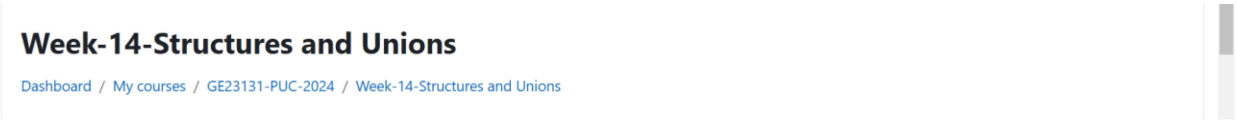
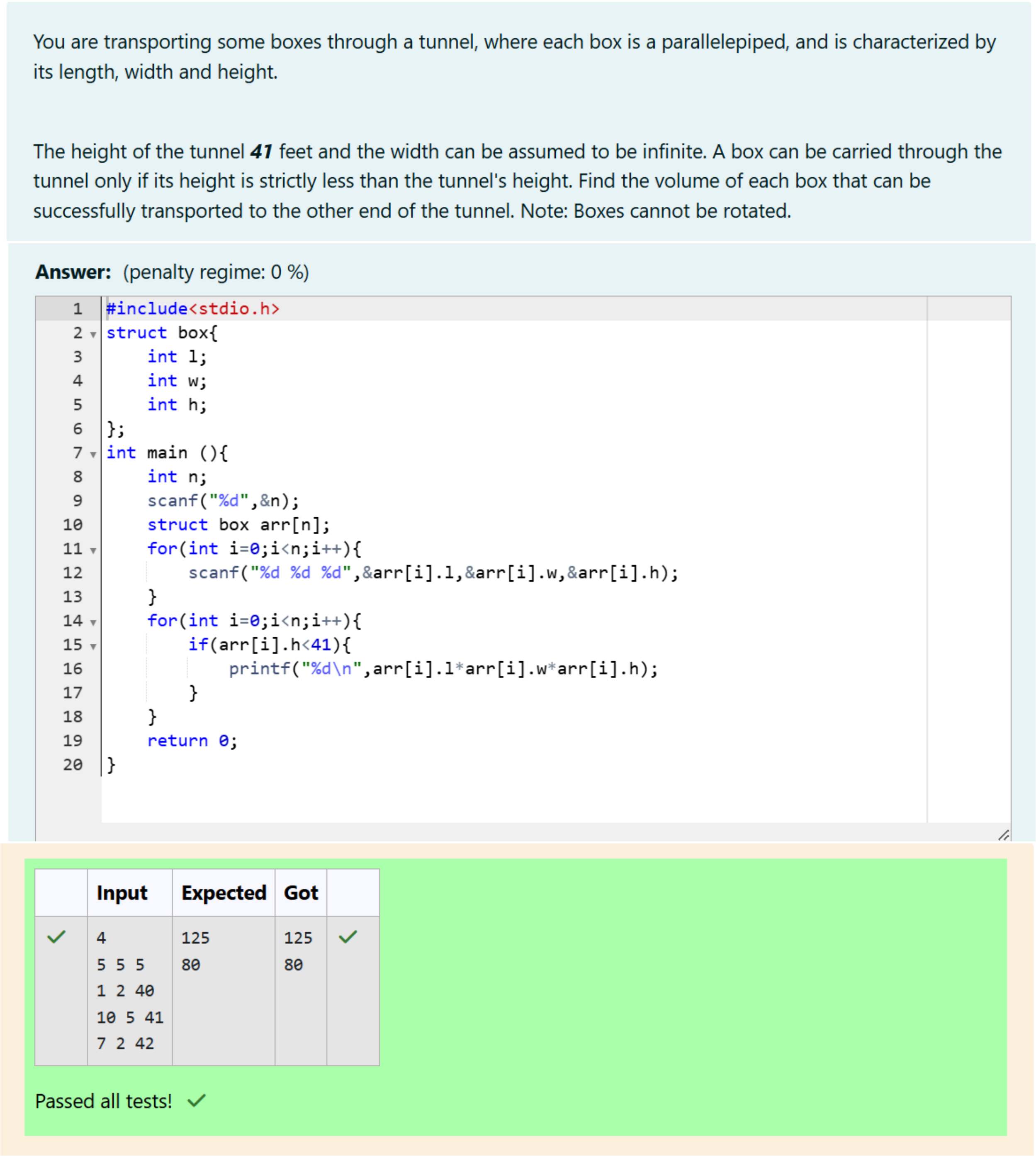
SANKARA GOMATHI.R.

