COM 5336 ASSIGNMENT #2

DUE BY 11:59PM **3/31**/2016 (Thu)

10% penalty applies to 1-day late submissions received between 11:59PM 3/31 and 11:59PM 4/1.
No submission will be accepted after 11:59PM 4/1/2016

Objective

Implement the Advanced Encryption Standard (AES).

Description

First, implement GF(256) as follows. We use 8 bits to represent a polynomial of degree at most 7 as explained in the class. We can also use 8 bits to represent a monic polynomial of degree 8. For example, $m(x)=x^8+x^4+x^3+x+1$ can be represented as 0x1b. Represent GF(256) as F2[x]/m(x). Implement the following field operations as C/C++ functions:

```
uint8_t GF256_add(uint8_t a, uint8_tb, uint8_t mx);
// returns a + b. mx is the irreducible polynomial

uint8_t GF256_mult_x(uint8_t a, uint8_t mx);
// Multiplied by x. mx is the irreducible polynomial

uint8_t GF256_mult(uint8_t a, uint8_t b, uint8_t mx);
// General multiplication: mx is the irreducible polynomial

uint8_t GF256_inv(uint8_t *a, uint8_t mx);
// Returns the multiplicative inverse of a. mx is the irreducible polynomial
```

Implement AES as follows.

```
void AES_Encrypt(uint8_t* Plaintext, uint8_t* Ciphertext, uint8_t* Key);
void AES_Decrypt(uint8_t* Plaintext, uint8_t* Ciphertext, uint8_t* Key);
```

Write a main function that calls `AES_Encrypt()` and `AES_Decrypt()`. Show the (intermediate) state of each round as well as the final result in hex. Note that you MUST NOT use table look-ups for SubBytes. Compute the multiplicative inverse by calling GF256_inv().

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.

Sample Input Subroutine Implementation

```
void input(uint8_t* input){
   uint8_t temp[49];
   int ch,i,c;
   int temp1[49];
   for(c=0;c<48;c++)
       temp[c]=getchar();                  //collect input(hex)
   for(c=0;c<48;c++){                      //transform to decimal
       if(temp[c] >= '0' && temp[c] <= '9')  temp1[c]=    temp[c]-'0';
       if(temp[c] >= 'a' && temp[c] <= 'f')  temp1[c]= 10+temp[c]-'a';
       if(temp[c] >= 'A' && temp[c] <= 'F')  temp1[c]= 10+temp[c]-'A';
       for(i=0;i<16;i++)
           input[i]=((16*temp1[3*i])+temp1[3*i+1]);     //transform to binary
   }
}
```