1. (6 points) Briefly explain how a color lookup table works and why it is used.

2. (6 points) Briefly explain how double buffering works and why it is used.

3. (6 points) We are given the following bit patterns
   
   $A = \begin{array}{c}
   11000111 \\
   10101010
   \end{array}$

   Give the resulting patterns when the raster operation xor is applied:
   
   a. $C = (A \ xor \ B)$
   b. $D = (C \ xor \ A)$

4. (6 points) The even/odd parity rule can be used with overlapping polygons. Use the even/odd parity rule to determine which of the points is “inside” and which is “outside” for these overlapping polygons:

![Overlapping Polygons](image)

5. Given the following binary region code assignment:
   
   0001 left
   0010 right
   0100 bottom
   1000 top

   and a unit square clipping window $x_{w_{min}} = 0, y_{w_{min}} = 0, x_{w_{max}} = 1,$ and $y_{w_{max}} = 1.$

   a. (6 points) Determine the region codes used in the Cohen-Sutherland clipping algorithm for the following points:
      
      a. (2.6,0.1)
      b. (0.1,0.1)
      c. (-0.1,-0.1)

   (b) (4 points) Can a line segment with codes (1010, 0010) be trivially accepted, trivially rejected, or neither? Briefly justify your answer.

   (c) (4 points) Is 0111 a valid vertex code for the Cohen-Sutherland algorithm? Briefly justify your answer.

   (d) (6 points) What needs to be done to extend the 2-D Cohen-Sutherland line clipping algorithm to 3-D? What is the maximum number of times that a 3-D line might be clipped by this algorithm? Be sure to briefly justify your answers.
6. We would like to adapt Bresenham’s line drawing algorithm to draw dashed lines, where the dash pattern is specified by a pixel mask. For example, the mask 1111000 produces a dashed line, 5 pixels “on” and then 3 pixels “off” ... and so on.

(a) (5 points) Why might the apparent length of dashes along lines vary, depending on the slope? By this I mean, if we rotate a line then the dashes will appear to get longer and shorter depending on the line orientation (slope). Give a brief mathematical justification of your answer.

(b) (5 points) How would you modify Bresenham’s line drawing algorithm to correct this problem?

7. (6 points) Identify the bug in this flood fill function that is intended to fill 4-connected neighbors. Briefly describe how you would fix this bug.

```c
void floodFill4 (int x, int y, int fillColor, int interiorColor) {
    int currentColor;
    
    currentColor = getPixel (x, y);
    if (currentColor == interiorColor){
        /* set current pixel to fill color */
        setPixel (x, y, fillColor);
        /* Visit the pixel’s 4-connected neighbors */
        floodFill4 (x+1, y, fillColor, interiorColor);
        floodFill4 (x-1, y, fillColor, interiorColor);
        floodFill4 (x, y+1, fillColor, interiorColor);
        floodFill4 (x, y-1, fillColor, interiorColor);
    }
}
```

8. (10 points) We are given the unit quaternion \( q = (s, \mathbf{v}) \) that describes a rotation about an arbitrary axis that passes through the origin. Given only \( q \), derive a mathematical expression for 3D axis of rotation (be sure to give a unit vector), and the rotation angle.

9. (10 points) Assume homogeneous transform matrices, where

\[
\begin{align*}
T(t_x, t_y, t_z) & \quad \text{gives general 3D translation} \\
S(s_x, s_y, s_z) & \quad \text{gives uniform scaling, i.e. } s_x = s_y = s_z \\
R(\theta_x, \theta_y, \theta_z) & \quad \text{gives general 3D rotation}
\end{align*}
\]

Given the above definitions, which of the following 3D graphics transformations commute? Briefly justify your answers.

a. \( TS \)
b. \( SR \)
c. \( S_1S_2 \)
d. \( R_1R_2 \)
e. \( T_1T_2 \)
10. (10 points) Write a 2D homogeneous transform matrix $M$ that when applied to a point yields the relations:

$$x' = x + ay$$
$$y' = -y$$

In words, what two basic computer graphics transforms happen when we apply $M$ to a point?

11. (10 points) Give a brief example of how hierarchical bounding boxes can be used to make clipping of complicated scenes more efficient. Feel free to draw pictures to illustrate your point.

12. (15 points CS680 required, 10 points CS480 extra credit) We are given the triangle with vertices

$\mathbf{P}_1 = (1, 2)$
$\mathbf{P}_2 = (4, 2)$
$\mathbf{P}_3 = (1, 6)$

We are also given $(r, g, b)$ colors (in the range $0 : 255$) at the three vertices

$\mathbf{C}_1 = (200, 200, 0)$
$\mathbf{C}_2 = (0, 100, 200)$
$\mathbf{C}_3 = (200, 0, 100)$

Use bilinear interpolation to obtain the color at a point inside the triangle $\mathbf{Q} = (2, 3)$. 