/\*

Name: Daniel Pinedo

Class: CS2

Assignment #: 3 prep

All Compilers Used: n/a

Operating Systems: n/a

Date/time of last successful compile:

Email: d.p@ieee.org

\*/

Euclidian algorithm design (C++)

Definition: The greatest common divisor gcd is the largest natural number that divides both a and b without leaving a remainder.

Algorithm design:

// Below is recursive algorithm

// function prototype

gcd(a, b) = int gcd (int a , int b); // only solves for positive numbers

if ( b == 0 ) { //step 1

return a;

}

else if (a >= b && b > 0) { //step 2

return gcd (b , a%b); // only works if a >= b

}

else { //step3

return gcd (b , a); // if a < b switch them.

}

// example debugging procedures below

/\*

a = 10, b = 5

step 1. 5 != 0 therefore false

step 2. 10 > =5 therefore true. return gcd(a=5, b=0).

step 1. 0==0 therefore true. return 5

a = 341, b = 743

step 1. b != 0 therefore false

step 2. 341 < 743 therefore false

step 3. return (a=743, b = 341)

step 1. 341 != 0 therefore false

step 2. 743 >= 341 therefore true. return gcd(341, 61)

step 1. 61 != 0 therefore false

step 2. 341 >= 61 therefore true. return gcd(61, 36)

step 1. 36 != 0 therefore false

step 2. 61 >= 36 therefore true. return gcd (36, 25)

step 1. 25 != 0 therefore false

step 2. 36 >= 25 therefore true. return gcd(25, 11)

step 1. 11 != 0 therefore false

step 2. 25 >= 11 therefore true. return gcd(11, 3)

step 1. 3 != 0 therefore false

step 2. 11 >= 3 therefore true. return gcd(3, 2)

step 1. 2 != 0 therefore false

step 2. 3 >= 2 therefore true. return (2, 1)

step 1. 1 != 0 therefore false

step 2. 2 >= 1 therefore true. return (1, 1)

step 1. 1 != 0 therefore false

step 2. 1 >= 1 therefore true. return (1, 0)

step 1. 0 == 0 therefore true. return 1.

\*/