240701470



Week-14-Structures and Unions

Dashboard / My courses / GE23131-PUC-2024 / 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.

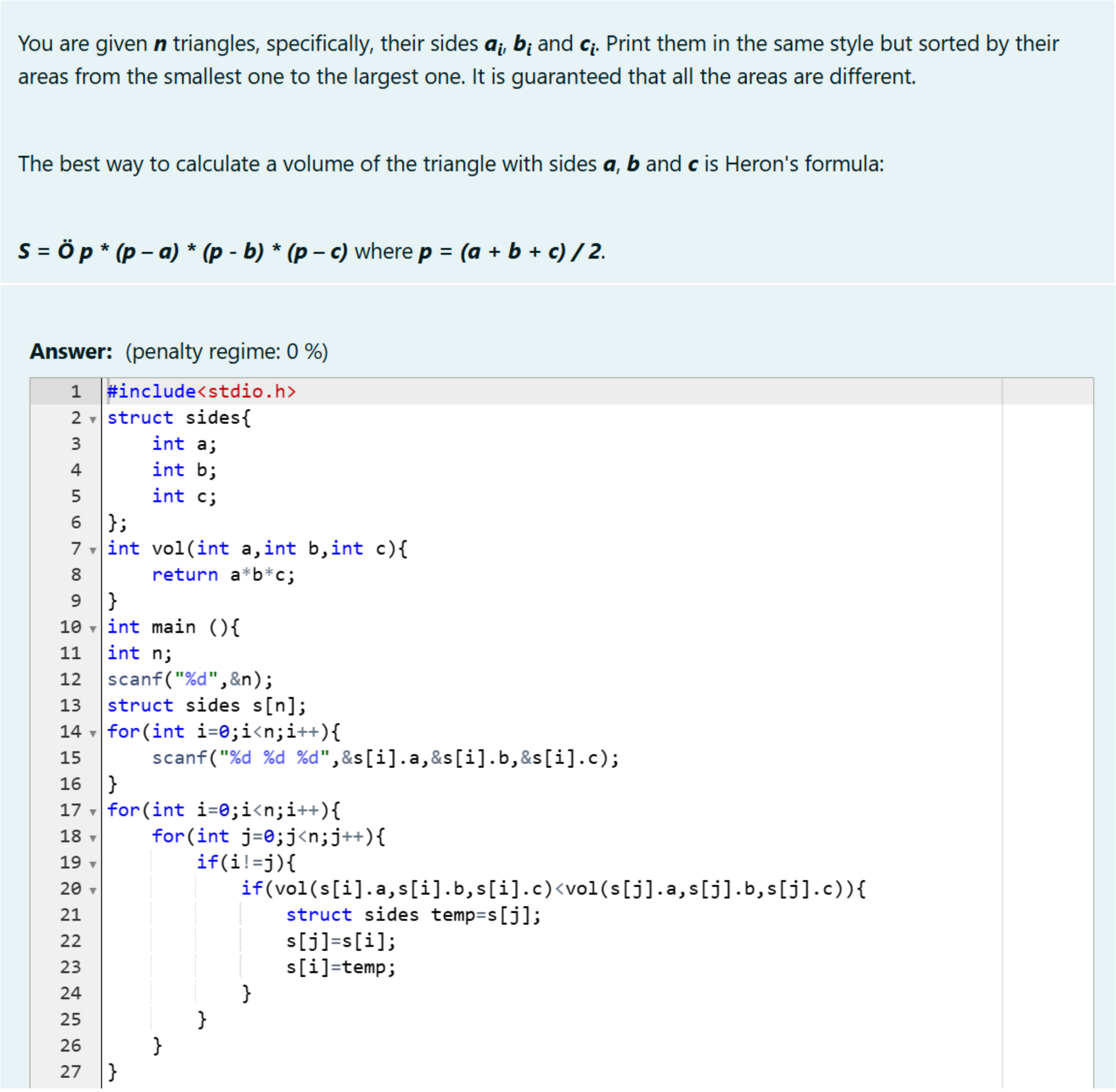
The height of the tunnel ***41*** feet and the width can be assumed to be infinite. A box can be carried through the tunnel only if its height is strictly less than the tunnel's height. Find the volume of each box that can be successfully transported to the other end of the tunnel. Note: Boxes cannot be rotated.

