COM 5336 ASSIGNMENT #1

DUE BY 11:59PM **3/8**/2017 (Wed)

10% penalty applies to 1-day late submissions received between 12:00 AM 3/9 and 11:59PM 3/9.
No submission will be accepted after 12:00 AM 3/10/2017

Objective

Implement big number arithmetic.

Description

Big number arithmetic: The easiest way is to use C++ and define a class for big numbers. You can overload all arithmetic operations to make your program nice and easy to understand. If you do not wish to use C++, you can use any structure similar to the **`struct`** in C.

Below is an example of C++ class definition. It is recommended to do everything in hex due to its simplicity. Moreover, when you implement modular exponentiation you'll find it very easy to manipulate.

```
//Big number class definition. This is just an example.
class BigNumber{
 private:
   bool sgn;
   unsigned int num_of_bits;
   uint8_t *data;

 public:
   //constructors
   BigNumber();
   BigNumber(int);  //directly convert from an int
   BigNumber(bool, unsigned int, uint8_t*);

   //overloaded arithmetic operators as member functions
   BigNumber operator+(BigNumber);
   BigNumber operator-(BigNumber);
   BigNumber operator*(BigNumber);
   BigNumber operator/(BigNumber); //integer division: 3/2==1
   BigNumber operator%(BigNumber);

   //interface functions
   void Print();
   void GetData(bool& ,unsigned int& , uint8_t*);
};
```

Sample I/O (`Input shown in` **`bold face`**.)

```
a= f1245ab3341ff34618188817676819ee
b= ffa24387539639853800bbecbcb494990
a+b = 1011e7eeba95956de6b9893d63332b1637e
a-b = e12a367abee68fadc4987c589b9c1ed05e
a*b = f0cc0ef5e2f7d593719ce283c6efb373d86a14d50f9f5c5
ba42a6bae39ff8d173e0
a/b = f
a%b = 17c3b6455c31d593397d7e9767e1cca7e
```

Grading

Your program MUST BE compatible with Dev C/C++ or GNU C/C++ compilers. If you are using other compilers, please make sure your final program is compatible. **You will get no points if your program is not compilable using the abovementioned compilers.** If your program is compilable but the result is not completely correct, you'll still get partial credits. Your program should be well-commented, well-structured, and easy to understand. You may lose up to 30% of points if you fail to do so.

Submission

Put all your source codes in a folder containing main functions, function implementations, class definitions, or compilation instructions, if any. Compress them as a single zip file. DO NOT submit executable files. Name your zip file as your student ID number (i.e. 100012345.zip). Submit your source code on iLMS at http://lms.nthu.edu.tw.