

## Week-14-Structures and Unions

### Week-14-01-Practice Session-Coding

Question 1

Correct

Flag question

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 **41** 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<sub>i</sub>**, **width<sub>i</sub>** and **height<sub>i</sub>**, which are length, width and height in feet of the *i*-th box.

Constraints

**1 ≤ n ≤ 100**

**1 ≤ length<sub>i</sub>, width<sub>i</sub>, height<sub>i</sub> ≤ 100**

Output Format

For every box from the input which has a height lesser than **41** feet, print its volume in a separate line.

Sample Input 0

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

[Source code](#)

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2
3 int main(){
4     int n;
5     scanf("%d",&n);
6
7     struct boxes{
8         int l,b,h,v;
9     } box[n];
10
11    for(int i=0;i<n;i++){
12        scanf("%d %d %d",&box[i].l,&box[i].b,&box[i].h);
13        if(box[i].h<41){
14            box[i].v=box[i].l*box[i].b*box[i].h;
15            printf("%d\n",box[i].v);
16        }
17    }
18    return 0;
19 }
```

Result

|   | <b>Input</b> | <b>Expected</b> | <b>Got</b> |   |
|---|--------------|-----------------|------------|---|
| ✓ | 4            | 125             | 125        | ✓ |
|   | 5 5 5        | 80              | 80         |   |
|   | 1 2 40       |                 |            |   |
|   | 10 5 41      |                 |            |   |
|   | 7 2 42       |                 |            |   |

Passed all tests! ✓

Question 2  
Correct  
Flag question

You are given  $n$  triangles, specifically, their sides  $a_i$ ,  $b_i$  and  $c_i$ . Print them in the same style but sorted by their areas from the smallest one to the largest one. It is guaranteed that all the areas are different.

The best way to calculate a volume of the triangle with sides  $a$ ,  $b$  and  $c$  is Heron's formula:

$$S = \sqrt{p * (p - a) * (p - b) * (p - c)} \text{ where } p = (a + b + c) / 2.$$

Input Format

First line of each test file contains a single integer  $n$ .  $n$  lines follow with  $a_i$ ,  $b_i$  and  $c_i$  on each separated by single spaces.

Constraints

$$1 \leq n \leq 100$$

$$1 \leq a_i, b_i, c_i \leq 70$$

$$a_i + b_i > c_i, a_i + c_i > b_i \text{ and } b_i + c_i > a_i$$

Output Format

Print exactly  $n$  lines. On each line print 3 integers separated by single spaces, which are  $a_i$ ,  $b_i$  and  $c_i$  of the corresponding triangle.

Sample Input 0

```
3
7 24 25
5 12 13
3 4 5
```

Activate Windows

© 2014 Microsoft Corporation. All rights reserved. Microsoft, the Microsoft logo, Microsoft Visual Studio, and other Microsoft products and services mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

[Source code](#)

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 #include<math.h>
3
4 int main(){
5     int n;
6     scanf("%d",&n);
7
8     struct triangle{
9         int a,b,c;
10        int area;
11    } triangles[n];
12
13    for(int i=0;i<n;i++){
14        scanf("%d %d %d",&triangles[i].a,&triangles[i].b,&triangles[i].c);
15        int p=(triangles[i].a+triangles[i].b+triangles[i].c)/2;
16        triangles[i].area=sqrt(p*(p-triangles[i].a)*(p-triangles[i].b)*(p-triangles[i].c));
17    }
18
19    for(int i=0;i<n-1;i++){
20        for(int j=0;j<n-i-1;j++){
21            if(triangles[j].area>triangles[j+1].area){
22                struct triangle temp=triangles[j];
23                triangles[j]=triangles[j+1];
24                triangles[j+1]=temp;
25            }
26        }
27    }
28
29    for(int i=0;i<n;i++){
30        printf("%d %d %d\n",triangles[i].a,triangles[i].b,triangles[i].c);
31    }
32    return 0;
33 }
```

## Result

|   | Input                            | Expected                    | Got                         |   |
|---|----------------------------------|-----------------------------|-----------------------------|---|
| ✓ | 3<br>7 24 25<br>5 12 13<br>3 4 5 | 3 4 5<br>5 12 13<br>7 24 25 | 3 4 5<br>5 12 13<br>7 24 25 | ✓ |
|   |                                  |                             |                             |   |
|   |                                  |                             |                             |   |
|   |                                  |                             |                             |   |

Passed all tests! ✓