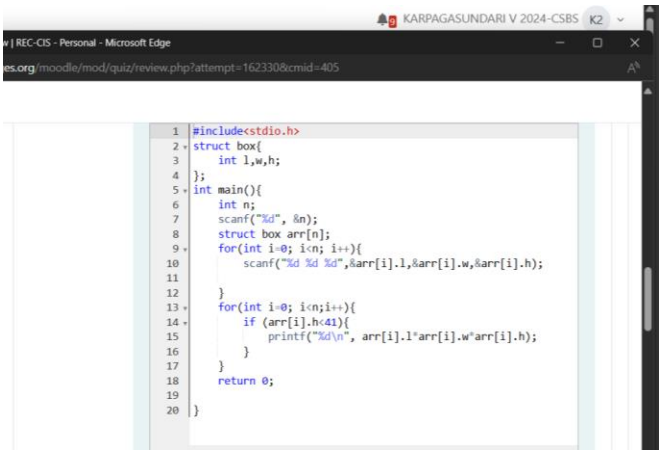
WEEK 14

1. You are transporting some boxes through a tunnel, where each box is a parallele piped, 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.

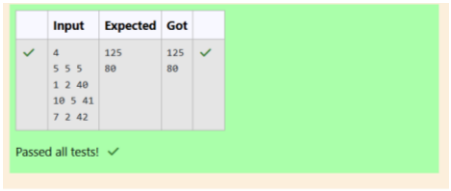
Input Format : The first line contains a single integer n, denoting the number of boxes. n lines follow with three integers on each separated by single spaces - lengthi, widthi and heighti which are length, width and height in feet of the i-th box.

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

Program:



Output:

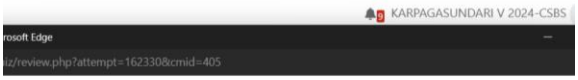


2. You are given n triangles, specifically, their sides ai, bi and ci. Print them in the same style but sorted by their areas from the smallest one to the largest one. It is guaranteed that all the areas are different. The best way to calculate a volume of the triangle with sides a, b and c is Heron's formula: p * (p – a) * (p - b) * (p – c) where p = (a + b + c) / 2.S =

Input Format: First line of each test file contains single integer n. n lines follow with ai, bi and ci on each separated by single spaces.

Output Format: Print exactly n lines. On each line print 3 integers separated by single spaces, which are ai, bi and ci of the corresponding triangle.

Program:




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

struct sides{
    int a,b,c;
};
int vol(int a, int b, int c){
    float p = (a+b+c)/2;
    return sqrt((p*(p-a)*(p-b)*(p-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=i; 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;
```

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! ✓