Anthony Constantino
HW 2

1) Psuedocode for a Mix Columns algorithm

```
def mixColumns():
    Create a copy matrix m representing the current state.
    for each row:
        for each column:
            setup matrix m with values from original matrix o

    Create a matrix c with these values
    0x02, 0x03, 0x01, 0x01
    0x01, 0x02, 0x03, 0x01
    0x01, 0x01, 0x02, 0x03
    0x03, 0x01, 0x01, 0x02

    for each column:
        m[0][column] = finiteFieldMultiply(o[0][column], m[0][0]) ^ ... ^
        finiteFieldMultiply(o[3][column], m[0][3])

        m[0][column] = finiteFieldMultiply(o[0][column], m[1][0]) ^ ... ^
        finiteFieldMultiply(o[3][column], m[1][3])

        m[0][column] = finiteFieldMultiply(o[0][column], m[2][0]) ^ ... ^
        finiteFieldMultiply(o[3][column], m[2][3])

        m[0][column] = finiteFieldMultiply(o[0][column], m[3][0]) ^ ... ^
        finiteFieldMultiply(o[3][column], m[3][3])

    for each row:
        for each column:
            copy the fields of new matrix m into original matrix o
```

2) Psuedocode for a Finite Field Multiply algorithm

```
def finiteFieldMultiply(a, b): <-where a and b are ints

    solution = 0x00

    for each of the bits:
        check if we have an odd bit in b:
            solution = soluxtion XOR a

        check if a has last bit set:
            a = a shift bits left
            a = XOR 0x11b <- this will reset all the lower bits
        else:
```

a = a shift bits left

b = b shift bits right