Mathematical Methods and Numerical Techniques I — Homework Problems

1. **Reading Assignment**
   Read Boas, remaining parts of Chapter 1.

2. **Products of Vectors**
   Write a C++ program that accepts the components of two vectors \( \mathbf{a} \) and \( \mathbf{b} \) as keyboard input, and delivers their scalar product \( \mathbf{a} \cdot \mathbf{b} \) as well as their vector product \( \mathbf{a} \times \mathbf{b} \) as output on the screen. (Comment, compile, and test it, too!)

3. **Unit Conversion**
   The conversion factor between inches and centimeters is 1 in = 2.54 cm. Also, one foot is 12 inches. Implement (test, compile) C++ code that takes your height in centimeters as keyboard input, and delivers your height in feet and inches on the screen. (*Example:* My height is 183.5 cm, or 6 ft., 0.25 in.)

4. **Greatest Common Divisor**
   In the lecture, we discussed an algorithm that calculates the greatest common divisor (GCD) of two integer numbers. Construct a C++ program that accepts two numbers as keyboard input, calculates their GCD, and prints it on the screen, Comment, build, and test it.

5. **Prime Numbers**
   On the back of this page you will find a C++ program that produces a list of prime numbers.

   (a) What exactly does this program accomplish, and how does it go about it? Analyze the source code, and comment on the algorithm used. Can you think of ways of improving it?

   (b) The distribution of prime numbers among the integers is a central subject of number theory. A rather mysterious problem in this field is the distribution of prime number twins, pairs of prime numbers \( \{p, p + 2\} \) that differ only by 2 (e.g., 3 and 5, 5 and 7, 11 and 13, but also 99989 and 99991). It is not even known whether there are infinitely many of these pairs. — Modify the C++ program on the back of the page as to count and return a list of all prime number twins not exceeding some given number. Comment on your changes.
/* The FindPrimes Program */
#include<iostream>
using namespace std;

int main()
{
    /* Upper Limit */
    long Limit;
    cout << "Upper Limit For Prime Search: ";
    cin >> Limit;

    /* Number of Primes */
    long Counter = 0;

    /* Current Number Examined */
    long Current = 1;

    /* Announce Prime Output */
    cout << "Primes: ";

    /* Loop Through Values */
    do
    {
        /* Increment Current Number */
        ++Current;

        /* Initialize Factor Variable */
        long Factor = 2;

        /* Initialize Flag */
        bool IsAPrime = true;

        /* Check Whether Current Number Is Prime */
        while ((Factor * Factor <= Current) && (IsAPrime == true))
        {
            if ((Current % Factor) == 0)
            {
                /* Number Was Evenly Divisible */
                IsAPrime = false;
            }
            else
            {
                /* Check Next Number */
                ++Factor;
            }
        }

        /* Count and Output Prime */
        if (IsAPrime == true)
        {
            cout << Current << " ";
            ++Counter;
        }
    }
    while (Current < Limit);

    /* Return Number of Primes */
    cout << "\n\nTotal Number of Primes Not Exceeding " << Limit << ": " << Counter << "\n";

    /* Terminate Program */
    return 0;
}