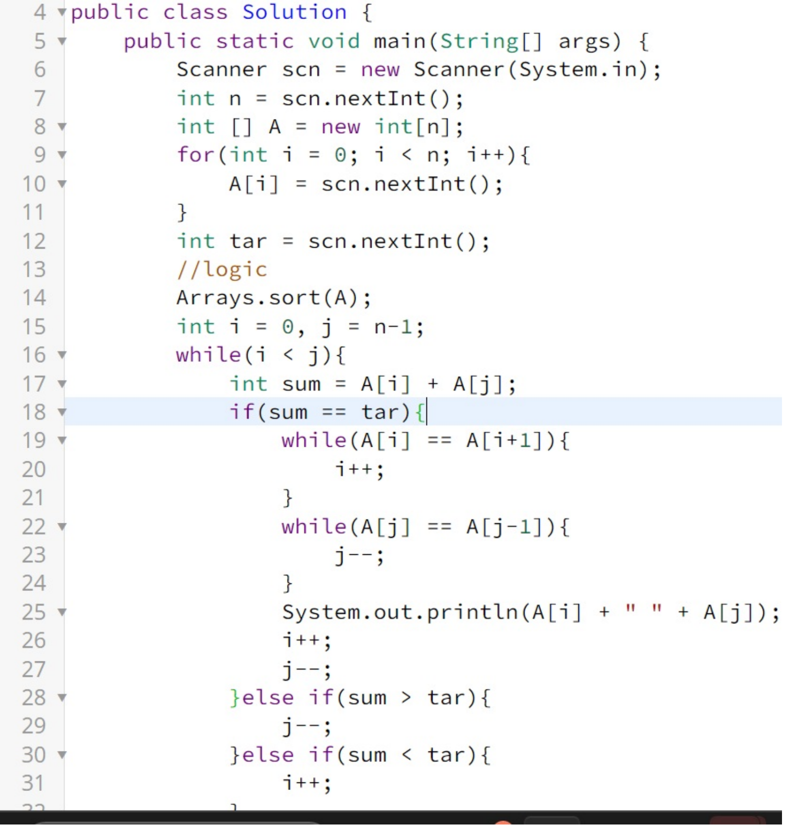
S = 3+5 = 8



```
        // logic
        Arrays.sort(A);
        int i = 0, j = n-1;
        while(i < j){
            int sum = A[i] + A[j];

            if(sum == tar){
                while(A[i] == A[i+1]){
                    i++;
                }
                while(A[j] == A[j-1]){
                    j--;
                }

                System.out.println(A[i] + " " + A[j]);
                i++;
                j--;
            }else if(sum > tar){
                j--;
            }else if(sum < tar){
                i++;
            }
        }
    }
```



```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();
        }
        int tar = scn.nextInt();
        //logic
        Arrays.sort(A);
        int i = 0, j = n-1;
        while(i < j){
            int sum = A[i] + A[j];
            if(sum == tar){
                while(A[i] == A[i+1]){
                    i++;
                }
                while(A[j] == A[j-1]){
                    j--;
                }
                System.out.println(A[i] + " " + A[j]);
                i++;
                j--;
            }else if(sum > tar){
                j--;
            }else if(sum < tar){
                i++;
            }
```

1	2	3	3	3	5	5	5	5	6
0	1	2	3	4	5	6	7	8	9

tar = 8

Boats to Save people (LC-881) / Count Boats.

A →

wt.

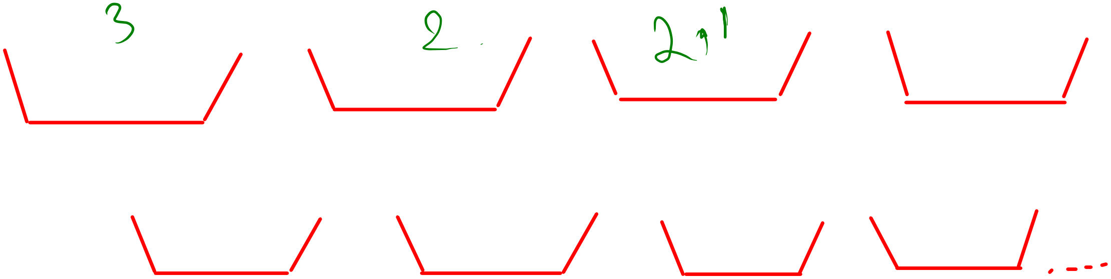
3	2	1	2
0	1	2	3

1 boat at max. '3' wt

cap.

