```
'''
Homework Number:   1
Name: Sneha Mahapatra
ECN Login: mahapat0
Due Date: 01/23/20
'''
import os.path
from cryptBreak import *
from BitVector import *
PassPhrase = "Hopes and dreams of a million years"
BLOCKSIZE = 16
numbytes = BLOCKSIZE//8

if __name__ == '__main__':
    bruteForce()

def bruteForce():
    allPValues = tuple(range(0, 2 ** 16))
    for key in allPValues:
        plain = cryptBreak('encrypted.txt', key)
        if "Mark Twain" in plain:
            print("Encryption Broken!")
            print("Key: ",key)
            print("Message: ",plain)
            if os.path.isfile('decrypted.txt'):
                FILEOUT = open('decrypted.txt', 'w')  # (d)
                FILEOUT.write(plain)  # (e)
                FILEOUT.close()
            else:
                print("File decrypted.txt does not exist")
            break

def cryptBreak(ciphertextFile, key):
    FILEIN = open(ciphertextFile)  # (J)
    encrypted_bv = BitVector(hexstring=FILEIN.read())
    bv_iv = BitVector(bitlist=[0] * BLOCKSIZE)  # (F)
    for i in range(0, len(PassPhrase) // numbytes):  # (G)
        textstr = PassPhrase[i * numbytes:(i + 1) * numbytes]  # (H)
        bv_iv ^= BitVector(textstring=textstr)  # (I)
    key_bv = BitVector(bitlist=[0] * BLOCKSIZE)  # (P)
    key_bv = BitVector(intVal=key, size=16)
    msg_decrypted_bv = BitVector(size=0)  # (T)
    previous_decrypted_block = bv_iv  # (U)
    for i in range(0, len(encrypted_bv) // BLOCKSIZE):  # (V)
        bv = encrypted_bv[i * BLOCKSIZE:(i + 1) * BLOCKSIZE]  # (W)
        temp = bv.deep_copy()  # (X)
        bv ^= previous_decrypted_block  # (Y)
        previous_decrypted_block = temp  # (Z)
        bv ^= key_bv  # (a)
        msg_decrypted_bv += bv  # (b)
    outputtext = msg_decrypted_bv.get_text_from_bitvector()  # (c)
    return outputtext
```

It is my belief that nearly any invented quotation, played with confidence, stands a good chance to deceive.

- Mark Twain

Encryption Broken!
Key: 25202
Message: It is my belief that nearly any invented quotation, played with confidence, stands a good chance to deceive

For HW1 we have created a program that uses brute force attack to find the right key. The Brute Force attack will check through 2^16 key spaces. We checked through range(0,2^16) and then changed to a bit vector, used the decryption method given in DecryptForFun.py and checked whether the string "Mark Twain" appeared in the file.