Question 1 Correct	You are transporting some boxes through a tunnel, where each box is a parallelepiped, and is characterized by its length, width and height.
№ Flag question	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 _i , width _i and height _i which are length, width and height in feet of the i-th box.
	Constraints
	1 ≤ n ≤ 100
	$1 \leq length_i$, $width_i$, $height_i \leq 100$
	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
	555
	1 2 40

Sample Output 0

125

80

Explanation 0

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

The second box is sufficiently low, its volume is $\mathbf{1} \times \mathbf{2} \times \mathbf{4} = \mathbf{80}$.

The third box is exactly **41** feet tall, so it cannot pass. The same can be said about the fourth box.

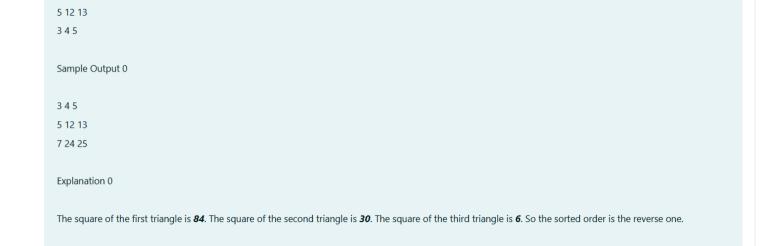
10 5 41 7 2 42

```
Answer: (penalty regime: 0 %)
      #include<stdio.h>
       int main()
           int n;
           scanf("%d",&n);
           for(int i=0;i<n;i++)</pre>
               int len,wid,hei;
   8
               scanf("%d %d %d",&len,&wid,&hei);
   9
               if(hei<41)
  10
  11 v
  12
                   int vol=len*wid*hei;
                   printf("%d\n",vol);
  13
  14
  15
  16
           return 0;
  17 }
```

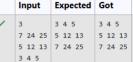
	Input	Expected	Got	
~	4	125	125	~
	5 5 5	80	80	
	1 2 40			
	10 5 41			
	7 2 42			

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 \boldsymbol{a} , \boldsymbol{b} and \boldsymbol{c} is Heron's formula:		
	$S = \ddot{O} p * (p - a) * (p - b) * (p - c)$ 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 ≤ n ≤ 100		
	$1 \leq a_{i}, b_{i}, c_{i} \leq 70$		
	$a_i + b_i > c_i$, $a_i + c_i > b_i$ 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		



```
|#include<stdio.h>
    #include<math.h>
    #include<stdlib.h>
    typedef struct{
        int a,b,c;
        double area;
    }triangle;
    double calculate area(int a,int b,int c)
9 ,
10
        double p=(a+b+c)/2.0;
        return sqrt(p*(p-a)*(p-b)*(p-c));
11
12
13
    int compare(const void *t1,const void *t2)
14 •
        triangle *tri1=(triangle*)t1;
15
16
        triangle *tri2=(triangle*)t2;
17
        if(tri1->area < tri2 ->area)
18
        return -1:
19
        if(tri1->area>tri2->area)
20
        return 1;
21
        return 0;
22
    int main()
24 +
25
        int n:
26
        scanf("%d",&n);
27
        triangle triangles[n];
28
        for(int i=0;i<n;i++)</pre>
29
30
            int a.b.c:
            scanf("%d %d %d",&a,&b,&c);
31
32
            triangles[i].a=a;
33
            triangles[i].b=b;
34
            triangles[i].c=c;
35
            triangles[i].area=calculate_area(a,b,c);
36
37
        qsort(triangles,n,sizeof(triangle),compare);
38
        for(int i=0;i<n;i++)</pre>
39
            printf("%d %d %d\n",triangles[i].a,triangles[i].b,triangles[i].c);
40
41
42
        return 0;
43
```



Passed all tests! <



