Programming Assignment #2 (due 23:59:59 10/28, 2015)

Problem: Polynomial Greatest Common Divisor

The greatest common divisor (GCD) for two or more univariate polynomials is important and elementary in algebra. For example, polynomial GCD can be used in root-finding algorithm when polynomials have multiple roots. Further, GCD computations allow to compute the square-free factorization, even for multivariate polynomials. In computer algebra systems, most of the modern theory of polynomial GCD has been developed to satisfy the need of efficiency.

In this assignment, you need to write a C++ programming to derive the GCD of two univariate polynomials and return the GCD.

Provided files: (1) main.cpp, (2) GCD.cpp, (3) GCD.h, (4) example

- **main.cpp** – It executes the function **FindGCD**() and checks the answer. It can be changed if necessary for you to debug.
- **GCD.cpp & GCD.h** – These are program files you need to implement.
- **example** – it is an exemplary input test case accompanied with answers, which can be used to test your program.

INPUT

An **example** sequence is shown below:

```
x^3-1                    ←1st input1
x^2-2x+1                 ←1st input2
x-1                      ←1st answer
x^2+x+1                  ←2nd input1
x-1                      ←2nd input2
1                        ←2nd answer
…
```

The polynomial $A_0x^0 + A_1x^1 + A_2x^2 + A_3x^3 + \cdots + A_{998}x^{998} + A_{999}x^{999}$, is stored in an **long long integer** array with size=1000 as follows:

array[0]	array[1]	array[2]	array[3]	…	array[998]	array[999]
A_0	A_1	A_2	A_3	…	A_{998}	A_{999}

The two inputs and answer are constructed as this structure, and the output of FindGCD() should also be the same structure with the same array size.

Note that each of the two inputs would not be constant **0** and is smaller than **INT_MAX**, but the coefficients may exceed **INT_MAX** in your calculation.

Also, the returned GCD should be the **polynomial with least integer coefficients**, and the coefficient of the largest power should be **positive**. If there is no GCD between two inputs, which also means GCD is a constant A_0 ,you should return **1**.

Example:

GCD you calculate	GCD you should return
`3x^3-6`	`x^3-2`
`-3x^3+6`	`x^3-2`
`6`	`1`

Language

C or C++

Platform

You may develop your software on UNIX/Linux.

Compile: $ g++ main.cpp GCD.cpp

Execution: $ ./a.out

Submission

Please update the following materials to **E3** website by the deadline.

(1) GCD.h

(2) GCD.cpp

Grading

(1) example correct: 60%

(2) hidden cases correct: 10%

(3) hidden cases ranking: 30% (ranking: run time only)