CSC 315 - FINAL EXAM
Dr. A.J. Pounds
Fall 2007 IDP

Name ________________________________ **Section** ____

This test is administered under the auspices of the Mercer University Honor Code.

Some potentially useful mathematical entities...

$$T\left(d_x, d_y, d_z\right) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S\left(s_x, s_y, s_z\right) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_x\left(\theta\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y\left(\theta\right) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z\left(\theta\right) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Show all work and draw a box around numerical answers.

1. What is **scissoring**? *(5 pts)*

2. What is **antialiasing**? Why is it needed? *(Hint: a picture could help.)* *(5 pts)*

3. What is the **frame buffer**? *(5 pts)*

4. Why are **homogenous coordinates**? How are they beneficial when it comes to two and three dimensional transformations? *(5 pts)*

5. What is the **Z-Buffer**? What is it used for and how does it work? *(5 pts)*

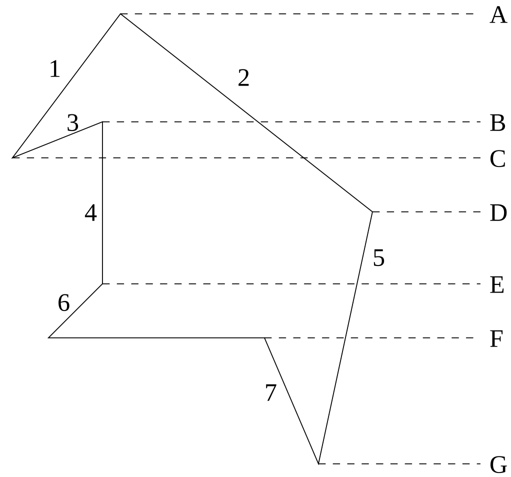
6. What are **display lists**? How are they used in graphics? Where should they be placed for optimal graphics performance? *(5 pts)*

7. What is a **frustrum**? How is it used in graphics programming? How are the VUP, VPN, VRP, u, v, and n vectors defined in the frustrum? *(Hint: draw a figure and label it.)* *(15 pts)*

8. One computes the position of a pixel on a $45°$ arc of a circle (but not at the position of $0°$ or $45°$).

   (a.) How many other pixels can be drawn with no computation? *(10 pts)*

   (b.) If the circle is centered at $(0, 0)$, and the first pixel is drawn at (x, y), what are the other pixel positions that can be drawn without computation? *(10 pts)*



9. The polygon above is made of line segments one through seven. Why is there no need to consider the line segment connecting segment six and seven when filling the polygon with the parity filling algorithm? *(4 pts)*

10. When filling the polygon above with the parity filling algorithm it is necessary to keep the polygon line segments in three separate tables: the waiting table, the active element table, and the deleted element table. For enhanced performance, it is also necessary to modify the loop indices so that line segments in the deleted and waiting tables are minimally considered during the fill operation. For each blank below, list the elements in the waiting, active “$<$ $>$” , and deleted “( )” tables. Use **D** as your guide. *(6 pts)*

    A:

    B:

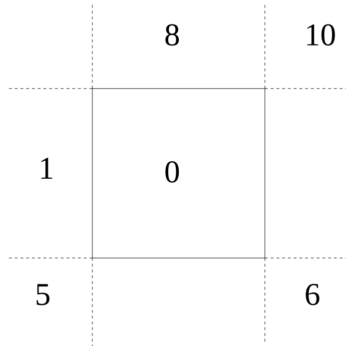
    C:

    D: (1) (2) (3) $<$ 4 5 $>$ 6 7

    E:

    F:

    G:

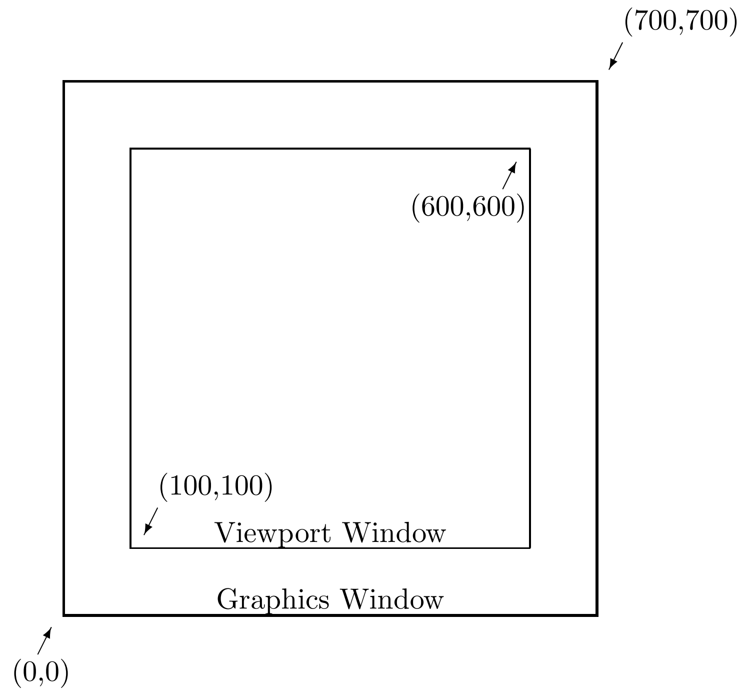


11. The diagram above shows the region outcodes (decimal) for the Cohen-Sutherland line clipping algorithm.

    (a.) Complete the diagram by placing the correct decimal numbers in the empty regions. *(15 pts)*

