Math 401 - Groupwork #3
Computer Graphics (part 2)

Instructions: First download the file `cube.m` from ELMS onto your computer. Open the file in MATLAB. The file is a MATLAB script, which is a sequence of MATLAB commands (like a program). When you click "Run" in MATLAB, all of the commands will be executed. This script enters the vertices of a cube, and then plots a perspective projection. Here is a brief explanation of the code in this script:

```
syms T(a,b,c) RX(t) RY(t) RZ(t) P(d)
T(a,b,c)=[1 0 0 a;0 1 0 b;0 0 1 c;0 0 0 1];
RX(t)=[1 0 0 0 ; 0 cos(t) -sin(t) 0;0 sin(t) cos(t) 0 ; 0 0 0 1];
RY(t)=[cos(t) 0 sin(t) 0;0 1 0 0;-sin(t) 0 cos(t) 0 ;0 0 0 1];
RZ(t)=[cos(t) -sin(t) 0 0;sin(t) cos(t) 0 0 ; 0 0 1 0; 0 0 0 1];
P(d)=[1 0 0 0;0 1 0 0;0 0 0 0;0 0 -1/d 1];
```

This defines functions in MATLAB to create 4×4 translation and rotation matrices, to be applied to homogeneous coordinates in space. It also creates the perspective projection matrix, for which the "eye" is at the point $(0, 0, 15)$.

```
C = transpose([
    3   3   3   1
    3  -3   3   1
   -3  -3   3   1
   -3   3   3   1
    3   3  -3   1
    3  -3  -3   1
   -3  -3  -3   1
   -3   3  -3   1
]);
```

This code enters all of the points of a cube as the columns of a matrix `C`. Note that the points entered are homogeneous coordinates.

```
D = P(15)*C;
h = D(4,:).^(-1);
H = diag(h);
E = D*H;
```

The above code carries out the perspective projection, with $d = 15$. First, we apply the perspective projection matrix `P(15)` to `C`. Now the columns of `D` are homogeneous coordinates in which the fourth entry is not 1. So we have to extract the fourth entries, take their reciprocals, and multiply each column of `D` by the appropriate reciprocal. Here, we accomplished this by putting the reciprocals in a diagonal matrix `H`, and then multiplying `D` by `H`. Note that `D(4,:)` is the fourth row of `D`, and the `.^(-1)` raises each entry of the row to the -1 power (takes the reciprocal).

The remaining code creates a scatter plot of the projected vertices, and then uses the `plot` command to connect the vertices with edges (so that it looks like a cube). The edges on the top square are red, the edges on the bottom square are blue, and the vertical edges are black. You shouldn't edit this portion of the code.

Instructions: Work through the problems below in order. I recommend that you work directly in the script, adding/removing lines of code as needed. To "remove" a line, you should put a % in front of it (comment it out), rather than actually deleting it. You do not have to turn in your script at the end, so it is ok if you make a mess in it. When you apply a transformation to `C` in the script, you may just want to save the result as `C`, so that you do not need to edit the perspective projection portion. For example, you can type

```
C = T(1,4,3)*C;
```

after `C` is initially defined, but before the perspective projection.

Some problems below ask you to write down some matrices. Do so on a piece of paper. If the matrix was computed from other matrices, indicate symbolically what you did, though you don't need to show the computation (the computation should be done in MATLAB). For example, you can write things like $T(1,4,3)RX(\pi/2) =$ [explicit matrix written here] as your answers. You will also need to save some of the images you produce and submit them. There is a save button on the window that a figure pops up in. Please switch the file format so that the image saves as a bitmap (.bmp).

Do as many of the problems as you can, and do not stress about finishing all of them. You will receive a grade based on the correctness of what you've turned in, **provided that everyone is actively working on it during class.** At the end, turn in your paper answers and email me (`ayashins@math.umd.edu`) all of your .bmp files. Include all group members' names in the email body and cc all group members the email itself. Don't forget to attach the .bmp files.

1. For this part (and all other parts below), keep your axes fixed from -8 to 8 for both x and y. Experiment a little by changing the value of d in the perspective projection. (Approximately) Determine the smallest value of d so that the entire cube is visible in the figure (include the red lines on its top.) Write down this value and save your image as `cg1.bmp`.

2. For this part (and all other parts below), plot your final image as a perspective projection with $d = 15$, unless it states otherwise.

   (a) Write down the 4×4 matrix which rotates the cube around the z-axis by $\pi/6$ radians, and then rotates about the x-axis by $-\pi/3$ radians. Apply this matrix to the cube and plot the perspective projection. Save this as `cg2a.bmp`.

   (b) Write down the 4×4 matrix which does the same two rotations, but in the opposite order. Apply this matrix to the cube and plot the perspective projection. Save this as `cg2b.bmp`.

3. Consider the rotation by $2\pi/3$ radians whose axis is the horizontal segment connecting $(-3,3,3)$ to $(3,3,3)$ (with orientation in the positive x-direction.) Write down the 4×4 matrix of this rotation, apply it to the cube, and plot. Save your image as `cg3.bmp`.

4. A horizontal shear in $\mathbb{R}^3$ is a linear transformation $f : \mathbb{R}^3 \to \mathbb{R}^3$ whose matrix is $\begin{bmatrix} 1 & a & b \\ 0 & 1 & c \\ 0 & 0 & 1 \end{bmatrix}$ (I called it horizontal because the shearing is parallel to the xy-plane.) Apply the shear with $a = b = c = 0.5$ to the cube and plot it. Then experiment and increase the values of a, b, c until the upper right corner of the cube is very close to the upper right boundary of the figure. Write down your a, b, c values and save this image as `cg4.bmp`.

5. (a) Find the matrix of the transformation which rotates the cube so that we get a view of the cube directly through one of the corners towards the center. That is, the opposite corner should be directly behind the front corner. (Hint: you'll want to rotate the cube so that the diagonal [the segment which connects opposite points] rotates onto the z-axis. Don't just find it by experimentation; do the geometry to figure out the appropriate angles.) Apply the matrix and plot the perspective projection. Save this as `cg5a.bmp`.

   (b) Consider a rotation by $\pi/6$ radians whose axis of rotation is one of the diagonals of the cube. Write down the matrix for such a transformation, apply it to the cube, and save the image as `cg5b.bmp`

6. Orthogonal projection onto the xy-plane is different from perspective projection. Orthogonal projection onto the xy-plane is a linear transformation which maps (x, y, z) to $(x, y, 0)$.

   (a) Viewing orthogonal projection as a linear transformation $f : \mathbb{R}^3 \to \mathbb{R}^3$, find its standard 3×3 matrix. Then write down the 4×4 matrix of orthogonal projection to be used on homogeneous coordinates. Apply the orthogonal projection to the original cube, plot the result in the xy-plane, and save it as `cg6a.bmp`. (Make it so your script does the orthogonal projection instead of the perspective projection. You may want to comment out the perspective projection lines.)

   (b) Now plot the perspective projection of your original cube (in its original position) with $d = 10000$, and save the image as `cg6b.bmp`. What do you suspect the relationship between orthogonal projection and perspective projection is?