```
<html>

<script id="vertex-shader" type="x-shader/x-vertex">

attribute  vec4 vPosition;
attribute  vec4 vColor;
varying vec4 fColor;

uniform vec3 theta;
uniform int partNo;

void main()
{
    // Compute the sines and cosines of theta for each of
    //   the three axes in one computation.
    vec3 angles = radians( theta );
    vec3 c = cos( angles );
    vec3 s = sin( angles );

    // Remeber: thse matrices are column-major

    // ================================================
    // Build transformation matrix that is applied to part 3

    // The size in y is scaled by 2.
    // No change of size in x and z.
    mat4 modelScale3 = mat4( 1.0, 0.0, 0.0, 0.0,
                  0.0, 2.0, 0.0, 0.0,
                  0.0, 0.0, 1.0, 0.0,
                  0.0, 0.0, 0.0, 1.0);

    // translate along x such that the hinge on the origin and
    // along the z-axis.
    mat4 t1 = mat4( 1.0, 0.0, 0.0, 0.0,
                  0.0, 1.0, 0.0, 0.0,
                  0.0, 0.0, 1.0, 0.0,
                  -0.5, 0.0, 0.0, 1.0);

    // rotate around z for the angle specified in theta[0].
    mat4 rz = mat4( c[0], -s[0], 0.0, 0.0,
                s[0],  c[0],  0.0, 0.0,
                0.0,   0.0,  1.0, 0.0,
                0.0,  0.0,  0.0, 1.0 );

    // translate the hinge back and move the block up (to be
    // "connected" to part 2.
    mat4 t2 = mat4( 1.0, 0.0, 0.0, 0.0,
                  0.0, 1.0, 0.0, 0.0,
                  0.0, 0.0, 1.0, 0.0,
                  0.5, 2.0, 0.0, 1.0);
```

```
    // concatenate them together as M3 for later use.
    mat4 M3 = t2 * rz * t1;

    // ====================================================
    // Build transformation matrix that is applied to part 2

    // The size in y is scaled by 2.
    // No change of size in x and z.
    mat4 modelScale2 = mat4( 1.0, 0.0, 0.0, 0.0,
                 0.0, 2.0, 0.0, 0.0,
                 0.0, 0.0, 1.0, 0.0,
                 0.0, 0.0, 0.0, 1.0);

    // translation t1 is same as part 3.

    // rotate around z for the angle specified in theta[1].
    rz = mat4( c[1], -s[1], 0.0, 0.0,
               s[1],  c[1],  0.0, 0.0,
               0.0,   0.0,  1.0, 0.0,
               0.0,  0.0,  0.0, 1.0 );

    // the block is moved higher because part 1 is 3 nuit high
    t2[3][1] = 3.0;

    // concatenate them together as M2 for later use.
    mat4 M2 = t2 * rz * t1;

    // ====================================================
    // Build transformation matrix that is applied to part 1

    // The size in y is scaled by 3.
    // No change of size in x and z.
    mat4 modelScale1 = mat4( 1.0, 0.0, 0.0, 0.0,
                 0.0, 3.0, 0.0, 0.0,
                 0.0, 0.0, 1.0, 0.0,
                 0.0, 0.0, 0.0, 1.0);

    // rotate around y for the angle specified in theta[2].
    mat4 ry = mat4( c[2], 0.0, -s[2], 0.0,
              0.0,  1.0,  0.0,  0.0,
                s[2],  0.0, c[2], 0.0,
                0.0,  0.0,  0.0, 1.0 );

    mat4 M1 = ry;

    // Because the model is too big and too tall for the default
    // clipping volume, the following two matrices will scale
    // down its size and lower its position for display.

    mat4 scale = mat4( 0.2, 0.0, 0.0, 0.0,
                 0.0, 0.2, 0.0, 0.0,
```

```
                0.0, 0.0, 0.2, 0.0,
                0.0, 0.0, 0.0, 1.0);

    mat4 trans = mat4( 1.0, 0.0, 0.0, 0.0,
                0.0, 1.0, 0.0, 0.0,
                0.0, 0.0, 1.0, 0.0,
                0.0, -3.0, 0.0, 1.0);


    fColor = vColor;
    if (partNo == 1)
       gl_Position = scale*trans*M1*modelScale1*vPosition;
    else if (partNo == 2)
       gl_Position = scale*trans*M1*M2 *modelScale2*vPosition;
    else if (partNo == 3)
       gl_Position = scale*trans*M1*M2*M3*modelScale3*vPosition;
}
</script>

<script id="fragment-shader" type="x-shader/x-fragment">

precision mediump float;

varying vec4 fColor;

void
main()
{
    gl_FragColor = fColor;
}
</script>

<script type="text/javascript" src="../Common/webgl-utils.js"></script>
<script type="text/javascript" src="../Common/initShaders.js"></script>
<script type="text/javascript" src="../Common/MVnew.js"></script>
<script type="text/javascript" src="ass2_Q6.js"></script>

<body>
<br/>
<br/>
<br/>

<canvas id="gl-canvas" width="512"" height="512">
Oops ... your browser doesn't support the HTML5 canvas element
</canvas>



</body>
</html>
```