Math 401 - Groupwork #2
2D Computer Graphics

Instructions: First, download the file `smiley.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 in order from top to bottom. This script plots a bunch of points that make a smiley face. Here is a brief explanation of the code in this script:

```
syms T(a,b) R(t)
T(a,b) = [1 0 a;0 1 b;0 0 1];
R(t) = [cos(t) -sin(t) 0;sin(t) cos(t) 0;0 0 1];
```

This defines functions in MATLAB to create 3×3 translation and rotation matrices for the plane. Afterwards, you can type things like `T(3,-2)` or `R(pi/4)` and MATLAB will create the appropriate 3×3 translation or rotation matrix (to be used on homogeneous coordinates of points in the plane.)

```
S = transpose([
        -1     2     1
         1     2     1
        -3     0     1
         3     0     1
        -1    -3     1
         1    -3     1
        -2    -2     1
         2    -2     1
         0    -3.5   1
        -2.5  -1     1
         2.5  -1     1
])
```

This code enters all of the points of a smiley face as the columns of a matrix `S`. Note that the points entered are homogeneous coordinates in the plane. For convenience, each point was entered as a row, and then a transpose was taken (so all rows become columns).

```
scatter(S(1,:),S(2,:),'filled')
axis square
axis([-5 5 -5 5])
```

This creates a scatter plot of all of the points in the smiley face. The command `S(1,:)` returns the first row of `S` (all the x-coordinates) and `S(2,:)` returns the second row of `S` (all the y-coordinates). Note that we do not need the third row of `S` for the plot, because it is just the 1's from the homogeneous coordinates.

Now say we want to shift the whole smiley to the left by 2 units and then plot it. Here's the code to do it:

```
SN = T(-2,0)*S;
scatter(SN(1,:),SN(2,:),'filled')
axis square
axis([-5 5 -5 5])
```

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.

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 $T(1,4)R(\pi/2) =$ [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).

At the end, turn in your paper answers and email me (`nmanning@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.

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.**

1. Write down the matrix that rotates the face around the left eye by $\pi/6$ radians. Apply the matrix to the smiley face and plot. Save this as `cg1.bmp`.

2. Write down the matrix that moves the left eye to the origin and rotates the right eye to $(\sqrt{2}, \sqrt{2})$. Apply this matrix to the (original) smiley face and plot. Save this as `cg2.bmp`.

3. Use the matrix that reflects through the x-axis (and some other stuff) to write down the matrix that reflects through the line $y = \sqrt{3}x$. (Don't just use the final formula we saw in class. Compute it directly using other matrices as building blocks.) Apply this matrix to the (original) smiley face and plot. Save this as `cg3.bmp`.

4. Write down the matrix that reflects through the horizontal line connecting the two eyes. Apply this matrix to the (original) smiley face and plot. Note that it goes off the picture. Adjust your axes so that the smiley is visible, and save as `cg4.bmp`.

5. Revert your axes to the original values (both -5 to 5). A matrix of the form $\begin{bmatrix} 1 & c \\ 0 & 1 \end{bmatrix}$ is a horizontal shear of the plane. Find and write down the largest value of c for which the sheared smiley face remains within the window. Find this value approximately by experimentation (trial and error), rather than exactly by solving for it. Save your sheared smiley as `cg5.bmp`.

6. Using the original smiley, replace the line

   `scatter(S(1,:),S(2,:),'filled')` with `plot(S(1,:),S(2,:))`

   and observe that it draws a line segment connecting consecutive points in `S`. It should look like a mess because of the order of points in S. Make adjustments to draw a nice smiley face with a continuous smile. Use `scatter` for the eyes and `plot` for the smile. You need to use the `hold` command so that the `plot` doesn't delete what the `scatter` created. You can do the following:

```
scatter(something goes here)
hold on
plot(something else here)
hold off
```

   Save your image as `cg6.bmp`.

7. Is there a sequence of linear transformations that will leave the eyes in place but turn the smile into a frown? If so, write them down and make an image, saved as `cg7.bmp`. If not, explain why not.