2 people in a boat allowed

Ans = 3.



count min. no. of boat to ship all people cap = 3.

1 2 2 3

i j

$s = \underline{1+3}$

$\underline{s < cap.}$ i++ , j--

s > cap

↳ only one person

j--

cap = 3.

ans = ~~0~~ ~~1~~ ~~2~~ 3

1 2 2 3

j i

$i \le j$ false →

$i < j$

$s = 1+2 = 3$

$3 \le 3$

$s > cap$

↳ ans++

j - -

$s = 2+2$

$4 > 3.$

$s == cap$

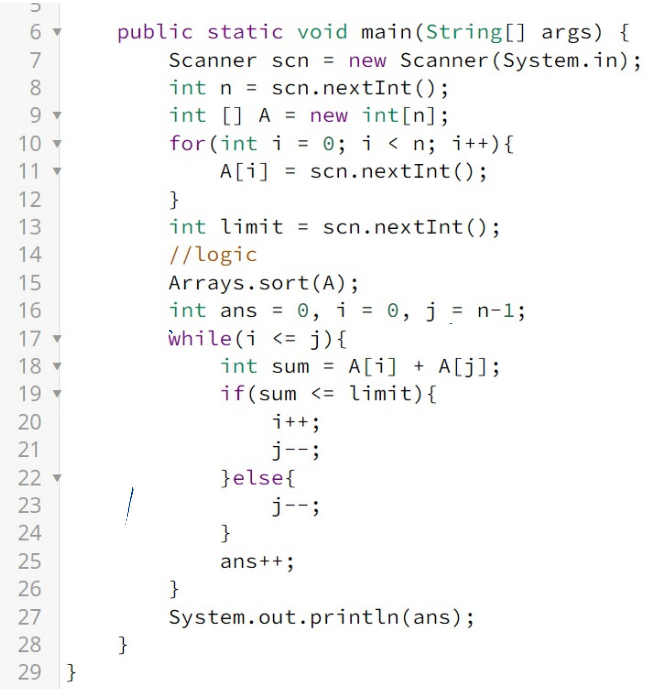
↳ ans++

i++

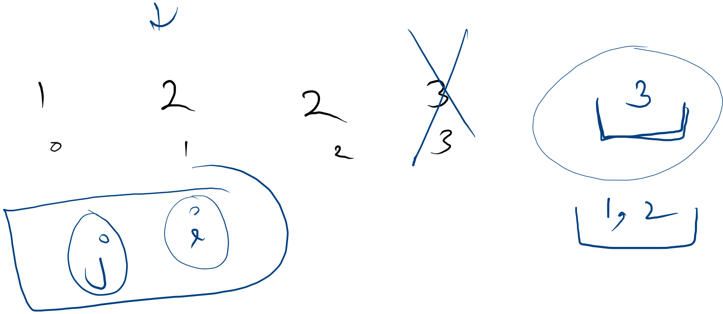
j - -

ans = ~~0~~ ~~1~~ ~~2~~ 3

limit = 3



```java
    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 limit = scn.nextInt();
        //logic
        Arrays.sort(A);
        int ans = 0, i = 0, j = n-1;
        while(i <= j){
            int sum = A[i] + A[j];
            if(sum <= limit){
                i++;
                j--;
            }else{
                j--;
            }
            ans++;
        }
        System.out.println(ans);
    }
}
```



