

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  $5 \times 5 \times 5 = 125$ .

The second box is sufficiently low, its volume is  $1 \times 2 \times 4 = 80$ .

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 %)

```
1 #include<stdio.h>
2 int main(){
3     int a;
4     scanf("%d",&a);
5     for (int i=0;i<a;i++){
6         int l,w,h;
7         scanf("%d %d %d",&l,&w,&h);
8         if(h<41){
9             printf("%d\n",(l*w*h));
10        }
11    }
12 }
```

	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:

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

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.

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

$$1 \leq n \leq 100$$

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

$$a_i + b_i > c_i, a_i + c_i > b_i \text{ and } b_i + c_i > a_i$$

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 %)

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<math.h>
4 int cmp(const void*a,const void* b){
5     return(*(int*)a)-*(int*)b);
6 }
7 int main(){
8     int a;
9     scanf("%d",&a);
10
11     int b[a][3];
12     int c[a],d[a];
13     for(int i=0;i<a;i++){
14         scanf("%d %d %d",&b[i][0],&b[i][1]
15
16         int p=(b[i][0]+b[i][1]+b[i][2]);
17         c[i]=sqrt((p*(p-b[i][0]))*(p-b[i][1]
18         d[i]=c[i];
19     }
20     qsort(d,a,sizeof(int),cmp);
21     for(int i=0;i<a;i++){
22         for(int j=0;j<a;j++){
23             if(d[i]==c[j]){
24                 printf("%d %d %d\n",b[j][0]
25                 break;
26             }
27         }
28     }
29 }
```

	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! ✓