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.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20 points</td>
<td>Line Drawing</td>
</tr>
<tr>
<td>2</td>
<td>15 points</td>
<td>Vectors</td>
</tr>
<tr>
<td>3</td>
<td>20 points</td>
<td>Matrices</td>
</tr>
<tr>
<td>4</td>
<td>10 points</td>
<td>Transformations</td>
</tr>
<tr>
<td>5</td>
<td>15 points</td>
<td>OpenGL</td>
</tr>
<tr>
<td>6</td>
<td>20 points</td>
<td>Clipping</td>
</tr>
<tr>
<td></td>
<td>100 points</td>
<td>GRAND TOTAL</td>
</tr>
</tbody>
</table>
1. **Line Drawing (20 points)**

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

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 \( \Delta P_i \), when \( P_i \) is positive and negative respectively. Sketch and label your diagram.
2. **Vectors (15 points)**
   Given two planes $P_1: 2x + 3y - 4z + 5 = 0$ and $P_2: -3x + 2y - z = 0$.
   Find a plane that is perpendicular to both of them and goes through the origin.
3. **Matrices (20 points)**

   Given the following matrices:

   \[
   M_1 = \begin{bmatrix}
   0 & 1 & 0 \\
   1 & 0 & 0 \\
   0 & 0 & 1 \\
   \end{bmatrix}
   \]

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

   \[
   M = M_1M_2
   \]

   (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 $\theta$ degrees? You do not need to carry out the matrix multiplication, but you do need to specify the elements of each matrix.

![Diagram showing rotation by angle $\theta$ around an arbitrary point $(x,y)$]
5. **OpenGL (15 points)**

Write an OpenGL function to implement the above. Assume you have a function called: `drawpicture()`.

```c
// 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 boundaries 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: