1. 30ms may not sound like a lot of time, but a fast moving object can move a long distance in that time. So, if I reject a step when I collide, I may be pretty far from the object rather than right at it. Describe an algorithm to reliably get very close (within 1 pixel, for example) to the colliding object without actually touching it. (back)

```c
while(delta > 0)
{
    // Try to step the whole amount
    step = delta;
    do
    {
        badcollision = false;
        save_state();
        dist = advance(step);
        if(collision())
        {
            if(dist > 1)
            {
                // Bad collision
                restore_state();
                step = step / 2;
                badcollision = true;
            }
            else
            {
                // Good collision
                restore_state();
                delta = 0;
            }
        }
        else
        {
            // No collision, good step
            delta -= step;
        }
    }
    else
    {
        // No collision, good step
        delta -= step;
    }
}
2. How could you make more realistic steering? What variables would you need and what will the Euler step be?

Don't need rotational velocity.
Do need steering, a scalar.

How much we change the angle is dependent on how much we move, which is dependent on the velocity.

rotation += velocity * delta * steering;

3. For the figures, what lines, if any, can serve as separator lines?

(0, 0) to (4, 4)
(2, 0) to (8, 4)

4. Assuming the function in slide 11 is called with TestLR(A, B), what are the values of a, b, and c for the first iteration of the inner loop? TestLR tests vertices in A against edges in B.

(2, 0) to (8, 4)

\[ a = y_1 - y_2 = 0 - 4 = -4 \]
\[ b = x_2 - x_1 = 8 - 2 = 6 \]
\[ c = -a x_1 - b y_1 = 4 (2) - 6 (0) = 8 \]
\[ -4x + 6y + 8 = 0 \]

5. If you test vertex (4,4), is it on the good side or the bad side?

\[ r = -4 (4) + 6 (4) + 8 = -16 +24 + 8 = 16 \]
Yes, it is on the good side

6. What are the bounding boxes for the two polygons A and B?

A: (0, 0) to (4, 4)
B: (2, 0) to (8, 4)