

✓ Assessment-12-Recursive Functions

✓ Week-13-Passing Arrays and Strings to Functions

☰ Week-13-Passing Arrays and Strings to Functions

✓ Done

✓ Assessment-13-Passing Arrays and Strings to Functions

✓ Week-14-Structures and Unions

☰ Week-14-Structures and Unions

✓ Done

✓ Week-15-Pointers

☰ Week-15 Pointers

✓ Done

Answer: (penalty regime: 0 %)

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

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

Passed all tests! ✓

Answer: (penalty regime: 0 %)

```

1 #include<stdio.h>
2 #include<math.h>
3 #include<stdlib.h>
4 typedef struct {
5     double ar;
6     int a,b,c;
7 }
8 }
9 Triangle;
10 double cal(int a,int b,int c)
11 {
12     double p=(a+b+c)/2.0;
13     return sqrt(p*(p-a)*(p-b)*(p-c));
14 }
15 int com(const void*x,const void*y)
16 {
17     Triangle* t1=(Triangle*)x;
18     Triangle* t2=(Triangle*)y;
19     if(t1->ar < t2->ar) return -1;
20     if(t1->ar > t2->ar) return 1;
21     return 0;
22 }
23 int main()
24 {
25     int n;
26     scanf("%d",&n);
27     Triangle tri[n];
28     for(int i=0;i<n;i++)
29     {
30         int a,b,c;
31         scanf("%d %d %d",&a,&b,&c);
32         tri[i].a=a;
33         tri[i].b=b;
34         tri[i].c=c;
35         tri[i].ar=cal(a,b,c);
36     }
37 }
```

```
28         for(int i=0;i<n;i++)
29     {
30         int a,b,c;
31         scanf("%d %d %d",&a,&b,&c);
32         tri[i].a=a;
33         tri[i].b=b;
34         tri[i].c=c;
35         tri[i].ar=cal(a,b,c);
36     }
37     qsort(tri,n,sizeof(triangle),com);
38     for(int i=0;i<n;i++)
39     {
40         printf("%d %d %d\n",tri[i].a,tri[i].b,tri[i].c);
41     }
42 }
43 }
```

Input	Expected	Got
3 7 24 25 5 12 13 5 12 13 3 4 5	3 4 5 5 12 13 7 24 25 3 4 5	3 4 5 5 12 13 7 24 25 3 4 5

Passed all tests! ✓

Finish review

