Solutions: Test 2

1 Problem 1

No, they don't join smoothly. This is because the vector joining the last two control points of the first curve is $[1, 2](= [3 - 2, 3 - 1])$ and the vector joining the first two control points of the second curve is $[1, 1](= [4 - 3, 4 - 3])$. Those vectors would have to be parallel and pointing in the same direction in order for the two curves to join smoothly. And they are not.

To make the two curves join smoothly, one can move $(3, 3)$ to any point on the interval that joins $(2, 1)$ and $(4, 4)$. For example, the midpoint, $(3, 2.5)$ would do.

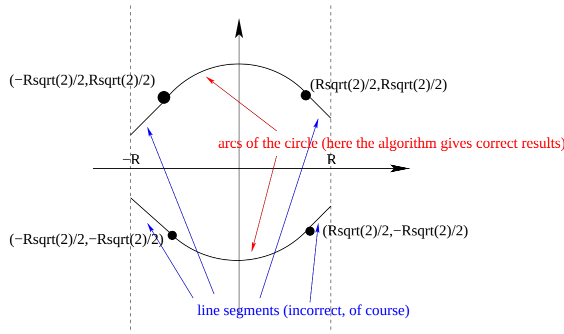
2 Problem 2

Here is one possibility.

```
draw(object1);
PushMatrix();
MulMatrix(T1);
draw(object2);
PushMatrix();
MulMatrix(T3);
draw(object3);
PopMatrix();
MulMatrix(T4);
draw(object5);
PopMatrix();
MulMatrix(T2);
draw(object4);
```

3 Problem 3

Basically, the algorithm will not be able to keep up with the slope of the circle for x values beyond $\frac{R\sqrt{2}}{2}$, and will draw a line (with slope -1 in the first quadrant and its symmetric image for other quadrants), leading to something like this:



4 Problem 4

Essentially, the problem with a cube is that it has 'too few' normals. For any direction, one can find a point on the sphere where the normal is pointing in that direction. Because of that, an image of a sphere contains colors comming towards the sphere from every direction (needed for environment mapping). This is *not* true for the cube: it has just six normals (one per face). There are lots of directions none of the normals is pointing to.

5 Problem 5

We have the following relationships, if T_n, E_n and V_n are numbers of vertices after n subdivision steps:

$$T_0 = V_0 = 4, E_0 = 6$$

and

$$T_{n+1} = 4T_n, V_{n+1} = E_n + V_n, E_n = 3/2T_n.$$

Therefore:

$$T_1 = 4T_0 = 16, E_1 = 3/2 * 16 = 24, V_1 = 6 + 4 = 10,$$

$$T_2 = 4T_1 = 64, E_2 = 3/2 * 64 = 96, V_2 = 10 + 24 = 34,$$

$$T_3 = 4T_2 = 256, E_3 = 3/2 * 256 = 384, V_3 = 34 + 96 = 130.$$

For N subdivision steps, the number of triangles is $T_N = 4^{N+1}$ and edges - $E_N = 3/2 * T_N = 6 * 4^N$. And the number of vertices is

$$V_N = E_{N-1} + V_{N-1} = 6 * 4^{N-1} + V_{N-1}.$$

This is a recursive equation that leads to

$$V_N = 6 * (4^{N-1} + 4^{N-2} + \ldots + 1) + V_0 = 2 * (4^N + 1)$$