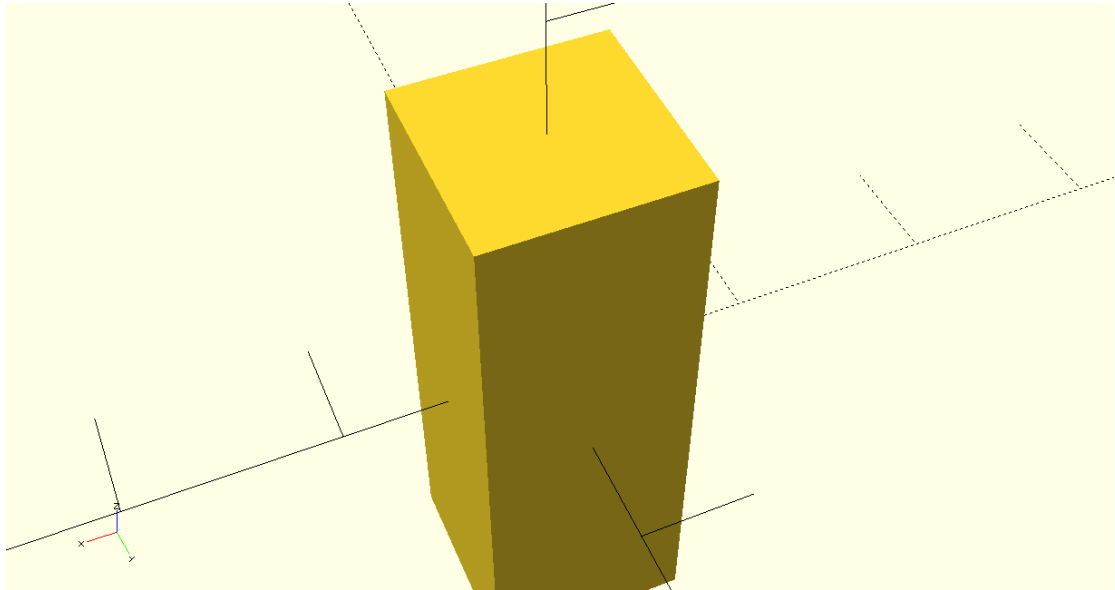```
1 $fn = 50;
2
3 cube([10,10,29], center = true);
4
```



I am responsible for the modeling part, first of all the first model I made a wooden cabinet, this is very simple to set a good rectangular length, width and height on it.

```
1   $fn = 50;
2
3   difference(){
4   minkowski(){
5   cube([10,20,10], center = true);
6   sphere(1);
7   }
8
9   // chop off the top
10
11 translate([0,0,5])cube([13,23,10], center = true);
12
13
14 // Hollow inside
15
16 minkowski(){
17   cube([8,18,8], center = true);
18   sphere(1);
19 }
20 }
```
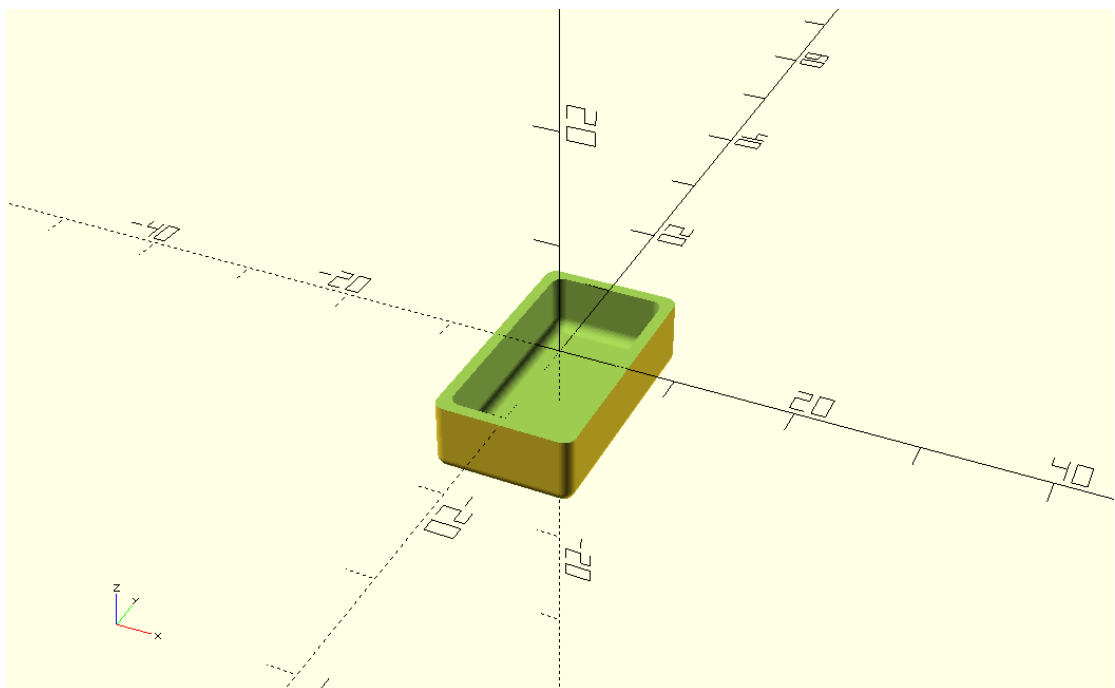
```
 1 side_length = 40;
 2 angle = 36; // angle between the sides of the
        rhombus in degrees
 3
 4 height = side_length * sin(angle);
 5 width = side_length * cos(angle);
 6 thickness = 25; // The thickness of the 3D
        shape (5 times the previous thickness)
 7
 8 module rhombus() {
 9   polygon(points = [
10   [0, 0],
11   [width, height],
12   [2 * width, 0],
13   [width, -height]
