1. An axis-aligned bounding box (AABB) in 2D is defined by four scalar values:
\[ (\text{min}X, \text{max}X, \text{min}Y, \text{max}Y) \]
We use \( (1, -1, 1, -1) \) to denote the empty AABB. Let
\[ bb_1 = (\text{min}X_1, \text{max}X_1, \text{min}Y_1, \text{max}Y_1) \]
and
\[ bb_2 = (\text{min}X_2, \text{max}X_2, \text{min}Y_2, \text{max}Y_2) \]
be two non-empty AABBs.

(a) What is the minimum AABB that contains the union of \( bb_1 \) and \( bb_2 \)?
(b) What is the minimum AABB that contains the intersection of \( bb_1 \) and \( bb_2 \)?
(c) What is the minimum AABB that contains the difference of \( bb_1 \) and \( bb_2 \) (i.e., \( bb_1 \setminus bb_2 \)).?

2. One way to make LOD transitions is to use an \( \alpha \) fade, where you lerp the \( \alpha \) channel to blend the two LODs. Assume that you have a triangle \( (p_1, p_2, p_3) \) and a vertex \( q \) that bisects the line \( p_2p_3 \), splitting the triangle into two triangles \( (p_1, p_2, q) \) and \( (p_1, q, p_3) \).

Assuming that triangles have the following representation:

```c
typedef struct {
    Vec3f_t verts[3];
    Vec3f_t normal;
    Vec3f_t color;
} Triangle_t;
```

Define a function

```c
void alphaLerp (Triangle_t *tri, Vec3f_t q, float t);
```

that takes as arguments the triangle \( \tri \), the point of bisection \( q \), and a parameter \( 0 \leq t \leq 1 \) that controls the blending of the two images. When \( t \) is 0, just the single triangle should be drawn, and when \( t \) is 1, just the triangle pair should be drawn. You may use mathematical notation or C code to write your answer, but it should clearly specify the OpenGL state used in rendering.