1. List in the proper order the matrices which, when multiplied together, will flip the “hangman” upside down, rotate him 30° about his right foot, and then translate him so that his right foot is attached to the gallows. Do not multiply the matrices together. Assume for this problem that the center of the window is at (0, 0). (20 points)

Figure 1: Original Stickman Position
2. List the **one** OpenGL statement which will set the Current Transformation Matrix (CTM) so that it flips an object about the x-axis. Assume the identity matrix has already been loaded. (5 points)

3. Assuming the viewports in Figure 1 and Figure 2 are defined with “glViewport(0, 0, ww, wh)” and that the hangman object is defined the same size as shown, and centered at the origin (i.e. its *world* space coordinates are centered about (0,0)), list in order the segment of OpenGL code (transformations) which will transform the hangman in Figure 1 to its position in Figure 2. (10 points)

4. List the OpenGL statements needed to define the **vertices** for the stickman object so that it appears as shown. (10 points)
5. The *focal length* of a camera lens is the distance from the center of the lens to the point at which parallel rays of light will all be focused. For a pinhole camera, the focal length is the distance from the pinhole to the film plane. The dimensions of a frame of 35-mm film are about 24 mm x 36 mm. Assuming that the human visual system has an angle of view of 60 degrees, what focal length should we use with 35-mm film to achieve a natural view? (10 pts)
6. Sometimes object description files contain polygons which can cause problems when performing various graphics operations. Such polygons include:

- polygons with vertices which all lie on the same line, *collinear* polygons, and
- polygons with vertices which do not all lie on the same plane, *non-coplanar*

Using the following C data structures and routines write each of the following test routines: (10 pts each)

typedef double point[3]; /* pos 0, 1, 2 are x, y, and z, respectively */
typedef double vector[3];

typedef struct {
    int n;    /* number of vertices */
    point v[];   /* array of vertices -- NO index array as in labs */
    point N;    /* polygon (face) normal */
} polygon;

extern double dot(point v1, point v2);  /* returns v1 . v2 */
extern void cross(point v1, point v2, point res); /* res = v1 X v2 */
extern void normalize(point v);  /* length v set to 1 */

(a) int colinear(polygon p)
    /* Returns 1 if all the vertices of the polygon lie on the same line;
    Otherwise, returns 0. HINT: Use parametric equation of a line and
    compare vectors. */

(b) int non_coplanar(polygon p)
    /* Returns 1 if all the vertices of the polygon DO NOT lie on the same
    plane; Otherwise, returns 0. HINT: Use plane equation. */
7. In the $X$-$Y$ coordinate system, point $A$ has coordinates $(4,2)$. What coordinates does it have in the $X'$-$Y'$ coordinate system? (10 pts)

![Figure 3: Coordinate System Transformation Problem](image)

8. Answer all of the following questions:

(a) Perform the cross product on the vectors, $<8,6,14>$ and $<4,4,12>$. (5 points)

(b) Perform the dot product on the vectors, $<2,2,6>$ and the resulting vector (answer) from problem (8a). Show all your work. (5 points)
(c) Given two points, $P_0$ and $P_1$, write the parametric equation of a line passing through these two points. (5 points)