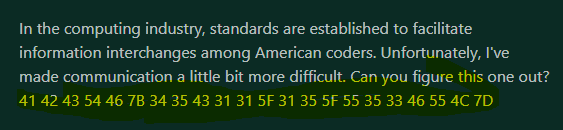
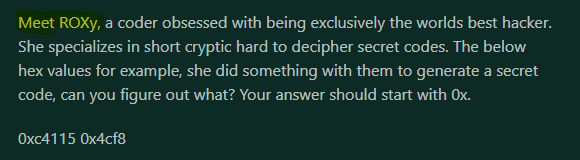
**CHARACTER ENCODING**



Convert hex to text, it’ll help you to find the anwser.

CTFlearn{45C11\_15\_U53FUL}

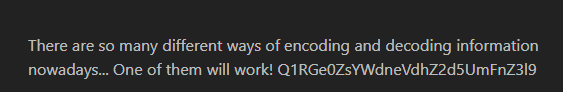
**HEXTROADINARY**



ROX mean XOR, you have to xor 0xc4115 ^ 0x3cf8 to find anwser

CTFlearn{0xc0ded}

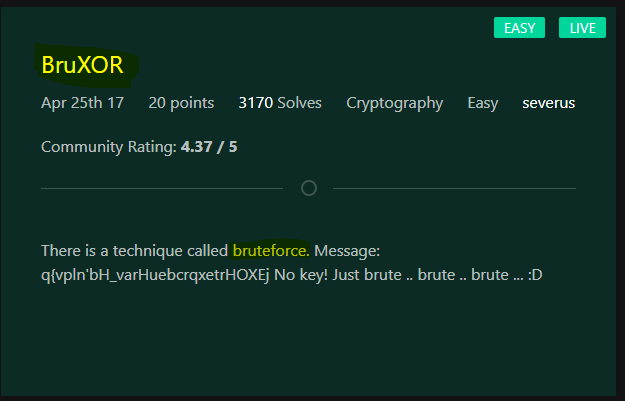
BASE 2 2 THE 6



You see, Q1RGe0ZsYWdneVdhZ2d5UmFnZ3l9 using 1-9 and a-z a think it’ll be Base64 and try decode it

