These are possible final projects for the course. If you have a suitable project which interests you (i.e., perhaps something related to a thesis you are working on), you may discuss it with me to see if you can substitute it for one of these.

Otherwise, choose one of these:

1. (Cryptanalysis) Complete Exercise 4.3 from the notes. You will devise a method for breaking the mono-alphabetic homophonic substitution cipher, and find the original message for the given ciphertext. You should turn in:
   - A clear description of the method.
   - A clear, well-reasoned justification of the method or algorithm.
   - The computer code you’ve written.
   - The decrypted message.

2. (Point-in-polygon problem) This is a revisit of Exercise 1.9. For this project, you should convincingly solve the point-in-polygon problem. You should devise and implement an algorithm for deciding if a given point is interior or exterior to a (not necessarily convex) polygon given by an ordered list of vertices. You should turn in:
   - A clear and precise description of the algorithm. If you choose one of the two methods I outlined, be aware that I gave only sketches of the algorithms - there are many details to be filled in!
   - A complete proof of the correctness of the algorithm. If you choose one of the algorithms I described, you may attempt to fill in the details from the outlined proofs I gave. However, I will grade such a proof with more scrutiny than an original attempt.
   - An electronic copy of your program source code (so I may compile it and submit test cases).

3. (Fractals) You will expand on the program you wrote for Exercise 2.2. Specifically, do some research to find two types of fractals other than the Mandelbrot set and modify your program to generate some of them. One of the two types should be the so-called ‘Newton type’, generated by iterations of Newton’s method (for this, you can use #include <complex.h>, which will get you access to a type called complex double, and a special identifier I for the complex number i). The other will be of your choosing. You should turn in:
   - A two page paper describing the mathematics of the fractals you’ve generated.
   - Several ‘pretty pictures’ you generated (under Windows, you can copy the screen with ALT-PRINT SCREEN and paste it into ‘Paint’ or your favorite image editor.)
   - An electronic copy of the source code you’ve written to generate the fractals.

4. (Newton’s method) Modify program newton1.c from Section 1.8 of the course notes to find complex roots. You cannot use complex.h - instead you should write four functions to complex addition, subtraction, multiplication and division. Using these, modify or rewrite the program newton1.c so it can find complex roots of the given polynomial (which may still be assumed to have only real coefficients). You should turn in the following:
   - A precise description of Newton’s method for finding complex roots of a polynomial.
   - A justification (i.e., sketch of proof) for why it works.
   - A root of the polynomial $f(x) = x^4 + x^3 + x^2 + x + 1$ to at least 10 digits of precision.
   - An electronic copy of your source code.

As always, you can work on this project in groups of upto three people. If you work in a group, turn in one project as a group, and clearly indicate who was responsible for each portion of the work. All written materials to be turned in should be either typed or very neatly handwritten. The project is due on May 10, but please note that the last class is on May 3!