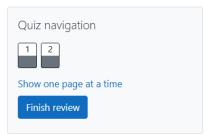
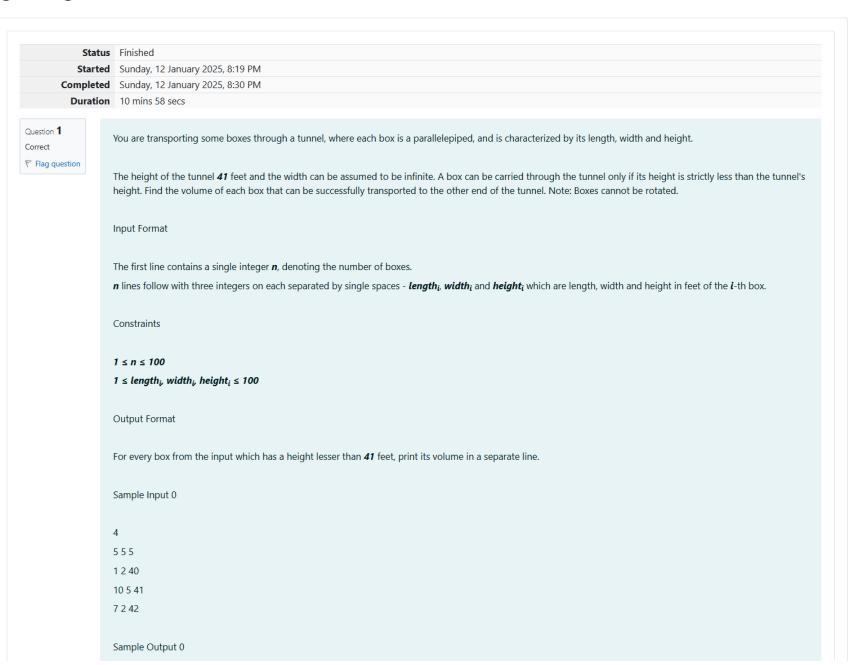
## GE23131-Programming Using C-2024





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  $1 \times 2 \times 4 = 80$ .

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

## Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 ₹ {
4
       int n;
       scanf("%d",&n);
5
6
       for(int i=0;i<n;i++)</pre>
7 ,
8
           int length, width, height;
           scanf("%d %d %d",&length,&width,&height);
9
10
11
           if(height<41)
12 ,
13
               int volume=length*width*height;
14
               printf("%d\n",volume);
15
16
17
18
```

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

Passed all tests! 🗸

▼ Flag question

guaranteed that all the areas are different.

The best way to calculate a volume of the triangle with sides  ${\it a}, {\it b}$  and  ${\it c}$  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

 $1 \le n \le 100$ 

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

 $a_i + b_i > c_i$ ,  $a_i + c_i > b_i$  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

5 12 13

3 4 5

Sample Output 0

3 4 5

5 12 13

7 24 25

Explanation 0

The square of the first triangle is 84. The square of the second triangle is 30. The square of the third triangle is 6. So the sorted order is the reverse one.

Answer: (penalty regime: 0 %)

```
2
   #include<math.h>
 3 #include<stdlib.h>
 4 typedef struct
5 ₹ {
 6
        double area;
7
        int a,b,c;
 8
    }Triangle;
9
10
    double calculate_area(int a,int b,int c)
11 • {
12
        double p=(a+b+c)/2.0;
13
        return sqrt(p*(p-a)*(p-c));
14
15
    int compare(const void*x,const void*y)
16 🔻 {
17
        Triangle *t1=(Triangle *)x;
        Triangle *t2=(Triangle *)y;
18
        if(t1->area<t2->area) return -1;
19
20
        if(t1->area>t2->area) return 1;
21
        return 0;
22
23
    int main()
24 ▼ {
25
        int n;
26
        scanf("%d",&n);
27
        Triangle triangles[n];
28
        for(int i=0;i<n;i++)</pre>
29 1
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
44
```

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

Passed all tests! <