CTFlearn{FlaggyWaggyRaggy}

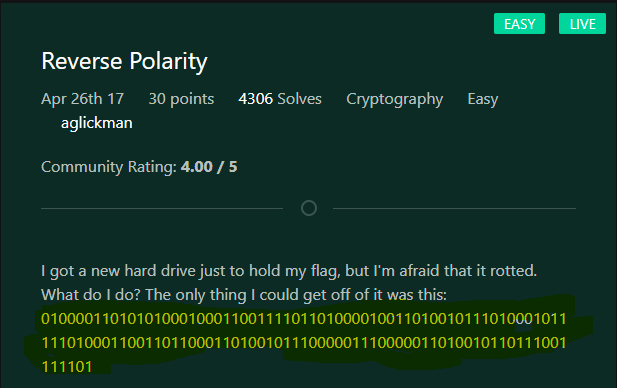
BRUXOR



You can see a title which help you explan to find flag

CTFlearn{y0u\_Have\_bruteforce\_XOR}

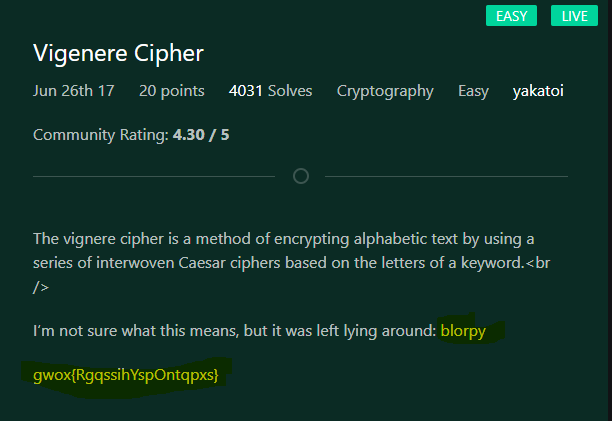
Reverse Polarity



You see that, convert binary to text. Easy 😊

CTFlearn{Bit\_Flippin}

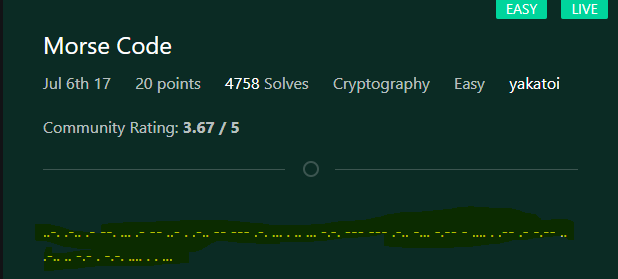
Vigenere Cipher



With key blorgy u can decode it

CTFlearn{CiphersAreAwesome}

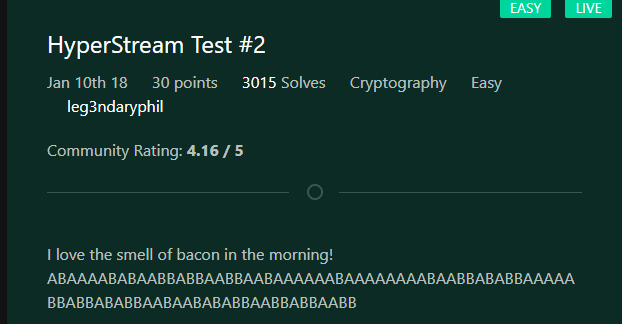
MORSE CODE



Mã morse là gì? <https://vi.wikipedia.org/wiki/M%C3%A3_Morse>

CTFlearn{SAMUELMORSEISCOOLBYTHEWAYILIKECHEES}

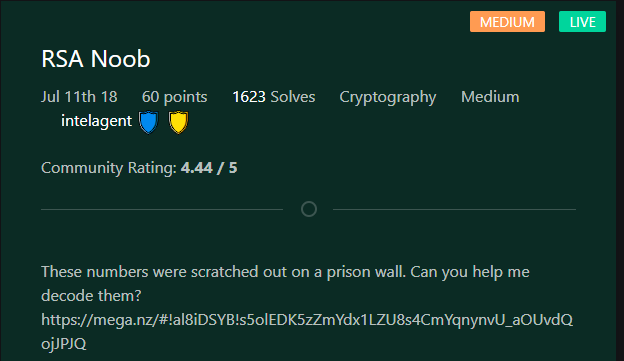
HYPERSTREAM



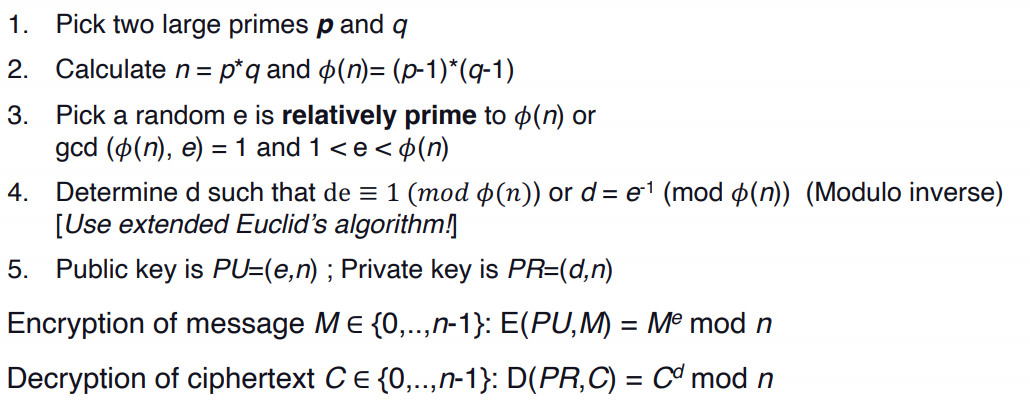
BACON decode

CTFlearn{ILOUEBACONDONTYOU}

RSA NOOD

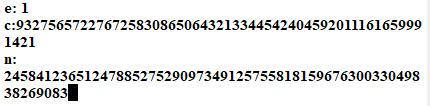


This challenge verry fun, it help you know about RSA work.



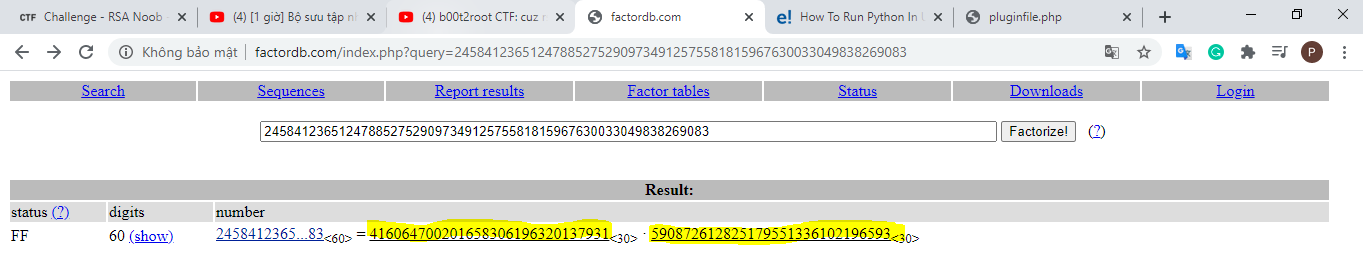
It’s talking about RSA.