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
struct box{
    int l;
    int w;
    int h;
};
int main (){
    int n;
    scanf("%d",&n);
    struct box arr[n];
    for(int i=0;i<n;i++){
        scanf("%d %d %d",&arr[i].l,&arr[i].w,&arr[i].h);
    }
    for(int i=0;i<n;i++){
        if(arr[i].h<41){
            printf("%d\n",arr[i].l*arr[i].w*arr[i].h);
        }
    }
    return 0;
}
```

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

Passed all tests! ✓



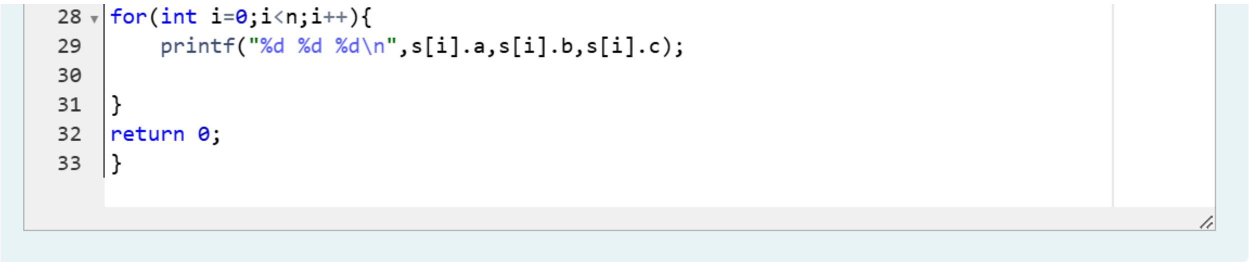
You are given n triangles, specifically, their sides a_i, b_i and c_i. Print them in the same style but sorted by their areas from the smallest one to the largest one. It is guaranteed that all the areas are different.

The best way to calculate a volume of the triangle with sides a, b and c is Heron's formula:

$S = \text{Ö}\ p * (p – a) * (p - b) * (p – c)$ where $p = (a + b + c) / 2$.

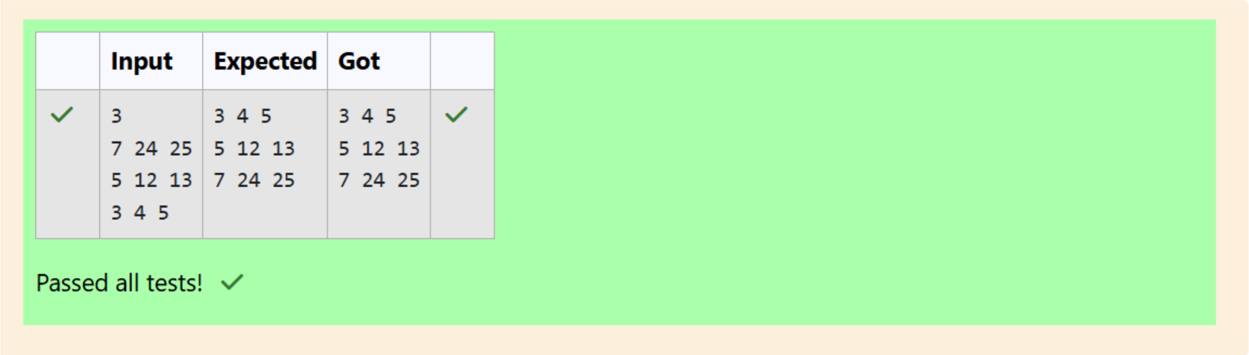
Answer: (penalty regime: 0 %)

```
#include<stdio.h>
struct sides{
    int a;
    int b;
    int c;
};
int vol(int a,int b,int c){
    return a*b*c;
}
int main (){
int n;
scanf("%d",&n);
struct sides s[n];
for(int i=0;i<n;i++){
    scanf("%d %d %d",&s[i].a,&s[i].b,&s[i].c);
}
for(int i=0;i<n;i++){
    for(int j=0;j<n;j++){
        if(i!=j){
            if(vol(s[i].a,s[i].b,s[i].c)<vol(s[j].a,s[j].b,s[j].c)){
                struct sides temp=s[j];
                s[j]=s[i];
                s[i]=temp;
            }
        }
    }
}
```



```
for(int i=0;i<n;i++){
    printf("%d %d %d\n",s[i].a,s[i].b,s[i].c);

}
return 0;
}
```



	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! ✓