



## Following the question style make an elaborate question on this topic

### Triangle Count from Side Lengths

Count how many distinct triplets ( $i < j < k$ ) from  $L$  can form a triangle (i.e., satisfy triangle inequality:  $a + b > c$ , for sorted sides).

Based on the ITPEC FE exam question format from the attached document, here's an elaborate question on counting triangle triplets from side lengths:<sup>[1]</sup>

**Q21.** From the answer group below, select the correct combination of answers to be inserted into **A** through **C** in the program.

The function **countTriangles** receives an integer array **L** containing side lengths and returns the count of distinct triplets (**i, j, k**) where  $i < j < k$  that can form valid triangles. A valid triangle must satisfy the triangle inequality: for any three sides **a, b**, and **c** where  $a \leq b \leq c$ , the condition  $a + b > c$  must hold.

For instance, if **L** contains the values **\*\*\*\***, the function should return **4** because the following triplets form valid triangles:<sup>[1]</sup>

- Indices (0, 1, 2): sides 4, 2, 3 → sorted as 2, 3, 4 →  $2 + 3 > 4$  ✓
- Indices (0, 2, 3): sides 4, 3, 4 → sorted as 3, 4, 4 →  $3 + 4 > 4$  ✓
- Indices (0, 1, 3): sides 4, 2, 4 → sorted as 2, 4, 4 →  $2 + 4 > 4$  ✓
- Indices (2, 3, 4): sides 3, 4, 7 → sorted as 3, 4, 7 →  $3 + 4 > 7$  is false ✗
- Indices (1, 2, 3): sides 2, 3, 4 → sorted as 2, 3, 4 →  $2 + 3 > 4$  ✓

Here, the array index starts at 0. The function **minOfThree** returns the minimum value among three integers, and **maxOfThree** returns the maximum value among three integers.

### Program

```
integer countTriangles(integer[] L)
    integer n ← number of elements in L
    integer count ← 0
    integer i, j, k
    integer a, b, c

    for increase i from 0 to n - 3 by 1
        for increase j from 「A」 to n - 2 by 1
            for increase k from 「B」 to n - 1 by 1
                a ← minOfThree(L[i], L[j], L[k])
```

```

        c ← maxOfThree(L[i], L[j], L[k])
        b ← L[i] + L[j] + L[k] - a - c

        if 「C」
            count ← count + 1
        endif
    endfor
endfor
endfor

return count

```

### Answer group

	A	B	C
a	i	j	$a + b > c$
b	i	j	$a + c > b$
c	i	$j + 1$	$a + b > c$
d	i	$j + 1$	$a + c > b$
e	$i + 1$	j	$a + b > c$
f	$i + 1$	j	$a + c > b$
g	$i + 1$	$j + 1$	$a + b > c$
h	$i + 1$	$j + 1$	$a + c > b$

**Answer: g**

### Explanation:

To ensure we examine distinct triplets where  $i < j < k$ , the loop variables must be properly constrained:

- **A** should be  $i + 1$  to ensure  $j > i$
- **B** should be  $j + 1$  to ensure  $k > j$
- **C** should be  $a + b > c$  because after sorting the three sides such that  $a \leq b \leq c$ , the triangle inequality requires the sum of the two smaller sides to exceed the largest side

This approach systematically checks all unique triplets in the array without repetition, correctly identifies the minimum, middle, and maximum values, and applies the triangle inequality test.<sup>[1]</sup>