14   ]);
15 }
16
17 module star() {
18   // Draw the first rhombus
19   rhombus();
20
21   // Rotate and draw the other four rhombuses
22   for (i = [1:4]) {
23   rotate([0, 0, i * 72]) {
24   rhombus();
25   }
26   }
27 }
28
29 // Extrude the 2D shape to create a 3D object
30 linear_extrude(height = thickness) {
31   star();
32 }
33
```

The second one is a pallet. First I drew a rectangle, then I set another rectangle to wrap the top half of the first rectangle, then I removed the top half of the first rectangle with the difference command, then I copied the code of the first rectangle and changed the length, width and height so that it was wrapped by the rest of the first rectangle and then removed it again to get a pallet with a hollow middle.
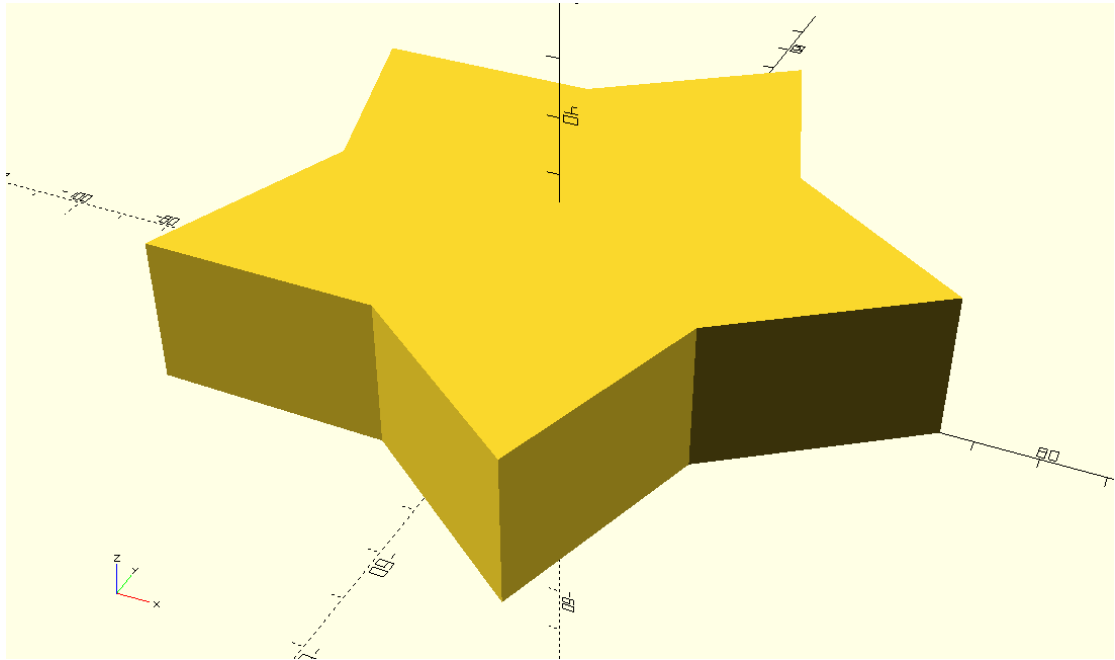
In the third part I create a 3D star-shaped object. The code first defines the rhombus, and then combines the five rhombuses into a star. Finally, the 2D star is stretched along the Z-axis to create a 3D star object.