In this challenge you have e,c,n



N = p\*q,

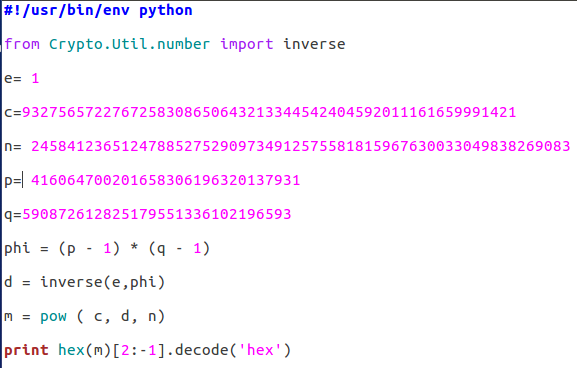
How to find p and q? using factordb.com



Show you p and q.

Sau khi có được p,q,e,n,c ta cần tìm thêm phi để tìm thêm d

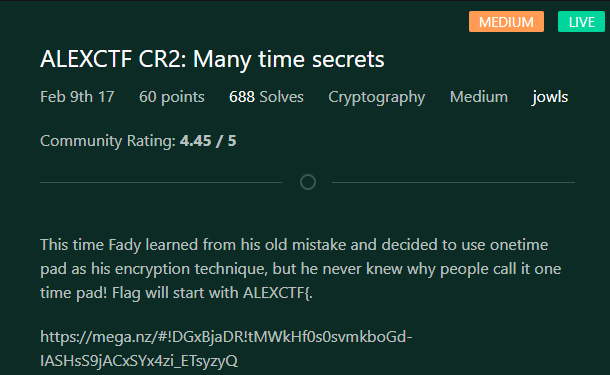
Sau khi có hết thì t sẽ giải mã bằng D = Cd mod n

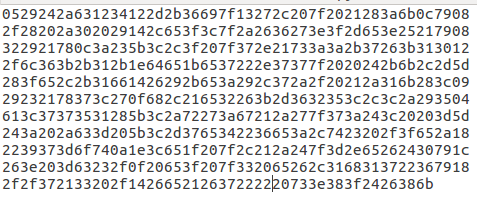


Viết ct python là xong, // print hex(m)[2: -1] nghĩa là bỏ đi 0x..

CTFlearn{b3tter\_up\_y0ur\_e}

ALEXCTF CR2: Many time secrets

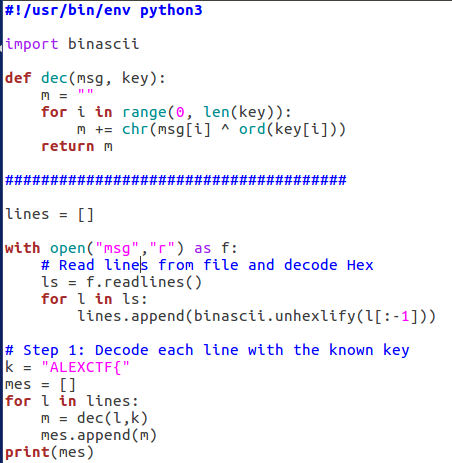




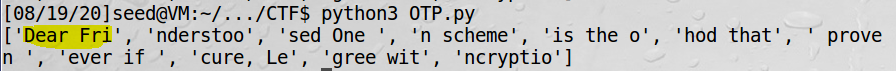
Mã hóa OTP xor plantext vs key, với key dài bằng plantext.

Ta có được đáp án bắt đầu với “ALEXCTF{“ thử xor xem ra gì(dựa vào tính chất đặc biệt của xor ta sẽ tìm key)

Ta viết chương trình python:

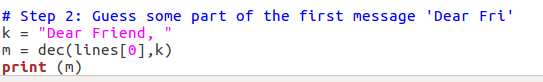


Chạy trương trình trên

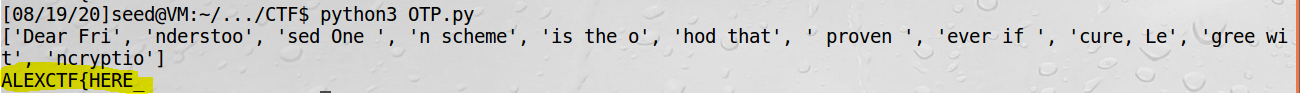


Ta thấy “Dear Fri” ta đoán này sẽ là “ Dear Friend,”

