Note :-

- Sort 01
- sort 012

This trick of these 2 question can be used for any kind of elements

⇒ Variation of 2 pointer approach

Reach Target

$n = 6$

$arr = [\overset{0}{-1}, \overset{1}{1}, \overset{2}{2}, \overset{3}{3}, \overset{4}{4}, \overset{5}{5}]$

Brute force approach :- n^2

$target = 4$

equations :- $arr[i] + arr[j] == target$

take advantage of sorted array

$$arr = \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline [-5, & -1, & 1, & 2, & 3, & 4, & 5] \end{array} \quad t = 4$$

↑ ↑

i° j°

$T.C = O(n)$

loop i < j

sum of i and j element = ~~0~~ ~~4~~ ~~5~~ 4

1, 6

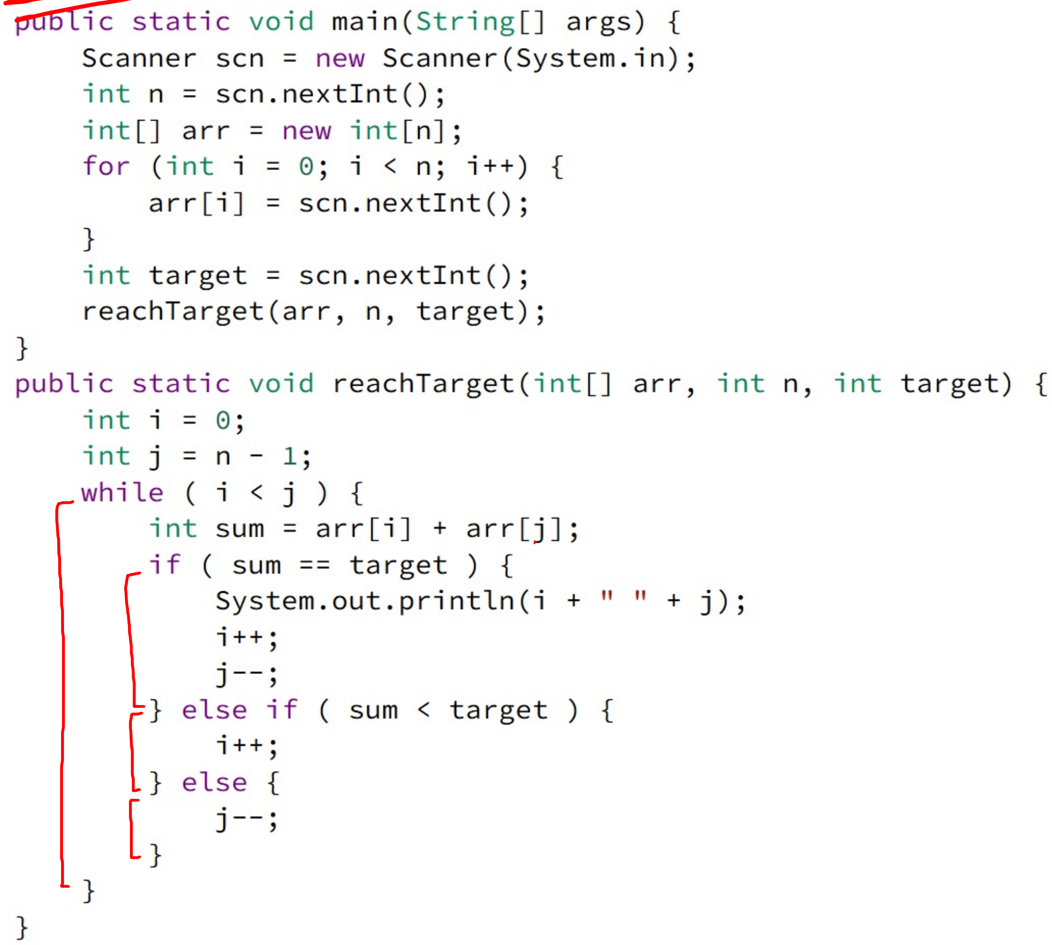
2, 4

template

Note :- this will work only when array is sorted

```
int i = 0;
int j = n-1;
while (i<j) {                    (i<=j)
        int sum = arr[i] + arr[j];
        if (sum == target){
              print (i + " " + j);
              i++, j--;
        } else if( sum < target){
              i++;
        } else if( sum > target) {
              j--;
        }
}
```

