Name: Kirithick Raja

Roll no.: 240701627

Week-14

Question 1:

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 - $\boldsymbol{length_i}$, $\boldsymbol{width_i}$ and $\boldsymbol{height_i}$ which are length, width and height in feet of the ***i***-th box.

Constraints

$\boldsymbol{1 \le n \le 100}$

$\boldsymbol{1 \le length_i, width_i, height_i \le 100}$

Output Format

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

Source code:

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
struct Box
{
    int length;
    int width;
    int height;
};
union Volume
{
    int value;
};
int main()
{
    int n;
    const int tunnel_height=41;
    scanf("%d",&n);
    struct Box boxes[n];
    union Volume volume;
    for(int i=0;i<n;i++)
    {
        scanf("%d %d %d",&boxes[i].length,&boxes[i].width,&boxes[i].height);
    }
    for(int i=0;i<n;i++)
    {
        if(boxes[i].height<tunnel_height)
        {
            volume.value=boxes[i].length*boxes[i].width*boxes[i].height;
            printf("%d\n",volume.value);
        }
    }
    return 0;
}
```

Result:

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

Passed all tests! ✓

Question 2:

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:

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

$1 \le n \le 100$

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

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

Source code:

```c
#include<stdio.h>
#include<math.h>
typedef struct
{
    int a,b,c;
    double area;
}
Triangle;
double calculate_area(int a,int b,int c)
{
    double p=(a+b+c)/2.0;
    return sqrt(p*(p-a)*(p-b)*(p-c));
}
void sort_triangles(Triangle triangles[],int n)
{
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n-i-1;j++)
        {
            if(triangles[j].area>triangles[j+1].area)
            {
                Triangle temp=triangles[j];
                triangles[j]=triangles[j+1];
                triangles[j+1]=temp;
            }
        }
    }
}
int main()
{
    int n;
    scanf("%d",&n);
    Triangle triangles[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d %d %d",&triangles[i].a,&triangles[i].b,&triangles[i].c);
        triangles[i].area=calculate_area(triangles[i].a,triangles[i].b,triangles[i].c);
    }
    sort_triangles(triangles,n);
    for(int i=0;i<n;i++)
    {
        printf("%d %d %d\n",triangles[i].a,triangles[i].b,triangles[i].c);
    }
    return 0;
}
```

Result:

	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! ✓