8 = 3

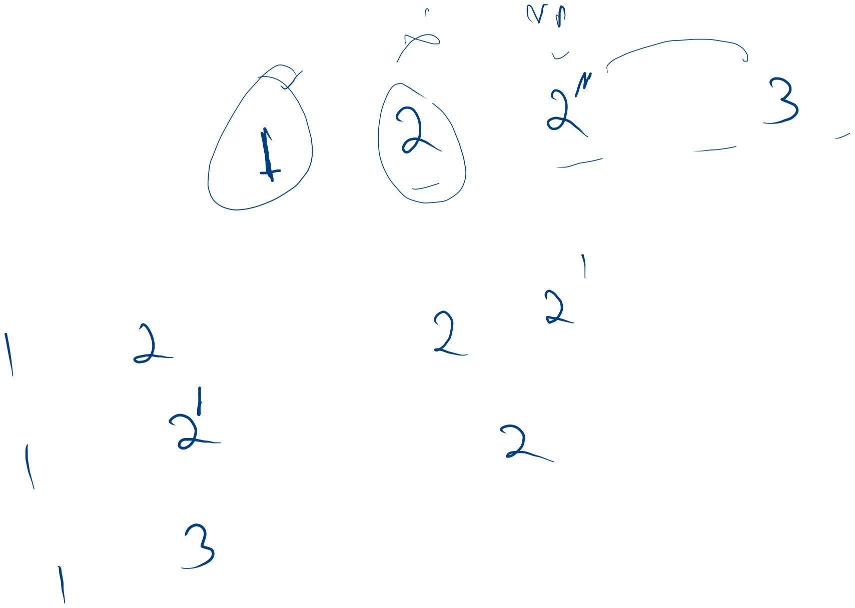
0 ≤ 2 ✓

1 ≤ 1

8 ≤ 4

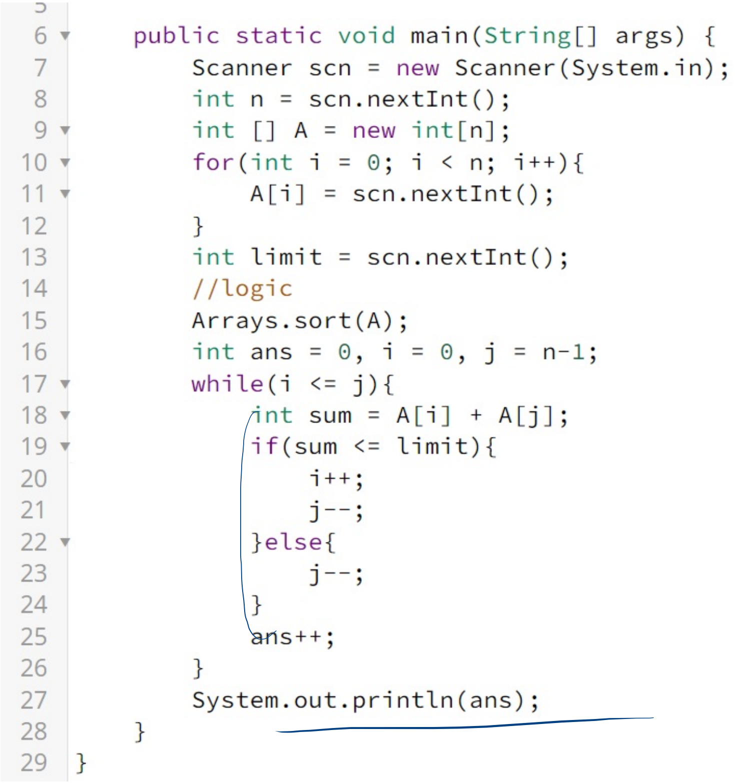
4 ≤ 3

i ≤ j


1
2
2'
3
1
2
2
2'
1
2'
2
1
3

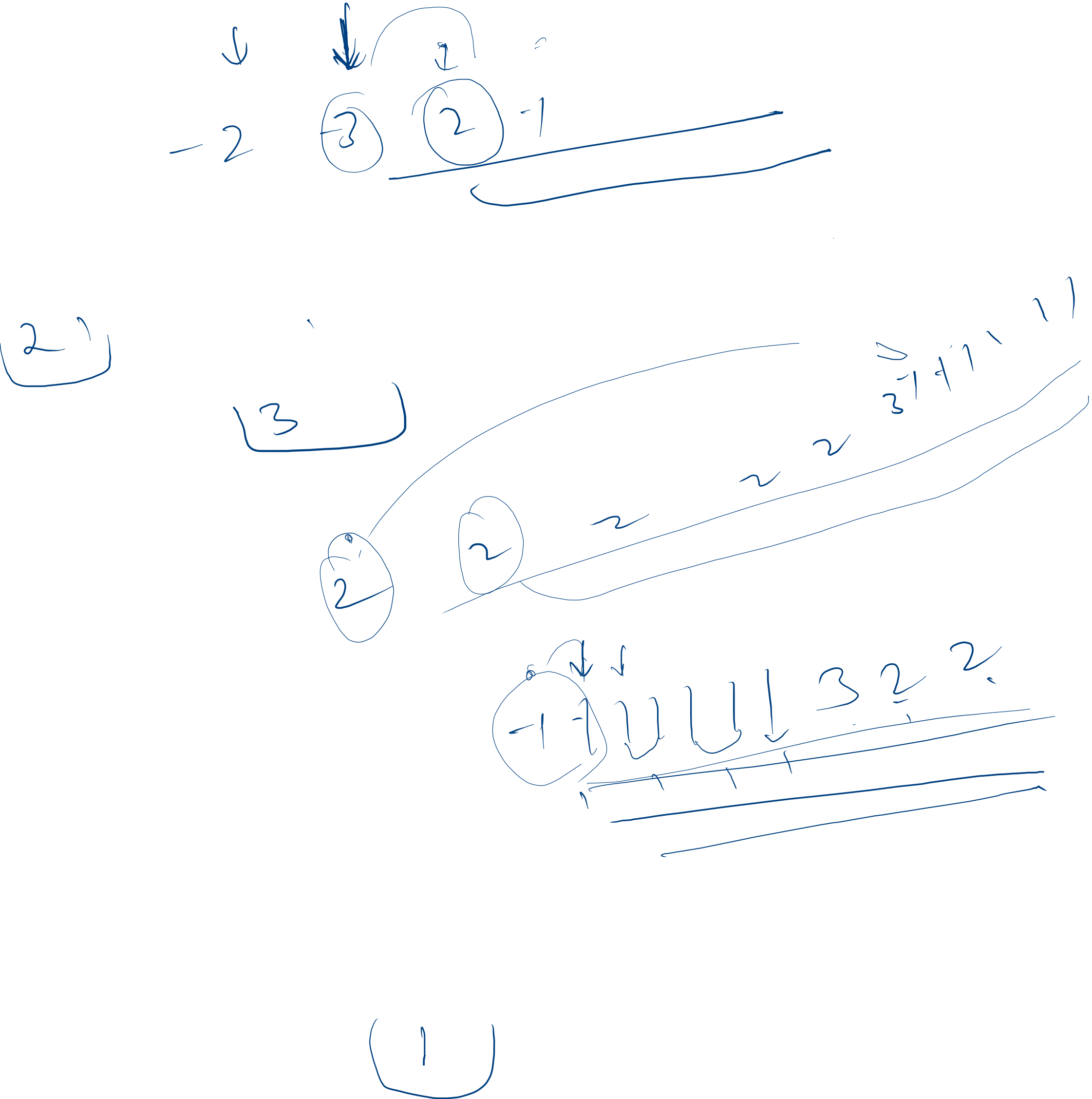
limit = 3

ans = 0



```
    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 limit = scn.nextInt();
        //logic
        Arrays.sort(A);
        int ans = 0, i = 0, j = n-1;
        while(i <= j){
            int sum = A[i] + A[j];
            if(sum <= limit){
                i++;
                j--;
            }else{
                j--;
            }
            ans++;
        }
