Encryption Key: 0111010101100101

Decrypted Message:

Always go to other people's funerals, otherwise they won't go to yours.

- Yogi Berra

Code:

```
'''
Homework Number: 1
Name: Alex Goebel
ECN Login: goebel2
Due Date: 1/28/2021
'''

#!/usr/bin/env python3
import sys
from BitVector import *


def cryptBreak(ciphertextFile, key_bv):
    #All of the code in this function is from DecryptForFun.py as shown in Lecture 2. The entire function wasn't carried over as some of it didn't seem necessary
    PassPhrase = "Hopes and dreams of a million years"
    BLOCKSIZE = 16
    numbytes = BLOCKSIZE // 8

    #This loop reduces the passphrase to a bit array which of the size BLOCKSIZE
    bv_iv = BitVector(bitlist = [0]*BLOCKSIZE)
    for i in range(0, len(PassPhrase) // numbytes):
```

```
    textstr = PassPhrase[i*numbytes:(i+1)*numbytes]
    bv_iv ^= BitVector(textstring = textstr)

#This just opens the file and puts the contents into a variable as a BitVector
inFile = open(ciphertextFile)
encrypted_bv = BitVector(hexstring = inFile.read())

#This is just an empty BitVector used to store the decrypted message
decryptedMessage_bv = BitVector(size = 0)

previousDecryptedBlock = bv_iv

#This loop is what carries out the XORing of the bit blocks with the decryption
for i in range(0, len(encrypted_bv) // BLOCKSIZE):
    bv = encrypted_bv[i*BLOCKSIZE:(i+1)*BLOCKSIZE]
    temp = bv.deep_copy()
    bv ^= previousDecryptedBlock
    previousDecryptedBlock = temp
    bv ^= key_bv
    decryptedMessage_bv += bv

#Variable that is returned with the plaintext decrypted message
outputText = decryptedMessage_bv.get_text_from_bitvector()


inFile.close()
return outputText
```

```
if __name__ == '__main__':


    ciphertextFile = 'encrypted.txt'


    keyRangeMax = 2 ** 16


    #This loop goes through all of the keys in the range and sends each one to cryptBreak.
    #It then checks the message returned to see if it is the correct one.
    #If not, it continues with the next key until the correct message is found
    for i in range(28000, keyRangeMax):
        #The following line is from DecryptForFun.py from Lecture 2
        key_bv = BitVector(intVal = i, size=16)
        decryptedMessage = cryptBreak(ciphertextFile, key_bv)
        if 'Yogi Berra' not in decryptedMessage:
            print('Not decrypted yet')
            print(key_bv)
        else:
            print('Encryption broken!')
            print(key_bv)
            print(decryptedMessage)
            break
```

Code Explanation:

In my main function I have the brute force analysis where I go through a for loop that creates a BitVector object for each key option and sends that object, along with the input file, to cryptBreak. The cryptBreak function is mostly taken from the code given in DecryptForFun.py. It first uses a loop to get the passphrase into a bit array. Then, it uses the ciphertext file to create a BitVector. After, it creates a place to store the decrypted message. Finally, it uses a for loop which XORs the bit blocks with the decryption and outputs the decrypted message. My main file then checks to see if 'Yogi Berra' is part of that decrypted message and if it isn't, it tries the next key. If it is there, the loop ends and the key, message, and alert is printed to let the user know the encryption has been broken and what it says.