

# COM 5335 ASSIGNMENT #1

DUE AT 11:59PM **3/5/2023** (Sun)

10% penalty will be applied to late submissions received between 0:00 AM 3/6 and 11:59PM 3/6, 20% penalty will be applied to late submissions received between 0:00 AM 3/7 and 11:59PM 3/7, and 30% penalty will be applied to late submissions received between 0:00 AM 3/8 and 11:59PM 3/8. No submission will be accepted after 0:00 AM 3/9/2023.

## Objective

Implement big number arithmetic in hexadecimal representation.

## 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. You can use any programming languages that **do not support big number**, (e.g., **you cannot use python to do this assignment!**).

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

```
a= f1245ab3341ff3461818881767676819ee
b= ffa24387539639853800bbebcb494990
a+b = 1011e7eeba95956de6b9893d63332b1637e
a-b = e12a367abee68fadc4987c589b9c1ed05e
a*b = f0cc0ef5e2f7d593719ce283c6efb373d86a14d50f9f5c5
ba42a6bae39ff8d173e0
a/b = f
a%b = 17c3b6455c31d593397d7e9767e1cca7e
```

For simplicity, both a and b in the **input** are **assumed to be positive**. If  $a < b$ , then  $a - b$  is negative, and **the negative sign is added to the output** (e.g.  $-e12\dots$ ). All other operations have positive sign.

## 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 eLearn at <http://elearn.nthu.edu.tw>.

## An Example of Big Number Implementation

Below is just an example of C++ class definition for your reference. **You are NOT REQUIRED to implement in this way.** (下面的 C++ 程式碼僅是範例供參考用，本作業沒有要求一定要這樣寫!) 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*);
};
```