RITTVIK S 2024-CSE 2116240701616

Week-14-Structures and Unions

Coding

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.

Source code

```c
#include <stdio.h>
int main()
{
    int n;
    scanf("%d", &n);
    for(int i=0;i<n;i++)
    {
        int len,width,height;
        scanf("%d %d %d",&len,&width,&height);
        if (height < 41)
        {
            int volume=len*width*height;
            printf("%d\n",volume);
        }
    }
}
```

Output

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

Passed all tests! ✓

Result

The above program is executed successfully and provides the above output.

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.

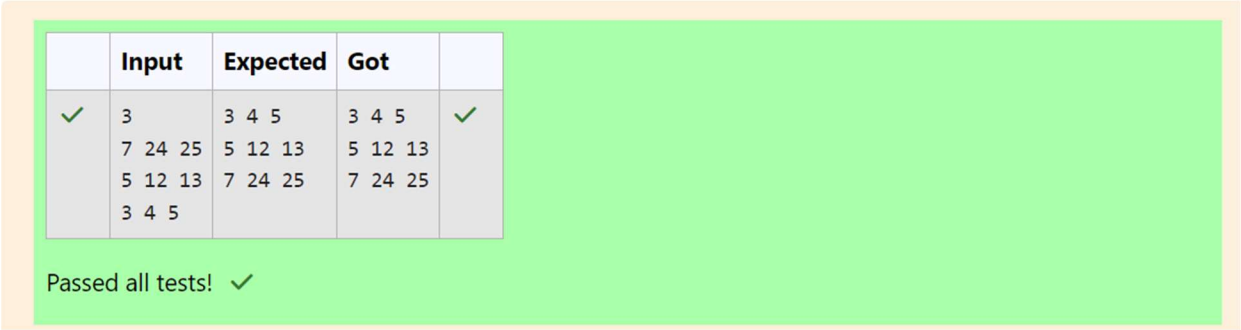
The best way to calculate a volume of the triangle with sides ***a***, ***b*** and ***c*** is Heron's formula:

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

Source code

```c
#include <stdio.h>
#include<math.h>
#include<stdlib.h>

typedef struct
{
    double area;
    int a,b,c;
}
tri;

double cal_area(int a,int b,int c)
{
    double p=(a+b+c)/2.0;
    return sqrt(p*(p-a)*(p-b)*(p-c));
}

int compare(const void*x,const void*y)
{
    tri*t1=(tri*)x;
    tri*t2=(tri*)y;
    if(t1->area < t2->area)
    return -1;
    if(t1->area > t2->area)
    return 1;
    return 0;
}
int main()
{
    int n;
    scanf("%d", &n);
    tri tris[n];
    for(int i=0;i<n;i++)
    {
        int a,b,c;
        scanf("%d %d %d", &a, &b, &c);
        tris[i].a = a;
        tris[i].b = b;
        tris[i].c = c;
        tris[i].area = cal_area(a,b,c);
    }
  qsort(tris,n,sizeof(tri),compare);
  for(int i=0;i<n;i++)
  {
        printf("%d %d %d\n",tris[i].a,tris[i].b,tris[i].c);
  }
  return 0;

}
```

Output



	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! ✓

Result

The above program is executed successfully and provides the above output.