

Revision.

↳ Trailing 0 in $n!$ $(n)!$

↳ $\gcd(\min, \max)$. ✓

1. Trailing zero in $n!$.

$n=25$.

$25!$

0 → 5, 2

2 3 4 5 6 7 8 9 10 11 12 13
14 15 16 17 18 19 20 21
22 23 24 25 26

5, 2 → count of 2 → 5

$$\frac{25}{5} = 5 + \frac{25}{25}$$

$$125 = 25 \left(\frac{125}{5} \right) + \left(\frac{125}{25} \right) + \left(\frac{125}{125} \right) \} = 31$$

2. $\gcd(\max, \min)$.

8 10 9 11 12
min max

gcd

8, 12

$\gcd \leq 8$

~~8~~ ~~7~~ ~~6~~ ~~5~~ 4 ✓
8 % 8 = 0 ✓
12 % 8 = 4 ✗

Target Sum

Problems	Submissions	Leaderboard	Discussions
----------	-------------	-------------	-------------

The given array is not sorted. The given array may or may not contain duplicate elements. Then take the target as an integer target. Return Pair of targets 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, 3, 5, 5}, and the target = 8 then result will have only one pair, i.e. (3, 5).

Note: Print the pairs such that the smallest integer comes first.

For example arr = {3, 3, 3, 2, 4}

output should be:

2 4
3 3

Reach Target $\rightarrow O(n^2)$
 ,, 2 $\rightarrow O(n)$.

Solution

tar = 8

tar = 8

arr \rightarrow 1 5 2 3 3 4 1

\downarrow

Arrays.sort(arr)

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

(8) ✓

while (i < j)

3 5

sum = (6) (7) (8)

(8) ✓ 1 1 2 2 2 3 3 4 5 5 6 6 6
 0 1 2 3 4 5 6 7 8 9 10 11 12
 i j
 8 = A(i) + A(j) = 7 + 1

while (A(i) == A(i+1))

8 = A(i) + A(j) = 7 + 1

```

import java.util.*;

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();

        Arrays.sort(A);

        int i = 0;
        int j = n-1;

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

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

tar = 4

index of b/w

2 2 2 2 2 4
 0 1 2 3 4 5
 i j
 4 = A(i) + A(j) = 2 + 2

A(i) = A(i+1)
 A(4) = A(5)

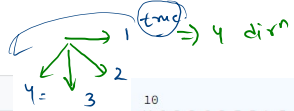
CrossWord

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

You are given a matrix of char And a string(word) You have to check if the word is present in matrix or not Word can be in any of the Following Directions:

- The word can be Horizontal --- From left to right.
- The word can be Vertical - From Top to bottom.
- The word can be along the Big-Diagonal wise - . From north-west to south-east or From north-east to south-west.

GREEN



true.

10
W C R P S R I J L J
D R T I B U B R I Y
Y E T N L P K O S T
J D S K U O O S O B
<u>G B A F E R Y I B R</u>
C L C B H A Q Y E O
D A W C R N L X X W
I C T E L G J N S N
Q K Q K Y E L L O W
H A O K X G R E E N
GREEN

false

G

0, 2

1 3

2 4

3 5

	0	1	2	3	4	5	6
0			G				
1				R			
2					E		
3						E	
4							N
5							
					S		

E

G.

+ (1, 5) -
+ (2, 4)

	0	1	2	3	4	5	6	7
0								
1						G		
2					R			
3				E				
4			E					
5		N						

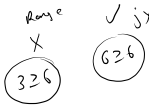


A diagram of a cell with a vertical line representing a membrane. The left side is dark, and the right side is light. Arrows point from the light side to the dark side, indicating movement.

GREEN
(
K)



int i, int j	
2	1
3	1
4	
5	



6×6
 $n=6$



```

}
public static boolean dirFour(char [][] A , String word, int i, int j){
    // from NE to SW
    i++;
    j--;
    for(int k = 1; k < word.length(); k++){
        if ( i >= A.length || j < 0){
            return false;
        }
        else if(word.charAt(k) != A[i][j]){
            return false;
        }
        i++;
        j--;
    }
    return true;
}

```

9 R E E N
 0 1 2 3 4
 k=4 2 3 4 5

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

d s
 d b

ij

	0	1	2	3	4
0	A	G	P	Q	G
1	R	T	A	R	A
2	X	V	E	T	N
3	Y	E	R	S	O
4	N	T	S	Q	P

