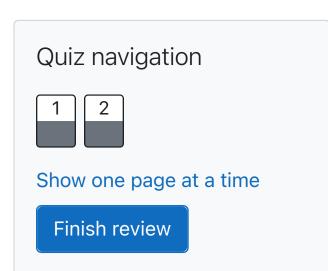
GE23131-Programming Using C-2024



Status Finished Started Monday, 13 January 2025, 11:21 AM Completed Monday, 13 January 2025, 11:34 AM **Duration** 12 mins 49 secs Question 1 You are transporting some boxes through a tunnel, where each box is a

Correct ▼ Flag question

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

n lines follow with three integers on each separated by single spaces - $length_i$,

The first line contains a single integer n, denoting the number of boxes.

width; and height; which are length, width and height in feet of the i-th box.

Constraints

 $1 \le n \le 100$

 $1 \le length_i$, 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

4

7 2 42

125

80

Explanation 0

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

about the fourth box.

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

1 #include <stdio.h> 2 v int main(){

Answer: (penalty regime: 0 %)

```
int n;
        scanf("%d",&n);
 4
        for(int i=0;i<n;i++){
            int length, width, height;
            scanf("%d %d %d",&length,&width,&height);
 8
9 •
            if(height<41){</pre>
                 int volume=length*width*height;
10
                 printf("%d\n",volume);
11
12
13
   }
14
```

Correct ▼ Flag question

Question **2**

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

You are given n triangles, specifically, their sides a_i , b_i and c_i . Print them in the

 $S = \ddot{O} p * (p - a) * (p - b) * (p - c)$ 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

Heron's formula:

and c_i on each separated by single spaces.

 $1 \le n \le 100$

Constraints

 $1 \le a_i$, b_i , $c_i \le 70$ $a_i + b_i > c_i$, $a_i + c_i > b_i$ 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

3 4 5 Sample Output 0

5 12 13

3 4 5 5 12 13

The square of the first triangle is **84**. The square of the second triangle is **30**.

Explanation 0

7 24 25

Answer: (penalty regime: 0 %) 1 #include <stdio.h>

The square of the third triangle is **6**. So the sorted order is the reverse one.

```
4 ▼ typedef struct {
5
       double area;
6
       int a,b,c;
```

#include <math.h> #include <stdlib.h>

```
}Triangle;
    double calculate_area(int a,int b,int c){
        double p=(a+b+c)/2.0;
10
        return sqrt(p*(p-a)*(p-b)*(p-c));
11
12
13
14 v int compare(const void*x,const void*y){
        Triangle *t1=(Triangle *)x;
15
16
        Triangle *t2=(Triangle *)y;
        if (t1->area < t2->area) return -1;
17
        if (t1->area > t2->area) return 1;
18
19
        return 0;
20
21
22 v int main(){
23
        int n;
        scanf("%d",&n);
24
        Triangle triangles[n];
25
26
27 •
        for (int i=0;i<n;i++){</pre>
28
            int a,b,c;
            scanf("%d %d %d",&a,&b,&c);
29
30
31
            triangles[i].a = a;
32
            triangles[i].b = b;
33
            triangles[i].c = c;
34
            triangles[i].area = calculate_area(a,b,c);
35
36
        qsort(triangles, n, sizeof(Triangle),compare);
37
38
39
        for(int i=0;i<n;i++){</pre>
            printf("%d %d %d\n",triangles[i].a, triangles[i].b
40
41
42
        return 0;
43
```

```
Expected Got
     Input
              3 4 5
                        3 4 5
     7 24 25 5 12 13
                        5 12 13
                        7 24 25
     5 12 13 7 24 25
     3 4 5
Passed all tests! <
```