1. (15 pts)
   (a) (6 pts)\[ T = \begin{pmatrix}
   1 & 0 & 0 & 5 \\
   0 & 1 & 0 & 2 \\
   0 & 0 & 1 & -1 \\
   0 & 0 & 0 & 1
   \end{pmatrix}. \]
   (b) (6 pts) First, compute \( \sin 45^\circ = \frac{1}{\sqrt{2}} \) and \( \cos 45^\circ = \frac{1}{\sqrt{2}} \). Then
   \[ R = \begin{pmatrix}
   1 & 0 & 0 & 0 \\
   0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\
   0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\
   0 & 0 & 0 & 1
   \end{pmatrix}. \]
   (c) (3 pts) The correct order is \( RT \). That is because we multiply with the matrix on the left and the point on the right. Let \( P \) be the point. Then \( (RT)P = R(TP) \). That is, \( P \) get translated, and then the result of the translation gets rotated.

2. (18 pts)
   (a) (10 pts) Compute \( n \) first:
   \[ n = \frac{E - O}{|E - O|} = \frac{(3, 4, 0)}{5} = \left( \frac{3}{5}, \frac{4}{5}, 0 \right). \]
   Then compute \( u \):
   \[ u = \frac{up \times n}{|up \times n|} = \frac{(0, 0, 1)}{1} = (0, 0, 1). \]
   Then compute \[ v = n \times u = \left( \frac{4}{5}, -\frac{3}{5}, 0 \right). \]
   (b) (8 pts) The upper-left 3\( \times \)3 submatrix consists of the elements of \( u, v, \) and \( n \). The fourth column consists of \( -e \cdot u = 0, -e \cdot v = 0, \) and \( -e \cdot n = -5 \), where \( e = E - O = (3, 4, 5) \). So the view matrix is
   \[ V = \begin{pmatrix}
   0 & 0 & 1 & 0 \\
   \frac{4}{5} & -\frac{3}{5} & 0 & 0 \\
   -\frac{3}{5} & \frac{4}{5} & 0 & -5 \\
   0 & 0 & 0 & 1
   \end{pmatrix}. \]
3. (6 pts) The eye is at the origin in the eye coordinate system, so its coordinates are (0, 0, 0). The look point is somewhere along the negative z-axis. It is not necessarily at (0, 0, −1). It depends on its distance from the eye point.

4. (18 pts)
   (a) (6 pts) In the function call, we see that \( l = -2, r = 2, b = -1, t = 1, \) \( n = 0.1, \) and \( f = 10. \) So the projection matrix is
   \[
   P = \begin{pmatrix}
   0.05 & 0 & 0 & 0 \\
   0 & 0.1 & 0 & 0 \\
   0 & 0 & -10.1/9.9 & -2/9.9 \\
   0 & 0 & -1 & 0 \\
   \end{pmatrix}.
   \]
   (b) (6 pts) Multiply:
   \[
   \begin{pmatrix}
   0.05 & 0 & 0 & 0 \\
   0 & 0.1 & 0 & 0 \\
   0 & 0 & -10.1/9.9 & -2/9.9 \\
   0 & 0 & -1 & 0 \\
   \end{pmatrix}
   \begin{pmatrix}
   2 \\
   1 \\
   -1 \\
   1 \\
   \end{pmatrix}
   = \begin{pmatrix}
   0.1 \\
   0.1 \\
   8.1/9.9 \\
   1 \\
   \end{pmatrix}.
   \]
   (c) (6 pts) We see that the \( x, y, \) and \( z \) coordinates of the result are all between \(-w\) and \( w\) \((w = 1)\), so the point should not be clipped.

5. (12 pts)
   (a) (6 pts) It is not done because the clipping calculations are much more complicated in the irregularly shaped view frustum. Once the frustum is regularized in clip coordinates, the clipping can be done much faster, so much faster that it is worth it to perform the lighting calculations on vertices that will later be clipped.
   (b) (6 pts) The transformation to clip coordinates is not an isometry. That is, it does not preserve angles. Thus, the lighting calculations at that stage would be based on the wrong angles between the vectors.

6. (6 pts) The benefit occurs when the texture is magnified and linear interpolation is used. The \( 2 \times 2 \) texture would show severe blurring, while the blurring would not be so severe in the \( 4 \times 4 \) texture. Under minification, it is not clear that there is an advantage. With the \( 4 \times 4 \) texture, it is more likely that the 4 nearest texels would all be the same color, but that might be a disadvantage, creating the annoying Moire patterns.

7. (10 pts) We have \( H = 3 \) and \( W = 7 \), so compute \( 2H = 6, 2(W - H) = 8, \) and \( 3H - 2W = -5. \)
8. (9 pts)

9. (6 pts) The framebuffer includes the depth buffer, which stores the depth of the current pixel. When a new fragment arrives at the framebuffer, its depth is compared to the depth stored in the depth buffer. If it is less (closer), then it replaces the previous pixel. If it is greater, then it is discarded. Thus, in the end, the closer objects remain in the framebuffer.