Week-14-Structures and unions



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 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 ≤ n ≤ 100

 $1 \le length_i$, width_i, height_i ≤ 100

Output Format

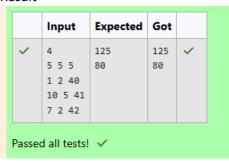
For every box from the input which has a height lesser than 41 feet, print its volume in a separate line.

Sample Input 0

Source code

```
#include <stdio.h>
2
    int main()
3 ₹ {
4
         int n;
         scanf("%d",&n);
5
6 ,
         for(int i=0;i<n;i++){</pre>
7
             int length,width,height;
             scanf("%d %d %d",&length,&width,&height);
8
9
             if(height < 41){
  int volume = length*width*height;</pre>
10
11
                  printf("%d\n",volume);
12
13
14
   1
15
```

Result



```
Question 2
Correct
Flag question
```

You are given \boldsymbol{n} triangles, specifically, their sides $\boldsymbol{a_b}$ $\boldsymbol{b_l}$ and $\boldsymbol{c_l}$. 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 $\boldsymbol{a_t}$ $\boldsymbol{b_t}$ and $\boldsymbol{c_t}$ is Heron's formula: $S = \vec{O} \, \boldsymbol{p} \, ^* (\boldsymbol{p} - \boldsymbol{a_l}) \, ^* (\boldsymbol{p} - \boldsymbol{b_l}) \, ^* (\boldsymbol{p} - \boldsymbol{c_l})$ where $\boldsymbol{p} = (\boldsymbol{a} + \boldsymbol{b} + \boldsymbol{c_l})/2$.

Input Format

First line of each test file contains a single integer \boldsymbol{n} , \boldsymbol{n} lines follow with $\boldsymbol{a_t}$, $\boldsymbol{b_t}$ and $\boldsymbol{c_t}$ on each separated by single spaces.

Constraints

I $\leq \boldsymbol{n} \leq 100$ I $\leq \boldsymbol{a_t}$, $\boldsymbol{b_t}$, $\boldsymbol{c_t} \leq 70$ $\boldsymbol{a_t} + \boldsymbol{b_t} \geq \boldsymbol{b_t}$ and $\boldsymbol{b_t} + \boldsymbol{c_t} \geq \boldsymbol{b_t}$ and $\boldsymbol{b_t} + \boldsymbol{c_t} \geq \boldsymbol{a_t}$ Output Format

Print exactly \boldsymbol{n} lines. On each line print 3 integers separated by single spaces, which are $\boldsymbol{a_t}$, $\boldsymbol{b_t}$ and $\boldsymbol{c_t}$ of the corresponding triangle.

Sample Input 0

Source code

5 12 13 3 4 5

```
1 |#include <stdio.h>
      #include <math.h>
  3
      #include <stdlib.h>
  4 v typedef struct {
          double area;
          int a,b,c;
   6
     }Triangle;
  8
  9 v double calculate_area(int a,int b,int c){
 10
          double p=(a+b+c)/2.0;
 11
          return sqrt(p*(p-a)*(p-b)*(p-c));
 12
 13 v int compare(const void*x,const void*y){
 14
          Triangle *t1 = (Triangle *)x;
          Triangle *t2 = (Triangle *)y;
 15
 16
          if(t1->area <t2->area)return -1;
 17
          if(t1->area >t2->area)return 1;
 18
          return 0;
 19
 20 v int main(){
         int n;
 21
          scanf("%d",&n);
 23
          Triangle triangles[n];
 24
 25 v
          for(int i=0;i<n;i++){</pre>
 26
              int a,b,c;
              scanf("%d %d %d",&a,&b,&c);
 27
 28
 29
              triangles[i].a = a;
 30
              triangles[i].b = b;
 31
              triangles[i].c = c;
 32
              triangles[i].area = calculate_area(a,b,c);
 33
          qsort(triangles, n, sizeof(Triangle),compare);
 34
 35
          for(int i=0;i<n; i++){</pre>
              printf("%d %d %d\n",triangles[i].a, triangles[i].b, triangles[i].c);
 37
 38
 39
          return 0;
 40
 41 }
```

Result

	Input	Expected	Got	
~	3	3 4 5	3 4 5	~
	7 24 25	5 12 13	5 12 13	
	5 12 13	7 24 25	7 24 25	
	3 4 5			