READ ME FIRST

- Don't spend too much time on any one problem.
- Amount of time spent on a problem is not necessarily proportional to the points.
- Scan through the entire test and do the easy problems first.
- If something is not clear, ASK.
- BE NEAT. We cannot give you points for something that we can't read.
- Write down your assumptions.
- Write an outline of your solution for each problem.

1	20 points	Line Drawing	
2	15 points	Vectors	
3	20 points	Matrices	
4	10 points	Transformations	
5	15 points	OpengGL	
6	20 points	Clipping	
	100 points	GRAND TOTAL	

1. **Line Drawing (20 points)**

   In class, we derived the decision variable $P_i = 2\Delta y x_i - 2\Delta x y_i + 2\Delta y + \Delta x(2b-1)$ and its initial value P_1 for the case $m <= 1$. For that case, $P_1 = 2\Delta y - \Delta x$; and if $P_i < 0$, a sideway move is generated and $\Delta P_i = 2\Delta y$, else, a diagonal move is generated and $\Delta P_i = 2\Delta y - 2\Delta x$.

   For the case when $m > 1$, $P_i = -2\Delta y x_i + 2\Delta x(y_i - b) + 2\Delta x - \Delta y$. Specify (1) P_1; (2) x_{i+1}, y_{i+1}, and ΔP_i, when P_i is positive and negative respectively. Sketch and label your diagram.

2. **Vectors (15 points)**
   Given two planes P1: $2x + 3y - 4z + 5 = 0$ and P2: $-3x + 2y - z = 0$.
   Find a plane that is perpendicular to both of them and goes through the origin.

3. **Matrices (20 points)**

Given the following matrices:

$$M_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M = M_1 M_2$$

(a) (**5 points**) What do the matrices M_1 and M_2 do? Be precise/complete. Hint: M_1 is not a rotation.

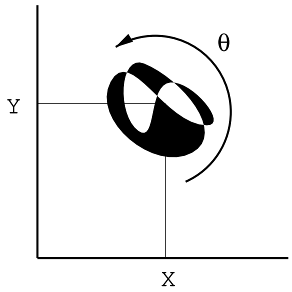
(b) (**5 points**) What does M do? Be precise/complete.

(c) (**5 points**) What is the inverse of M?

(d) (**5 points**) What is the determinant of M and M^{-1}?

4. **Transformations (10 points)**

   What is the 2D composite matrix M that will rotate objects about some arbitrary point (x,y) by θ degrees? You do not need to carry out the matrix multiplication, but you do need to specify the elements of each matrix.



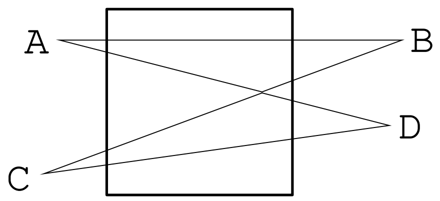
5. **OpenGL (15 points)**

   Write an OpenGL function to implement the above. Assume you have a function called: draw$_{picture}()$.

```
// This function rotates objects around the point (x,y)
// by theta degrees.  It also leaves the OpenGL state
// to the previous state prior to its call.

void RotateInPlace( float x, float y, float theta )
{











}
```

6. **Clipping (20 points)**

   Given polygon ABCD below, find the intermediate polygon as it is *successively* clipped against the following bounderies using the Sutherland-Hodgman algorithm. Redraw the intermediate polygon at each step and label any new vertices that may have been introduced.

   (a) (**5 points**) Clipped against the left edge:

   (b) (**5 points**) Clipped against the bottom edge:

   (c) (**5 points**) Clipped against the right edge:

   (d) (**5 points**) Clipped against the top edge: