Peyton Cross

CMSI 402

Assignment #3

13 March 2017

**Assignment #3**

7.1 You could probably just comment a reference to the Wikipedia page. The comments don’t add much other than state the obvious.

7.2 The programmer probably copied the code somewhere else and simply stated what each line does. And he probably added comments after he wrote the code.

7.4 It should simply verify that a and b are greater than 0

7.5 Error handling should be handled by the calling code.

8.1

Boolean ValidateIsRelativelyPrime (long a, long b) {

If (GCD(a,b) == 1 and isRelativelyPrime(a,b)) {

Assert true;

} else {

Assert false;

}

private long GCD( long a, long b )

{

// Get the absolute value of a and b

a = Math.abs( a );

b = Math.abs( b );

//Repeat until we're done

for( ; ; )

{

// Set remainder to the remainder of a / b

long remainder = a % b;

// If remainder is 0, we're done. Return b.

If( remainder == 0 ) return b;

// Set a = b and b = remainder.

a = b;

b = remainder;

};

}

8.3 Since I do not know how it is implemented I used a black box test. You could theoretically use an exhaustive but that is a lot of asserts.

8.5 No because it is pretty much the same exact method I used to validate the code.

8.9 Blackbox because they do not rely on knowing how the method was implemented.

8.11

Alice and Bob: 5 \* 4/2 = 10

Alice and Carmen: 5 \* 5/2 = 12.5

Bob and Carmen: 4 \* 5/1=20

(10 + 12.5 + 20)/3 = Approximately 14 bugs

8.12 It means that there is no way to know how many bugs there are. You can pretend that each tester found one bug in common instead.