1. Determine the following in GF(13).

GF(13)

Z	0	1	2	3	4	5	6	7	8	9	10	11	12
AI	0	12	11	10	9	8	7	6	5	4	3	2	1
MI	-	1	7	9	10	8	11	2	5	3	4	6	12

(a) $(7x^4+3x^3+x^2+10)-(9x^4+6x^3+7x^2+8x+2)$

$$\boxed{=(11x^4+10x^3+7x^2+5x+8)}$$

(b) $(7x^3+2x+9)\times(2x^3+x^2+8x+7)$

$$\left[(14x^6+7x^5+56x^4+49x^3)+(4x^4+2x^3+16x^2+14x)+(18x^3+9x^2+72x+63)\right](\text{mod } 13)$$

$$=(1x^6+7x^5+4x^4+10x^3)+(4x^4+2x^3+3x^2+x)+(5x^3+9x^2+7x+11)$$

$$\boxed{=x^6+7x^5+8x^4+4x^3+12x^2+8x+11}$$

(c)

$$\begin{array}{r|l} & 4x^2+9x \\ \hline 3x^3+4x^2+3 & 12\cancel{x}^5+4x^4+36x^3+12\cancel{x}^2+x \\ & 1\cancel{2}x^5+3x^4+0^3+1\cancel{2}x^2+0 \\ \hline & x^4+10x^3+x \end{array}$$

① $\dfrac{12x^5}{3x^3}=12\times3^{-1}=(12\times9)\%13=4$

② $1\times3^{-1}=1\times9=9$

$(27x^4+36x^3+27x)\ \%13$

$\downarrow$

$$\begin{array}{r} =\cancel{x}^4+\cancel{10}x^3+\cancel{x} \\ \cancel{x}^4+1\cancel{0}x^3+\cancel{x} \\ \hline 0 \end{array}$$

$$\boxed{\therefore 4x^2+9x}$$

2. For the finite field $GF(2^3)$, calculate the following for the modulus polynomial x^3+x+1.

(a) $(x^2+x+1)\times(x+1)$

$=x^3+x^2+x^2+x+x+1$

$=\cancel{x^3}+2\cancel{x}^2+2\cancel{x}+\cancel{1}\otimes(\cancel{x^3}+x+\cancel{1})$

$$\boxed{=x}$$

(b) $\left[(x+1)-(x^2+x+1)\right]\otimes(x^3+x+1)$

$$\boxed{=-x^2=x^2}$$

(c)

$$\begin{array}{r|l} & 1 \\ \hline x^2+1 & x^2+x+1 \end{array}$$

① Find MI of $x^2+1 \rightarrow (111)\cdot(010)$

$=x$

$\left[(x^2+x+1)\times(x)\right]\otimes(x^3+x+1)$

$=(\cancel{x^3}+x^2+\cancel{x}+0)\otimes(\cancel{x^3}+\cancel{x}+1)$

$=0+x^2+1$

$$\boxed{=x^2+1}$$

Ecryption explanation:

1. **Create subbytes and inverse subbytes table from given code in lecture**
2. **Generate round keys from given code**
   A. **Starting off, we read in 256 bit key and we arrange the key into a bite block where the first four bites are the entries in the first column, which also equals the first word word[0]. There will be 60 words for 256 bit key.**
   B. **Using this, we then create 14 round keys for each round of processing within the encryption algorithm itself.**
3. **The very first step in encryption is by reading in 128 bit block from the input message.txt file, and casting it as a BitVector. We will XOR this 128 bit long block with our first round key as part of encryption.**
4. **Using the code provided in lecture, we will change this 128 bit long block into a state array which is 4 x 4. Many a times during this process, we will change the data type of this 4 x 4 state array to either hex values or bit vector values depending on the modification we want to make it to the state array.**

Round:

1. **Starting off we will perform a byte substitution for our state array.**
   A. **This is Don by casting our state array to a 4x4 bit vector and, for each byte in the array, we split it into two halves - the first half being a row index and the second half being a column index. Using these to index the 16 by 16 sub bytes array.**
2. **Next, we will rotate our state array**
   A. **The first row in the array will remain the same**
   B. **Second will rotate to the left by one**
   C. **Third will rotate tot he left by two**
   D. **Fourth will rotate to the left by three.**
3. **Next we will mix the columns,**
   A. **The first column will be the XOR addition of 0x02 multiplied with the firstEntry in the first row with respect to the multiplicative inverse vis-a-vis the AES_modulus given in finite field GF(2^3). The second entry will be XORed with 0x03.**
   B. **Each time for each column, the multiplication with the hexadecimal values will clockwise rotate for all four columns.**
4. **Finally we exclusive or with the next round key**
5. **The final round does not include the mixing of columns**

Decryption:

1. **The steps for decryption stay the same as encryption except the ordering is different and slight variations are made in the functions themselves**

2. Inverse sub bites is a table made from the given a code and that is used for the substitution of bites in the same way that the encryption works.
3. Mix columns has values that include 0x09, 0x0B, 0x0D, 0x0E.
4. The shifting of rose is the same as encryption but the direction is changed.

The ordering for decryption is as follows

1. **Round key adding**
2. **Inverse row shift**
3. **Inverse sub bytes**
4. **Inverse column mix**