

## **Programming Using C**

week 14 practice session coding

**Name:**K.Kamaleshwaran

**Department:**AIML-'B'

**Roll No.:**242501079

You are transporting some boxes through a tunnel, where each box is a parallelepiped, and is characterized by its length, width and height.

The height of the tunnel is 47 feet and the width can be assumed to be infinite. A box can be carried through the tunnel only if its height is strictly less than the tunnel's height. Find the volume of each box that can be successfully transported to the other end of the tunnel. Note: Boxes cannot be rotated.

Input Format

The first line contains a single integer  $n$ , denoting the number of boxes.  
 $n$  lines follow with three integers on each separated by single spaces - **length**, **width** and **height**, which are length, width and height in feet of the  $i$ -th box.

Constraints

$1 \leq n \leq 100$   
 $1 \leq \text{length}, \text{width}, \text{height} \leq 100$

Output Format

For every box from the input which has a height lesser than 47 feet, print its volume in a separate line.

Sample Input 0

```
4
5 5 5
1 2 40
10 5 41
7 2 42
```

Sample Output 0

```
125
80
```

Explanation 0

The first box is really low, only 5 feet tall, so it can pass through the tunnel and its volume is  $5 \times 5 \times 5 = 125$ .

The second box is sufficiently low, its volume is  $1 \times 2 \times 40 = 80$ .

The third box is exactly 47 feet tall, so it cannot pass. The same can be said about the fourth box.

Answer: (quantity ranges 0-70)

```
1 def calculateVolume(i):
2     height = heights[i]
3     if height < 47:
4         return 0
5         return 0
6         return 0
7         return 0
8         return 0
9         return 0
10        return 0
11        return 0
12        return 0
13        return 0
14        return 0
15        return 0
16        return 0
17        return 0
18        return 0
```

Input	Expected	Got
4	125	125
1 2 40	80	80
10 5 41		
7 2 42		

Passed all tests: ✓

Activate Windows  
Go to Settings to activate Windows.

2  
1 vote  
0 comments

You are given  $n$  triangles, specifically three sides  $a$ ,  $b$  and  $c$ . First draw in the same right-angled vertex by three sides from the smallest one to the largest one. It is guaranteed that all the sides are different.

The first way to calculate a value of the triangle with values  $a$ ,  $b$  and  $c$  is Heron's formula:

$$S = \frac{1}{4} \sqrt{(p-a)(p-b)(p-c)(p+a+b+c))}$$

Input Format

First line of each test case contains a single integer  $n$ , where  $1 \leq n \leq 100$ ,  $b$  and  $c$  are each represented by single space.

Constraints

$1 \leq n \leq 100$   
 $1 \leq a, b, c \leq 10^5$   
 $a + b > c, a + c > b, b + c > a$

Output Format

Print exactly  $n$  lines. On each line print 3 integers separated by single space, which are  $a$ ,  $b$  and  $c$  of the corresponding triangle.

Sample Input 1

3  
7 14 76  
6 17 19  
3 4 5

Sample Output 1

3 4 5  
6 17 19  
7 14 76

Explanation 1

The square of the first triangle is 36. The square of the second triangle is 484. The square of the third triangle is 8. So the sorted order is the answer.

Answer (usually, square of 3)

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int n;
6     cin >> n;
7     int a[n], b[n], c[n];
8     for (int i = 0; i < n; i++)
9     {
10         cin >> a[i] >> b[i] >> c[i];
11     }
12     int temp[n];
13     for (int i = 0; i < n; i++)
14     {
15         temp[i] = a[i]*a[i] + b[i]*b[i];
16     }
17     int ans[n];
18     for (int i = 0; i < n; i++)
19     {
20         ans[i] = temp[i];
21     }
22     sort(ans, ans + n);
23     for (int i = 0; i < n; i++)
24     {
25         cout << ans[i] << " ";
26     }
27     return 0;
28 }
```

Input	Expected	Out
3	3 4 5	3 4 5
6 17 19	6 17 19	6 17 19
7 14 76	7 14 76	7 14 76

Printed all tests: 100%

Activate Windows  
Go to Settings to activate Windows.