code



```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();
    reachTarget(arr, n, target);
}
public static void reachTarget(int[] arr, int n, int target) {
    int i = 0;
    int j = n - 1;
    while ( i < j ) {
        int sum = arr[i] + arr[j];
        if ( sum == target ) {
            System.out.println(i + " " + j);
            i++;
            j--;
        } else if ( sum < target ) {
            i++;
        } else {
            j--;
        }
    }
}
```

target = 1

arr = [-4, -2, 0, 1, 3, 8]

j i

sum = ~~4~~ ~~-1~~ ~~1~~ 1

Target Sum (Very Imp)

- → array is not sorted
- → may contains duplicate
- → only print unique pair

3 3 2 4

arr = [2 , 3 , 3 , 4]

↑ ↑ ↑ ↑

i i j j

target = 6

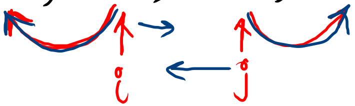
sum = ~~6~~ 6

Note :- sort the array first

2, 4

3, 3

arr = [1, 3, 3, 5, 5, 7, 8]   target = 8



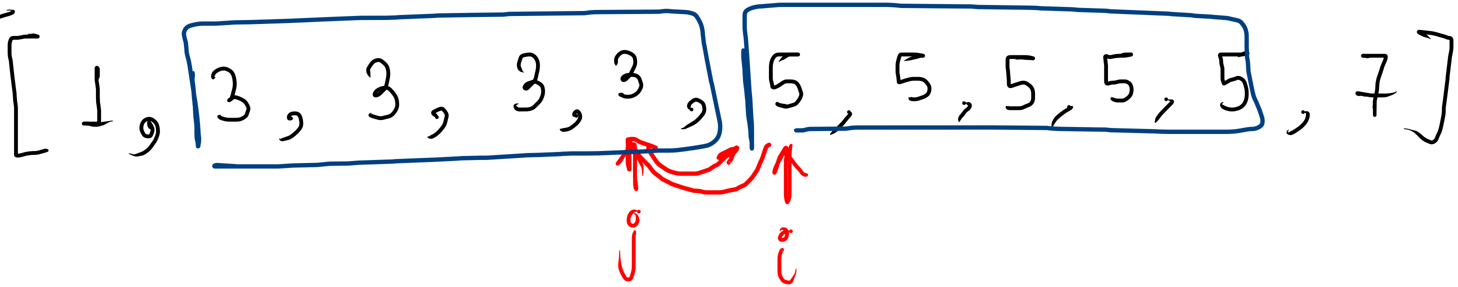
psudo code

```
1) sort array
2) declare i = 0, j = n-1
3) loop until i < j
    3.1) find sum = arr[i] + arr[j];
    3.2) check if sum == target
         Syso(arr[i] + "  " + arr[j]);
         i++, j--;
    while [ ~~if~~ (arr[i] == arr[i-1]){
                i++;
          }
    while [ ~~if~~ (arr[j] == arr[j+1]){
                j--;
          }
    3.3) check if sum < target
             i++
    3.4) check if sum > target
             j--
```

