Project Descriptions

There are three computer programming projects for this course. See below for their descriptions. You may pick one, which will be counted for credit. Each additional project that you submit will count as extra credit towards your final grade.

Directions/Format

Write a computer program in C/C++ or Java which performs one of the indicated operations described below. If you would like to write your program in a different programming language, be sure to get permission ahead of time in order to verify that I will be able to run your program.

- Please use the following naming convention, where the project tags are listed below.
  - If only one file, name it as follows: [your last name]_[project tag]
  - If multiply files, use the following: [your last name]_[project tag]_[file name]
- You can submit your projects by either emailing them to me or by bringing them to me on a flash drive. Note: if you plan to email me the file(s), be sure to leave enough time before the due date since the uncc email system tends to treat any code files as viruses.
- Be sure to properly comment your code. In the header, make sure to include your name as well as how the program should be run.

Grading

The projects will be graded based on the following criteria:

- whether the program will compile,
- whether the code uses the correct concept/formula/algorithm that the project highlights and does not just use a predefined function in one of the coding language's libraries, and
- whether the program does what it is supposed to do.

Due Date: December 12, 2013

Project Descriptions

1. The Greatest Common Divisor [GCD]

   Textbook Section: 1.4
   Directions: Implement the Euclidean Algorithm to compute the greatest common divisor of any two integers. These two numbers should be able to be positive, negative, zero, or any combination.
   Input: 2 integers
   Output: 1 integer (this should be the gcd of the given two integers)

2. Fibonacci-like Sequences [SEQ]

   Textbook Section: 3.5
   Directions: Given a degree 2 recurrence relation $a_n = ca_{n-1} + da_{n-2}$ and two initial conditions: a_i and a_{i+1} , find the closed formula (explicit formula) for the sequence. Note, in our in-class examples, i has always been 1 or 0.
   Input: a recurrence relation, 2 initial conditions, and i
   Output: the closed form expression

3. Connectivity of a Digraph [DIW]

   Textbook Section: 4.8
   Directions: Implement Warshall's Algorithm in order to find the transitive closure of a given relation by the use of the matrices W_k.
   Input: the relation R defining the digraph
   Output: the transitive closure R^{∞}