    (b.) List the **two** conditions that must be met for a line segment to **not** be trivially rejected. *(10 pts)*

    (c.) The 2D version of the Cohen-Sutherland algorithm uses 4 bit binary numbers to represent the outcodes. What is the minimum length of the binary number (in bits) needed to represent the outcodes for the 3D version of the Cohen-Sutherland algorithm? *(5 pts)*

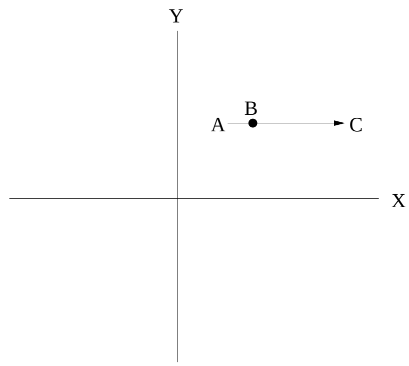


12. A function with domain [-100,50] and range [150,200] is to be mapped into the viewport shown above so that the function completely fills the viewport.

    (a.) Provide mathematical expressions which will map the function point (x, y) into pixel points (P_x, P_y) inside the viewport. *10 pts)*

    (b.) Which pixel point, (P_x, P_y), will be turned on by function point (-75,160)? *(10 pts)*

    (c.) What are the **mouse coordinates** for the point in part (b.)? *(5 pts)*



13. The arrow above has endpoints **A** and **C** defined by points with coordinates (3,3,0) and (9,3,0) respectively. In addition, the rotation point **B** is defined by the coordinates (5,3,0). You are viewing the figure from the positive Z-axis.

    (a) What will the coordinates of **A** and **C** be after a 60° rotation about the rotation point **B** within the xy-plane? *(10 pts)*

(b) In the space below, write an *OpenGL* code segment to carry out the transformations necessary in part (a.) of the previous problem. You may assume that routines exist to build transformation matrices, and for the processes of matrix-matrix and matrix-vector multiplication. If you choose to use *OpenGL* specific routines, they must be correctly ordered. The coordinate triples contain the coordinates **A** and **C** respectively. *(10 pts)*

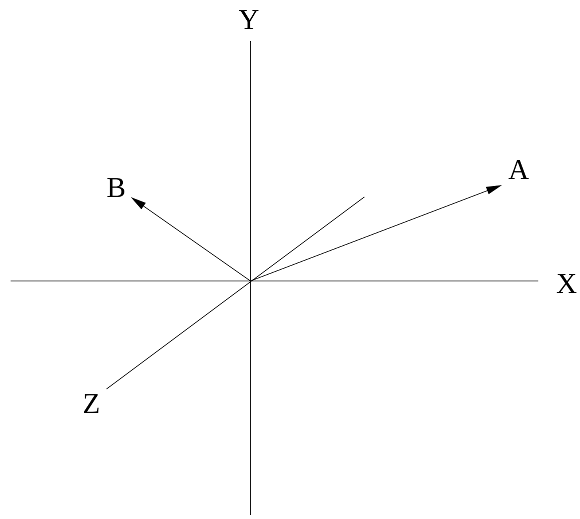
```
void movePoints( float x0, float y0, float z0,
                 float x1, float y1, float z1   )
{






     drawArrow( x0, y0, z0, x1, y1, z1 );


}
```

14. The program segment below draws a house like the one seen in class.

(a) Show explicitly how the code must me modified to place the house object in a display list. *(10 pts)*

```
void createWireHouse( struct house *face )
{
        int i, j;


          glPolygonMode(GL_FRONT, GL_LINE);
          glPolygonMode(GL_BACK, GL_LINE);


          for(j=0;j<7;j++)
          {

                glColor3f(face[j].color.red,
                          face[j].color.green,
                          face[j].color.blue);

                glBegin(GL_POLYGON);
                   for (i=0;i<face[j].sides;i++)
                   {
                      glVertex3f(face[j].point[i].x,
                                 face[j].point[i].y,
                                 face[j].point[i].z);
                   }
                glEnd();
          }

}
```

(b) Do any variables need to be declared globally to utilize the display list? If so, which one? *(10 pts)*



15. The diagram above shows two vectors starting at the origin. Vector **A** terminates at the point (8,3,4). Vector **B** terminates at the point (-4,3,3).

    (a) What is the angle between these two vectors. *(5 pts)*

    (b) You want to align the object AOB (where O represent the origin of the XYZ coordinate system) so that (1) $\vec{OB}$ is collinear with the positive Y-axis, (2) $\vec{OA}$ is contained in the YZ plane, (3) the lengths of $\vec{OA}$ and $\vec{OB}$ are preserved and (4) the angle between the vectors is preserved. Explicitly describe all the operations necessary to carry out this transformation. *(15 pts)*

(c) Carry out the requisite transformations you desribed on the previous page to give the end point of $\vec{OA}$ located in the YZ plane but not at the origin. *(15 pts)*

(Additional space ...)