## Practice Exercise #14: Count numbers not divisible by 2 divisors

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 **countNumbers.c** to read in a pair of distinct positive integers. If the inputs are not positive or distinct, keep reading the next pair until the values are positive and distinct. Let's call these inputs *divisor1* and *divisor2*.

Your program next reads in another pair of distinct positive integers *limit1* and *limit2*, where *limit1* < *limit2*. Again, if the two inputs are not positive or if *limit1* is not smaller than *limit2*, keep reading until the desired limits are read.

Your program then counts the number of integers from *limit1* to *limit2* (both inclusive) which are not divisible by *divisor1* and also not divisible by *divisor2*.

For example, if *divisor1* = 7, *divisor2* = 3, *limit1* = 10, and *limit2* = 30, then the answer is 12. (The 12 integers are 10, 11, 13, 16, 17, 19, 20, 22, 23, 25, 26 and 29.)

Your program should contain a function

```
count_numbers(int, int, int, int)
```

that takes in the necessary arguments and returns the answer.

Note: As this is an exercise on repetition statement, you should use a loop, instead of computing the number of values divisible/not divisible by a divisor in the given range of values.

**Sample runs:**

```
Enter 2 divisors: 7 3
Enter lower and upper limits: 10 30
Answer = 12


Enter 2 divisors: 6 6
Enter 2 divisors: -3 2
Enter 2 divisors: 5 8
Enter lower and upper limits: 20 10
Enter lower and upper limits: -20 -10
Enter lower and upper limits: 100 100
Enter lower and upper limits: 151 180
Answer = 21
```