

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 #define TUNNEL_HEIGHT 41
3 int main(){
4     int n;
5     scanf("%d",&n);
6     for(int i=0;i<n;i++){
7         int length,width,height;
8         scanf("%d %d %d",&length,&width,&height);
9         if(height <TUNNEL_HEIGHT)
10        {
11            int volume=length*width*height;
12            printf("%d\n",volume);
13        }
14    }
15    return 0;
16 }
```

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

Passed all tests! ✓

```

1 #include<stdio.h>
2 #include<math.h>
3 #include<stdlib.h>
4 double calculateArea(int a, int b,int c)
5 {
6     double p= (a+b+c)/2.0;
7     return sqrt(p*(p-a) * (p-b) * (p-c));
8 }
9 int compare(const void *t1, const void *t2)
10 {
11     int *triangless1=(int *)t1;
12     int *triangless2=(int *)t2;
13     double area1=calculateArea(triangless1[0],triangless1[1],triangless1[2]);
14     double area2=calculateArea(triangless2[0],triangless2[1],triangless2[2]);
15     if(area1<area2)
16     {
17         return -1;
18     }
19     if(area1>area2){
20         return 1;
21     }
22     return 0;
23 }
24 int main(){
25     int n;
26     scanf("%d",&n);
27     int triangles[n][3];
28     for(int i=0;i<n;i++)
29     {
30         scanf("%d %d %d",&triangles[i][0],&triangles[i][1],&triangles[i][2]);
31     }
32     qsort(triangles,n,sizeof(triangles[0]),compare);
33     for(int i=0;i<n;i++){
34         printf("%d %d %d\n",triangles[i][0],triangles[i][1],triangles[i][2]);
35     }
36     return 0;
37 }

```

```

19     if(area1>area2){
20         return 1;
21     }
22     return 0;
23 }
24 int main(){
25     int n;
26     scanf("%d",&n);
27     int triangles[n][3];
28     for(int i=0;i<n;i++)
29     {
30         scanf("%d %d %d",&triangles[i][0],&triangles[i][1],&triangles[i][2]);
31     }
32     qsort(triangles,n,sizeof(triangles[0]),compare);
33     for(int i=0;i<n;i++){
34         printf("%d %d %d\n",triangles[i][0],triangles[i][1],triangles[i][2]);
35     }
36     return 0;
37 }

```

	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! ✓