1. (18 pts) The point coordinate P can be expressed as \( P = 1*i + 2*j \), where \( i \) and \( j \) are basis vectors of unit length along the x and y axes, respectively. Describe the point \( P \) in terms of the 3 other coordinate systems given below.

\[
\begin{align*}
\text{i} & \quad \text{j} \\
\text{Ai} & \quad \text{Aj} \\
\text{Bj} & \quad \text{Bi} \\
\text{Ci} & \quad \text{Cj}
\end{align*}
\]

2. (5 pts) Write down the 4x4 matrix for rotating an object counterclockwise by 270 degrees around the Y axis.

3. (5 pts) Write down the 4x4 matrix for shearing an object by 2 in y and 3 in Z.

4. (10 pts) Decompose this matrix \( M \) into two matrices \( A \) and \( B \) such that \( p' = Mp = ABp \). Write down \( A \) and \( B \).

\[
\begin{pmatrix}
1 & 0 & 0 & 3 \\
0 & 2 & 0 & 2 \\
0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

5. (5 pts) Describe in words what \( M \) does, interpreting it as an operation in local coordinates that changes the coordinate frame. Be specific about the order of operations.

6. (5 pts) Describe in words what \( M \) above does, interpreting it as an operation in a fixed global coordinate system coordinates that moves the object. Be specific about the order of operations.

7. (5 pts) Give the OpenGL commands required to encode \( M \). You may assume the matrix stack has been initialized with \texttt{glIdentity()}.

8. (6 pts) Homogenize the point (8,15,9,5).

9. (15 pts) Given a triangle \( T \) with vertices \( a = (1, 1, 1, 1), b = (2, 2, 1, 1), c = (0, 0, -1, 1) \) and the transformation \( S = \)

\[
\begin{pmatrix}
2.828 & 0 & .707 & 1 \\
0 & 5 & 0 & 0 \\
-.707 & 1 & .707 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

Compute the vertices of \( T \) after applying transformation \( S \) to it.

10. (10 pts) Compute the normal of \( T \) before and after applying transformation \( S \) to it.
11. (16 pts) Give the 4x4 matrices that result from the OpenGL commands at the four lines A, B, C, and D below.

```c
glLoadIdentity();
glRotate(90, 0,0,1);
A
glTranslate(2,3,0);
B
glPushMatrix();
glTranslate(1,1,0);
glScale(1,.5,1);
C
glPopMatrix();
glScale(2,1,1);
D
```