Homework 2 (v1)

1) (50 points) You are expected to code a program (in C/C++ or Java) that admits as input a system of linear equations and then prints on screen the unique solution to this system (if it exists); or an error if there is no unique solution to the said system, or if it cannot be calculated.

Example:

```
$./solver -i system.txt -m GESP
```

the file system.txt in this case will contain the comma separated augmented matrix of the system. For example if the system is:

$$x_1 + x_2 = 3$$
$$x_1 - x_2 = 0$$

then your input file should contain:

1,1,3
1,-1,0

the -m flag denotes the method to use for solving the system:
GESP: Gaussian elimination with scaled partial pivoting
JCB: Jacobi's method. In this case, assume the stopping criterion to be:

$$\frac{\|x^{(k)} - x^{(k-1)}\|}{\|x^{(k)}\|} < 10^{-3}$$

using the Euclidean norm. Test your program with the systems from exercises 6.2-1c and 7.3-1a,1b.

2) (50 points) Digital images are stored as 2D matrices. For example an image of size 1000x1000 is basically a square matrix, every element of which is a non-negative integer representing the brightness of that point. The greater the matrix element, the brighter that point (or pixel) is.
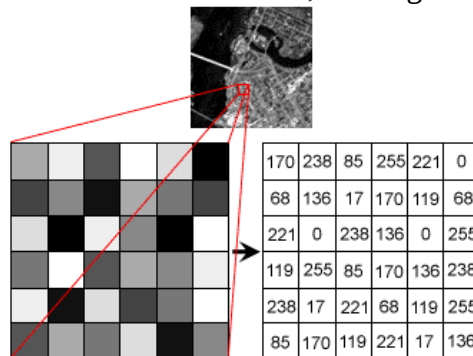

The coordinate system of a digital image is as follows (attention, the y axis is downwards):
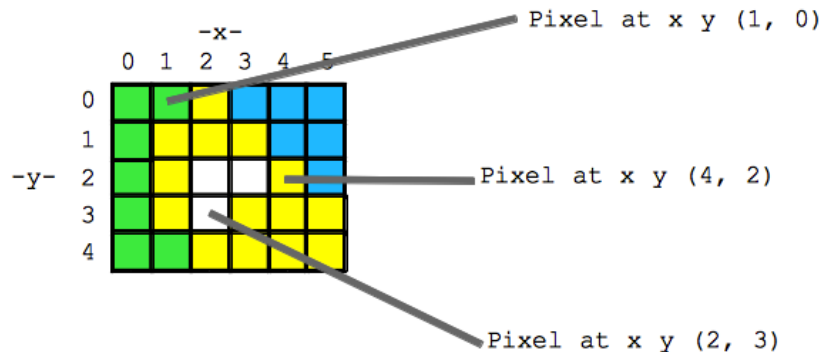
Hence, if K is an image, K(x,y) represents the pixel (or point) value at position (x,y).

Digital images have many problems..one of which is **image registration (görüntü çakıştırma)**.

Image registration is an often encountered problem especially in medical and military image processing applications. Take a look at the following example:


F   B
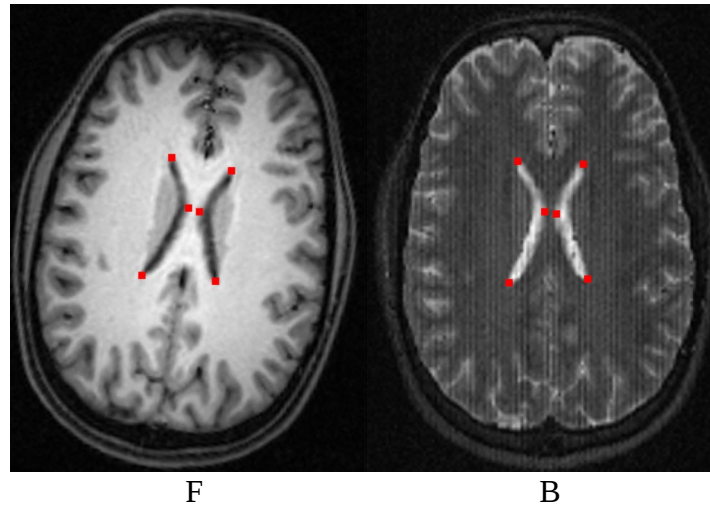
These are 2 magnetic resonance (MR) images of the **same** patient. The problem is that they are not "aligned". The left image is clearly a bit "tilted" to the right and also slightly larger. This happens very often, as the patients do not lie at *exactly* the same spot every time they have their MR images taken. It also happens when military satellites or aircraft take spy photographs from the sky of the same location at different times.

We want to "register" these two images, i.e., we want them to be superposed exactly.
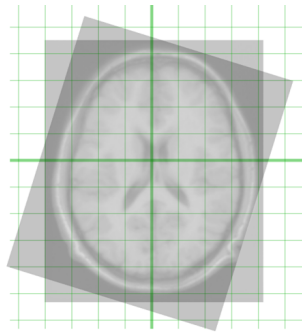
To achieve this, **we need to calculate the 3x3 transformation matrix that will convert image F into something like image C that aligns perfectly with image B, both in terms of rotation and scale.**

C
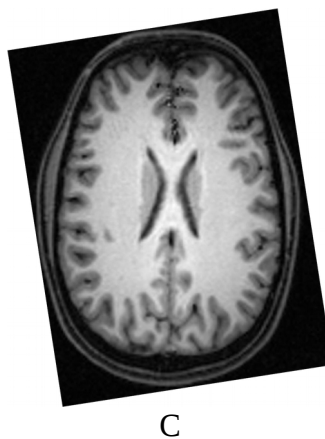
So the one million dollar question is given F and B, how do we find the matrix that converts the image pixels of F into C (that is aligned with B)?

Fortunately for us, there are some reference points (shown with red on both F and B, provided by some expert who knows what she's doing). These are the 3 points for which we know their correspondences.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Coordinates in image **B(x,y)**	Coordinates in image **F(x',y')**
[1,2]	[2,2]
[2,1]	[-1,4]
[3,1]	[-4,4]

Now, if you can calculate the matrix A:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

that would mean you'd know how to transform B into F (we want the opposite F->B). Consequently, if you calculate the **inverse of A**, you would be able to transform F and align it with B! **Your goal is to calculate** A^{-1} .

Express the problem as a system of linear equations where the unknowns are the coefficients of A. Calculate A, and then take its inverse by solving $AZ = I$. You are free to use any of the methods presented in class for solving the two systems of linear equations. Do not code anything, but please send a detailed and formal report about how you solved the problem.

Hint:

$$x a_{11} + y a_{12} + 1 a_{13} + 0 a_{21} + 0 a_{22} + 0 a_{23} = x'$$
$$0 a_{11} + 0 a_{12} + 0 a_{13} + x a_{21} + y a_{22} + 1 a_{23} = y'$$

Good luck.