Project Descriptions

Math 1165: Introduction to Discrete Structures

Directions and Formating

Write a computer program in C/C++ or Java which performs the indicated operations described in the project. If you want 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.

- Properly comment your code. In the header, include your name as well as how the program should be compiled and run.
- Submit your executable code through Canvas. Do not submit compiled code, a PDF or text file of your code. I am expecting a .cpp or .java file.
- Warning: I will run the programs in Linux. If you use Microsoft Visual Studio, make sure your code will compile outside of that environment. (There is a linux computer lab on the third floor of Fretwell.)

Grading

The projects will be graded based on the following criteria:

- whether the program compiles (25%),
- 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. Note, if you do not use the correct method, you will not get full credit, so be sure to read the directions carefully (25%), and
- whether the program does what it is supposed to do (50%).

Due Dates

Each project has two dates. The due date is when the project is due. Should you choose to revise or improve the project, the revision date is when the revised project is due. Note, if you do not submit your project by the first due date it is considered late. In this case, your project will be assessed a 20% late penalty, and you will not have a chance to submit a revised program. If you submit your project after the revision date, your project will not be graded.

Project Descriptions

Project 1: The Greatest Common Divisor [GCD]

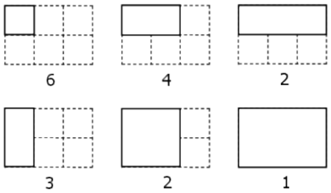
Textbook Section: 7.36
Directions: Implement the Euclidean Algorithm to compute the greatest common divisor of any two integers. These two integers should be able to be positive, negative, zero, or any combination thereof. (Hint: be careful when dealing with zero and negative numbers.)
Input: 2 integers
Output: 1 integer (ie: the gcd of the given integers)
Extra Credit: Find the linear combination of the 2 given integers which sums to their gcd.
Due Date: September 15
Revision Date: October 6

Project 2: Fibonacci-like Sequences [SEQ]

Textbook Section: 4.23
Directions: Given a degree 2 recurrence relation $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ and two initial conditions: a_0 and a_1, where $c_1^2 + 4c_2 \geq 0$, find the closed (explicit) formula for the sequence. (Hint: you will need to consider two cases: real and repeated roots.)
Input: c_1, c_2, a_0, and a_1 (you will need to check if $c_1^2 + 4c_2 \geq 0$)
Output: the closed formula expression for a_n
Extra Credit: Add a feature to be able to calculate the value of a_n for a given n.
Add a feature to be able to start at any index i (ie: given $a_1 = 2$ and $a_2 = 4$).
Be able to run the program with imaginary roots (ie: if $c_1^2 + 4c_2 < 0$)
Due Date: October 27
Revision Date: November 17

Project 3: Counting Rectangles [REC]

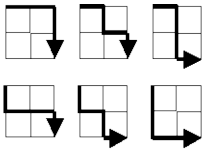
Textbook Section: 2.8
Set Up: A 2×3 rectangular grid contains exactly eighteen different rectangles:



Directions: Compute the number of different rectangles an $n \times m$ rectangular grid contains for $n \geq 1, m \geq 1$. Allow the user to choose n and m.
Input: the size of the grid (ie: $n = 5$ and $m = 3$ where the grid is 5×3)
Output: the number of rectangles contained in the grid
Due Date: November 21
Revision Date: December 5

Project 4: Counting Lattice Paths [LAT]

Textbook Section: 3.17
Set Up: Starting in the top left corner of a 2×2 grid, and only being able to move to the right and down, there are exactly 6 routes to the bottom right corner.



Directions: Compute the number of these types of routes through an $n \times n$ grid for $1 \leq n \leq 20$. Allow the user to choose n. (Hint: be careful of memory overflow issues.)
Input: the size of the grid (ie: $n = 10$ where the grid is $n \times n$)
Output: the number of routes from the top left corner to the bottom right corner
Extra Credit: allow the user to choose any n
Due Date: December 5
Revision Date: December 14