Practice Exercise #15: Count pairs of coprime numbers

http://www.comp.nus.edu.sg/~cs1010/4_misc/practice.html

Reference: Week 4

Date of release: 1 September 2014

Objective: Repetition statement

Task statement:

Write a program **countCoprimes.c** to read in a positive integer larger than 2, and smaller than or equal to 1000. Let's call this *limit*. There is no need for you to do input data validation.

Your program is to determine the number of pairs of integers in the range [2, *limit*] which are coprime.

Two positive integers *a* and *b* are said to be **coprime** (or **relatively prime**) if the only positive integer that divides both *a* and *b* is 1. Hence, 4 and 9 are coprime, but 24 and 15 are not.

For example, if *limit* is 7, then there are 11 pairs of coprime integers: (2, 3), (2, 5), (2, 7), (3, 4), (3, 5), (3, 7), (4, 5), (4, 7), (5, 6), (5, 7) and (6, 7).

Note that the pairs (2, 3) and (3, 2) are considered the same, so they are counted as one pair.

Your program should contain a function

```
count_coprimes(int limit)
```

that takes in *limit* and computes the number of pairs of integers in the range [2, *limit*] that are coprime.

The skeleton program provided contains a function **gcd()** that computes the Greatest Common Divisor of two integers. This function works, but is badly designed (and runs very slowly!). In Week 6 discussion session, we will discuss this and you will be shown a better version.

Because of the bad **gcd()** function given, your program will be tested with inputs not more than 1000.

Sample runs:

```
Enter limit: 7
Answer = 11

Enter limit: 100
Answer = 2944
```