```
1   overlap = 0.1; // desired overlap between cylinders
2   num_cylinders = 4; // number of cylinders
3   s = 10; // length of square side
4
5   r = s/6; // radius of cylinders
6   h = s; // height of square and cylinders
7
8   total_cylinder_length = r * 2 * num_cylinders + overlap * (num_cylinders - 1);
9   spacing = (s - total_cylinder_length) / 2;
10
11  translate([-s/2 + r/2 + overlap/2 - r - overlap + (r + overlap) / 2, -5, 0]) cube([s, s, h]);
12
13  for (i=[0:num_cylinders-1]) {
14      translate([-(total_cylinder_length/2) + r + i * (2 * r + overlap), -5, 0]) cylinder(h=h, r=r, $fn=16);
15  }
16
17  // Second side
18  translate([0, s/2 - r/2 - overlap/2 + r + overlap - (r + overlap) / 2, 0]) rotate([0, 0, 90]);
19
20  for (i=[0:num_cylinders-1]) {
21      translate([5, -(total_cylinder_length/2) + r + i * (2 * r + overlap), 0]) rotate([0, 0, 90]) cylinder(h=h, r=r, $fn=16);
22  }
23
24  // Third side
25  translate([s/2 - r/2 - overlap/2 + r + overlap - (r + overlap) / 2, 0, 0]) rotate([0, 0, 180]);
26
27  for (i=[0:num_cylinders-1]) {
28      translate([-(total_cylinder_length/2) + r + i * (2 * r + overlap), 5, 0]) rotate([0, 0, 180]) cylinder(h=h, r=r, $fn=16);
29  }
30
31  // Fourth side
32  translate([0, -s/2 + r/2 + overlap/2 - r - overlap + (r + overlap) / 2, 0]) rotate([0, 0, -90]);
33
34  for (i=[0:num_cylinders-1]) {
35      translate([-5, -(total_cylinder_length/2) + r + i * (2 * r + overlap), 0]) rotate([0, 0, -90]) cylinder(h=h, r=r, $fn=16);
36  }
```
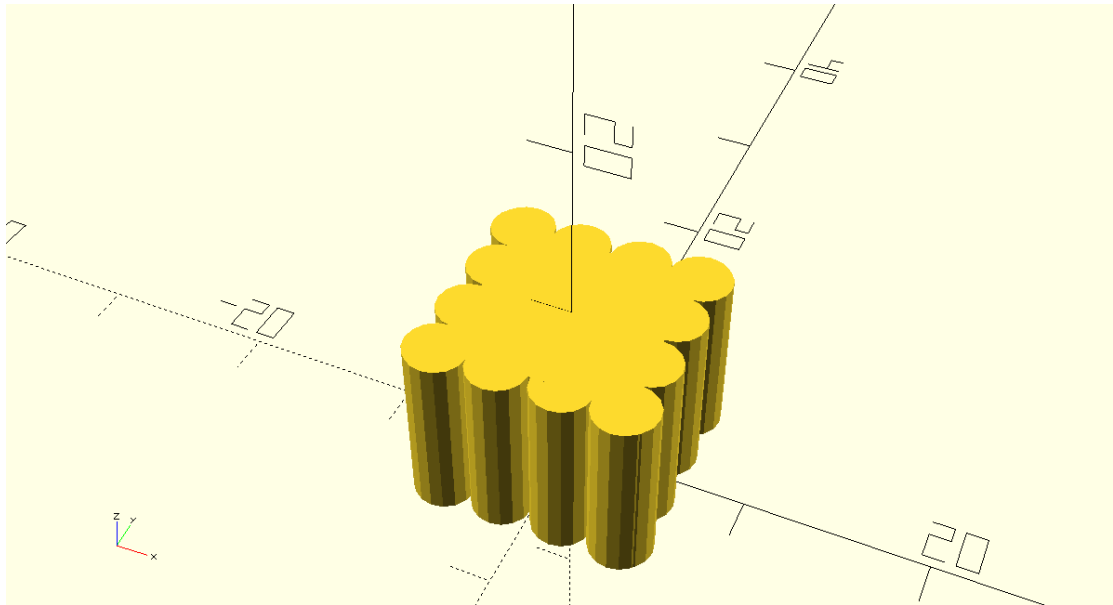
The fourth part first I first defined some variables such as overlap, number of cylinders, length of square sides, etc. Then the radius of the cylinder, the height and the spacing between the cylinders are calculated.

Next, I created a square in code and placed it at the appropriate location near the origin. Then, a series of cylinders are created on the four sides of each square. The cylinders are placed in the correct position by using a combination of loops and transformations (translate and rotate).