from cryptBreak import \*  
if \_\_name\_\_ == '\_\_main\_\_':  
 allBit = tuple(range(0, 2\*\*16))  
 def keygen(): # generates a key to try  
 for keyTry in allBit: #iterates through the set of keys  
 plain = cryptBreak('encrypted.txt', keyTry)  
 if "Mark Twain" in plain: #tests decoded message for correctness  
 print("Encryption Broken!")  
 print("Key: ",keyTry)  
 print("Message: ",plain)  
 FILEOUT = open('decrypted.txt', 'w') # (d)  
 FILEOUT.write(plain) # (e)  
 FILEOUT.close()  
 break

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