1. This problem is based on the Computer Graphics worksheet.
   (a) Download the free Image Processing Toolbox through the MathWorks website, if it's not already part of your standard MATLAB installation.
   (b) Read the documentation on imwarp. Show me a basic example of how it works.
   (c) `http://starwars.wikia.com/wiki/Opening_crawl` Take a picture of some text, embed it in 3-dimensional space, then project until it looks right. Your pictures should be 2-dimensional – you're actually doing a projection.

---

2. This problem is based on the Computer Graphics worksheet.
   (a) The projection acts a lot like a lightbox – the point from which you're projecting is the light source, and what you see is the shadow. Take a cube, or some other shape, and draw lines extending from the projection point to the plane onto which you're projecting. You'll probably want to write a function to automate the process.
   (b) Repeat with some other funny shape. You should get some really nice 3D pictures here; the goal is to find pictures that visually explain what projection does.

---

3. This problem is based on the Curve Fitting worksheet.
   (a) Read this: `https://www.researchgate.net/publication/247907373_Stupid_Data_Miner_Tricks_Overfitting_the_SP_500`. It's moderately entertaining. The first part is interesting if you have any Stat, but the polynomial fit is the interesting bit.
   (b) Regression often works differently at different time scales, especially for economic data. Find and import the data for the S&P 500 – probably the easiest way to do this is to download the historical data from `https://finance.yahoo.com/quote/%5EGSPC/history?p=%5EGSPC`.
   (c) Fit a curve to the data at different scales (for example: the past week, the year 2015, stuff like that). I'd like to see a period at which the growth is well-approximated linearly, one at which it's well-approximated exponentially, and one at which it's well-approximated quadratically. These might not all exist, but I'd like to see what happens.

There's more problems on the back!

---

4. This problem is based on the Cryptography worksheet.

   (a) There's nothing particularly special about 2 except that it's prime and that a bit is either off (0) or on (1). Most modern cryptographic algorithms use a much larger prime p, and quantum computers use qubits that can represent large primes exactly as classical computers use bits to represent arithmetic mod 2.

   (b) Modify the functions lfsr, lfsrlength, and lfsrsolve to work mod p. (This means that they should now take another argument.)

   (c) Using imshow like in the worksheet, show me an LFSR mod 3. You'll want to multiply by 127 rather than 255 when you imshow so you can see the three possible values of a trit (yes, that's the name for a "bit" which can be 0, 1, or 2). Find a length-5 base-3 LFSR of maximum period 242.

   (d) 127 is also prime, conveniently. Find a length-2 base-127 LFSR of maximum period 16128. (Don't try to do this one visually. A length 2 LFSR is completely determined by any pair of numbers, so generate 20,000 base-127 digits with the seed data (0,1) and then search for the next time that seed appears using find().)

---

5. This problem is based on the Cryptography worksheet. All of the stuff you need for this one is in the Files tab.

   (a) Read Washington's notes on breaking one-time pads. He doesn't call it this, but the technique he uses is called "crib dragging."

   (b) Show me what the .m files string2bits and bits2string do.

   (c) Break the three messages.

---

6. This problem is based on the Coding Theory worksheet.

   (a) Implement the Hamming (15,11) code: write down its generator and check matrix, and show me it working like you did on the fourth worksheet.

   (b) Implement the 5-repeat code. This code is determined by $0 \mapsto (0, 0, 0, 0, 0)$ and $1 \mapsto (1, 1, 1, 1, 1)$.

   (c) syndromedecode.m doesn't work on the 5-repeat code because 5-repeat is 2-errors-correcting. Write a function called repeatdecode that can correct the errors.

   (d) (If you're feeling extremely ambitious.) Washington's Cryptography book also describes the Hadamard code, which is a [32,6] code – that is, one with about the same coding gain as 5-repeat – with much more robust error-correcting properties. Implement the Hadamard matrix, the check matrix, and the decoding algorithm described on page 301.