Programming Project 02

This assignment is worth 20 points and must be **completed and turned in before 11:59 on Monday, January 30th , 2017.**

Assignment Overview
This assignment will exercise your ability to utilize control statements for C++ and to write some simple functions.

Background
Prime numbers, those numbers who have as integer divisors only the value 1 and the number itself, are a type of number that has been known since antiquity (Euclid's Elements for example, circa 300 BC).

However, sometimes we can describe numbers as *relatively prime,* that is numbers that share no factors except 1

Euler, one of the world's greatest mathematicians, described a function which eventually became associated with the Greek letter φ and was subsequently called Euler's totient. The idea is this:

Given an integer number $n >= 1$, for every number d in the range $1 \ldots n$ check to see if each d is relatively prime to n. If so, then d is a totative of the sequence, and we increase the count of the totient function φ by 1.

To do this we need a method to check the "relative primeness" of two numbers. Though there are better ways, a simple way is the greatest common divisor (gdc) method, and an easy way to do that was propose by Euclid more than 2300 years ago, appearing in his book Elements (300 BC). If the gcd of two numbers if 1, then they are relatively prime.

Here is an example. Let us say we would like to calculate φ(9). We would check every integer between 1 and 9 and observe the following:
- the gcd (1,9) is 1, 1 is a totative of 9, φ is 1
- the gcd (2,9) is 1, 2 is a totative of 9, φ is 2
- gcd (3,9) is 3, 3 is **not a** totative of 9, no change to φ
- etc.

In this sequence 3,6,9 are not totatives of 9 (they all have a gcd ≠ 1), φ(9) has a value of 6 and the totative list of 9 is 1,2,4,5,7,8.

Project Description / Specification
For this program we will do re-directed input.

Input
- Each test case is an individual file
- Each test case contains two lines:
  - first line, two integers each separated from the other by a single space. You are to calculate the gcd of these two numbers
  - second line, a single integer. You will calculate φ and provide a list of totatives for this number

Output for each test case will be:

- On the first line, the gcd of the first two numbers
- On the second line the list of totatives. Each number is separated from its neighbor by a single space.
- On the third line, the value of φ for the second line number

Requirements

1. If ***any*** input number from the test case is not an integer >= 1, print only the word Error on one line for that case and nothing else.
2. You cannot count on the order of the two gcd numbers, and order matters in the algorithm. You have to check and place them in the correct order to make your calculation
3. Write a function `gcd` that takes two argument `long` types and returns a `long`, the gcd of the two arguments. Euclid's algorithm is easily found on the web, go look for it!
4. Write a function `phi` that takes a single `long` argument and returns a `long`, the totative count for the range from 1… argument.
    a. Note that the function will have to print the totatives as they are found.
    b. Each number in the totative list is separated from its neighbor by one space.
5. Write the `main` function to perform all of the tasks indicated, including gathering input and using the `gcd` and `phi` function to make the calculation.
6. Place all code in a **single file** named **proj02.cpp**

Deliverables

proj02.cpp -- your source code solution (remember to include your section, the date, project number and comments).

1. Please be sure to use the specified file name, i.e. "**proj02.cpp**"
2. Save a copy of your file in your CS account disk space (H drive on CS computers).
3. Electronically submit a copy of the file.

Notes