Week-14-Structures and Unions

Navigation

- Dashboard
  - Site home
  - Site pages
  - My courses
    - GE23131-PUC-2024
      - Participants
      - Competencies
      - Grades
      - General
      - Skill Test-01-MCQ & Coding
      - Lecture Notes
      - Week-01-Overview of C, Constants, Variables and Da...
      - Assessment-01-Overview of C, Constants, Variables ...
      - Week-02-Operators and Expressions, Managing

◄Assessment-13-Passing Arrays and Strings to Functions
►Week-15-Pointers

Week-14-Structures and Unions

Done

◄Assessment-13-Passing Arrays and Strings to Functions
Jump to...
►Week-15-Pointers

GE23131-Programming Using C-2024

Navigation

- Dashboard
  - Site home
  - Site pages
  - My courses
    - GE23131-PUC-2024
      - Participants
      - Competencies
      - Grades
      - General
      - Skill Test-01-MCQ & Coding
      - Lecture Notes
      - Week-01-Overview of C, Constants, Variables and Da...
      - Assessment-01-Overview of C, Constants, Variables ...
      - Week-02-Operators and Expressions, Managing Input ...
      - Assessment-02-Operators and Expressions, Managing

Attempts allowed: 4

This quiz has been configured so that students may only attempt it using the Safe Exam Browser.

Time limit: 1 hour 30 mins

Grading method: Highest grade

Your attempts

Attempt 1

Status	Finished
Started	Wednesday, 15 January 2025, 12:06 PM
Completed	Wednesday, 15 January 2025, 12:40 PM
Duration	33 mins 55 secs

Review

The Safe Exam Browser keys could not be validated. Check that you're using Safe Exam Browser with the correct configuration file.

Launch Safe Exam Browser | Download configuration

GE23131-Programming Using C-2024

Quiz navigation

1 2

Show one page at a time

Finish review

Status	Finished
Started	Wednesday, 15 January 2025, 12:06 PM
Completed	Wednesday, 15 January 2025, 12:40 PM
Duration	33 mins 55 secs

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

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

1 ≤ $length_i$, $width_i$, $height_i$ ≤ 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
5 5 5
1 2 40
10 5 41
7 2 42
```

Sample Output 0

```
125
80
```

Explanation 0

The first box is really low, only ***5*** feet tall, so it can pass through the tunnel and its volume is ***5 x 5 x 5 = 125***.

The second box is sufficiently low, its volume is ***1 x 2 x 4= = 80***

The first box is really low, only ***5*** feet tall, so it can pass through the tunnel and its volume is ***5 x 5 x 5 = 125***.

The second box is sufficiently low, its volume is ***1 x 2 x 4= = 80***.

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

Answer: (penalty regime: 0 %)

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

```
    for(int i=0; i<n;i++){

        if(boxes[i].height< 41){
            int volume =boxes[i].length*boxes[i].width*boxes[i].height;
            printf("%d \n",volume);
        }
    }
    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! ✓

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.

Sample Input 0

```
3
7 24 25
5 12 13
3 4 5
```

Sample Output 0

3 4 5

5 12 13

7 24 25

Explanation 0

The square of the first triangle is ***84***. The square of the second triangle is ***30***. The square of the third triangle is ***6***. So the sorted order is the reverse one.

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<math.h>
typedef struct{
    int a; int b; int c;
    double area;
}Triangle;
void swap(Triangle *a,Triangle*b){
    Triangle temp=*a;
    *a=*b;
    *b=temp;
}
void asced(Triangle arr[],int n){
    for(int i=0;i<n;i++){
        for(int j=i+1;j<n;j++){
            if(arr[i].area>arr[j].area){
                swap(&arr[i],&arr[j]);
            }
        }
    }
}
int main(){
    int n;
```

```
17              }
18          }
19      }
20  }
21  int main(){
22      int n;
23      scanf("%d",&n);
24      Triangle triangles[n];
25      for(int i=0;i<n;i++){
26          scanf("%d %d %d",&triangles[i].a,&triangles[i].b,&triangles[i].c);
27          double s=(triangles[i].a+triangles[i].b+triangles[i].c)/2.0;
28          triangles[i].area=sqrt(s*(s-triangles[i].a)*(s-triangles[i].b)*(s-triangles[i].c));
29      }
30      asced(triangles,n);
31      for(int i=0;i<n;i++){
32          printf("%d %d %d\n",triangles[i].a,triangles[i].b,triangles[i].c);
33      }
34      return 0;
35  }
36
37
38
```

	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! ✓

```
22      int n;
23      scanf("%d",&n);
24      Triangle triangles[n];
25      for(int i=0;i<n;i++){
26          scanf("%d %d %d",&triangles[i].a,&triangles[i].b,&triangles[i].c);
27          double s=(triangles[i].a+triangles[i].b+triangles[i].c)/2.0;
28          triangles[i].area=sqrt(s*(s-triangles[i].a)*(s-triangles[i].b)*(s-triangles[i].c));
29      }
30      asced(triangles,n);
31      for(int i=0;i<n;i++){
32          printf("%d %d %d\n",triangles[i].a,triangles[i].b,triangles[i].c);
33      }
34      return 0;
35  }
36
37
38
```

	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! ✓

Finish review