sum = ~~9~~ ~~8~~ ~~8~~ 8

o/p	
1	7
3	5

ex :-



taget = 8

sum = 8

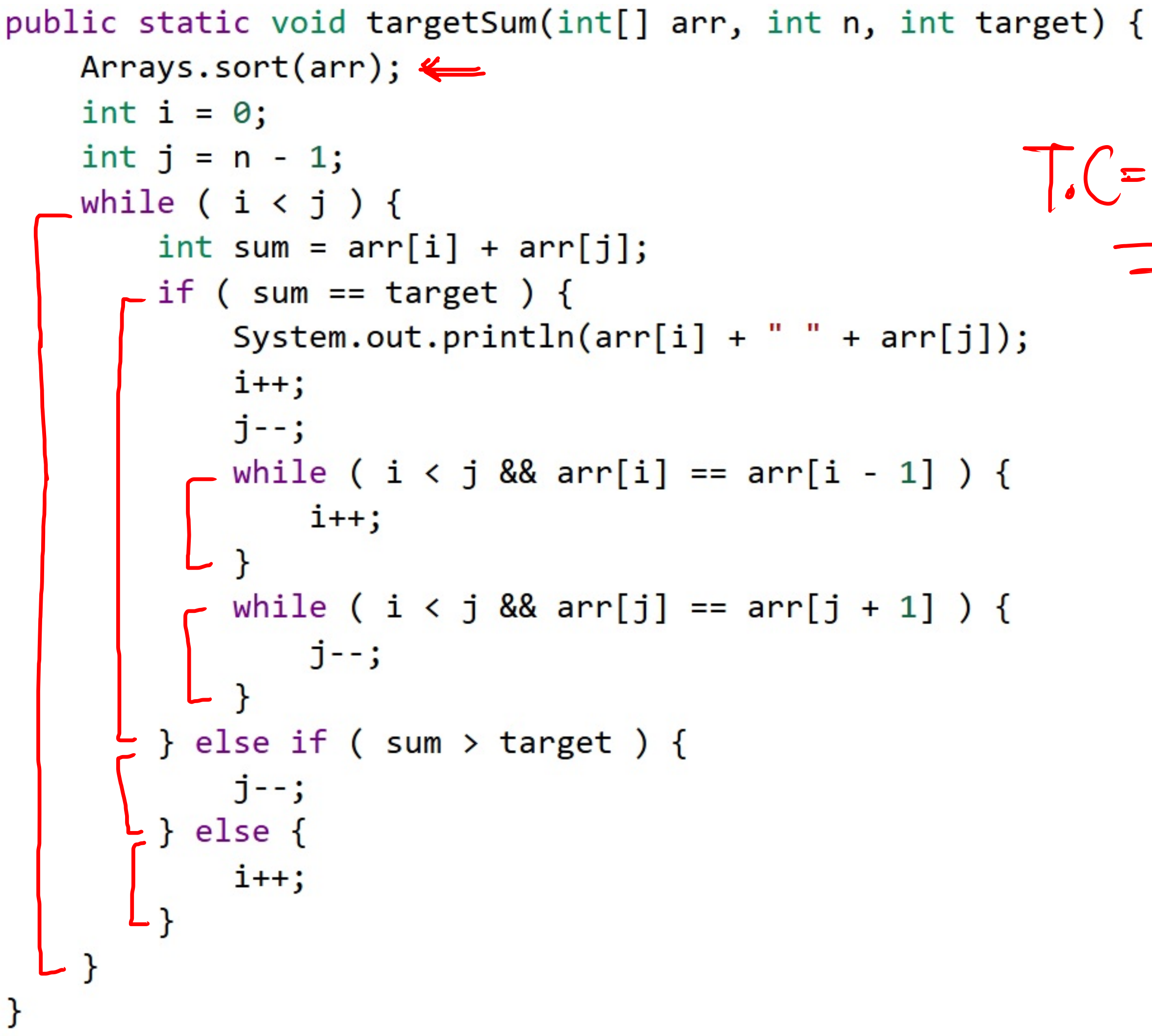
sum = ~~8~~ ~~8~~ 8

1	7
3	5

psudo code

```
1) sort array
2) dedare i = 0, j = n-1
3) loop until i < j
    3.1) find sum = arr[i] + arr[j];
    3.2) check if sum == taget
        Syso( arr[i] + "  " + arr[j] );
        i++, j--;
        while( arr[i] == arr[i-1] ){
            i++;
        }
        while( arr[j] == arr[j+1] ){
            j--;
        }
    3.3) check if sum < taget
        i++
    3.4) check if sum > taget
        j--
```

Code



```
public static void targetSum(int[] arr, int n, int target) {
    Arrays.sort(arr);
    int i = 0;
    int j = n - 1;
    while ( i < j ) {
        int sum = arr[i] + arr[j];
        if ( sum == target ) {
            System.out.println(arr[i] + " " + arr[j]);
            i++;
            j--;
            while ( i < j && arr[i] == arr[i - 1] ) {
                i++;
            }
            while ( i < j && arr[j] == arr[j + 1] ) {
                j--;
            }
        } else if ( sum > target ) {
            j--;
        } else {
            i++;
        }
    }
}
```

$T.C = O(n + n\log n)$

V. Imp

3 Sum

$$arr[i] + arr[j] + arr[k] = 0$$

$$\boxed{arr[i] + arr[j] = -1 * arr[k]}$$

$$arr[i] + arr[j] = \text{target}$$