1) MixColumns Pseudocode

```
void mixColumns(byte state[4][Nb]) {

    byte POLY_COLS[4][4] = {
        {0x02, 0x03, 0x01, 0x01},
        {0x01, 0x02, 0x03, 0x01},
        {0x01, 0x01, 0x02, 0x03},
        {0x03, 0x01, 0x01, 0x02},
    }
    byte temp[4][Nb]
    byte result

    temp = state

    /*
        For each element in the state, set it equal to
        the dot product between the current row from the state
        and the column from the POLY_COLS matrix
    */
    for row in rows[4] {
        for col in cols[4] {
            result = 0
            for factor in elements[4] {
                result = result ^ ffMultiply(temp[factor][row], POLY_COLS[col][factor])
            }
            state[col][row] = result
        }
    }
}
```

2) ffMultiply Pseudocode

```
byte ffMultiply(byte a, byte b) {
    byte r = 0x00, t = 0x00

    for bit in bits[8] {
        /*
            If the current bit at index 'bit' is 1,
            add (XOR) a with the result
        */
        if (b & (1 << bit)) {
            r =  r ^ a;
        }
        /*
            Multiply a by shifting it over 1 to the
            left and then adding (XOR) the result with
            the modulus (0x1b) if the 7th bit
            in a is one before the bit shift
        */
        t = a << 1;
        a =  (a & 0x80) ?  t ^ 0x1b : t
    }

    return r
}
```