        System.out.println(ans);
    }
}
```

2
0
i
0
j
?
0 ≤ 0
i <= j
0 < 0
b
0

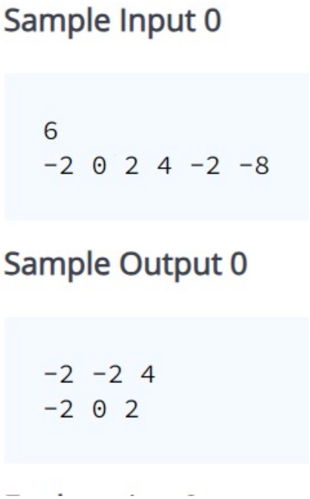

-2
3
2
-1
2 1
3
2
2
2
2
2
3
-1
-1
3
2
2
1

3 Sum / LC-15

tar=0

x
↓

-2	0	2	4	-2	-8
0	1	2	3	4	5



Sample Input 0

```
6
-2 0 2 4 -2 -8
```

Sample Output 0

```
-2 -2 4
-2 0 2
```

$x + y + z = 0$

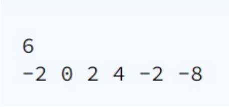
let $x = -2$

$\boxed{y + z = 2}$ → Reach Target

$x + y + z = 0$

$-2 + y + z = 0$

$\boxed{y + z = +2}$



```
6
-2 0 2 4 -2 -8
```

			c		
-8	-2	-2	0	2	4
0	1	2	3	4	5

$\overset{0}{\boxed{A[c]}} + A[i] + A[j] == 0$

$$\boxed{A[i] + A[j] = \underline{-A[c]}}$$