Thử dùng Key = Dear Friend,

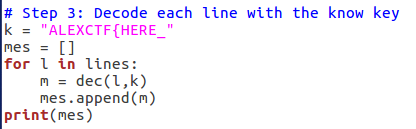


Chạy chương trình

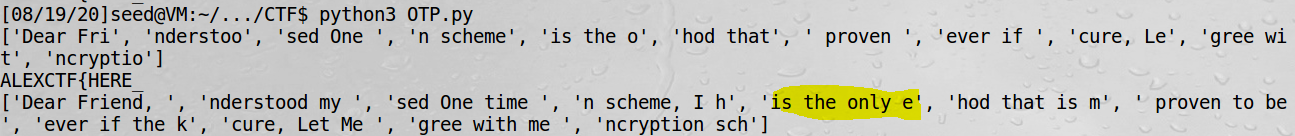


Chúng ta tìm được một chút manh mối ALEXCTF{HERE\_

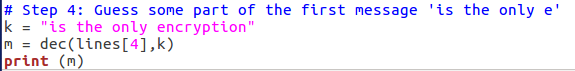
Đặt key = “ALEXCTF{HERE\_” để xem thêm manh mối.



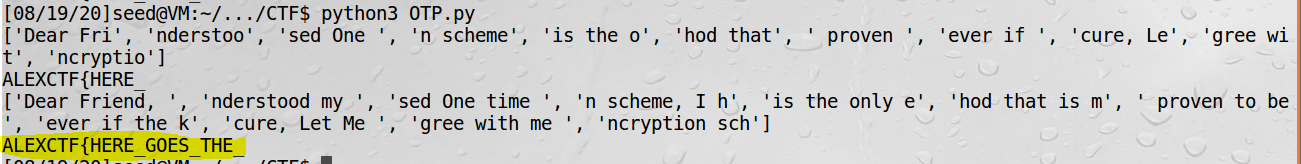
Chạy chương trình



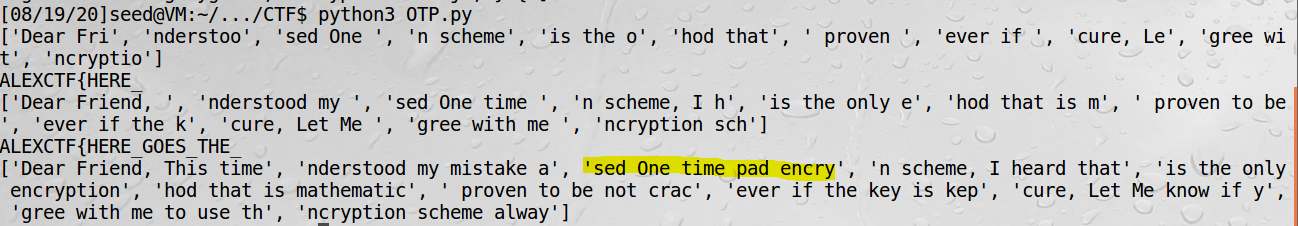
Chúng ta thấy “is he only e” tôi đoán là “is the only encryption”, hãy thử xem…



Chạy chương trình:



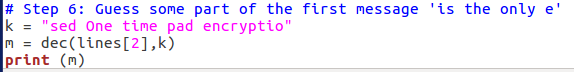
Oke chúng ta đã đúng, và lặp lại bước trên



Sed One time pad encry nó sẽ chắc chắn là Sed One time pad encryption

Let check it:

Chú ý vì phía trên mỗi hàng của file msg chỉ chứa 26 ký tự vì tính chất key của OTP phải bằng nên key của chúng ta là Sed One time pad encryptio



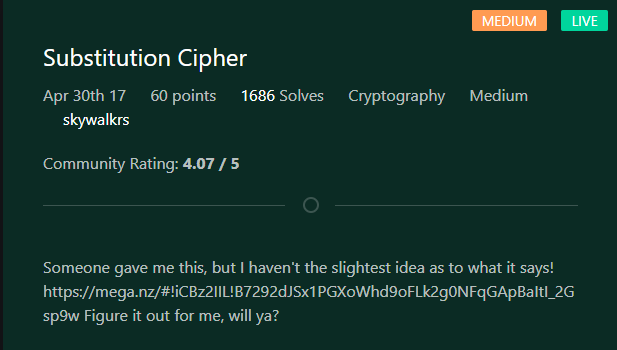
Chạy chương trình:

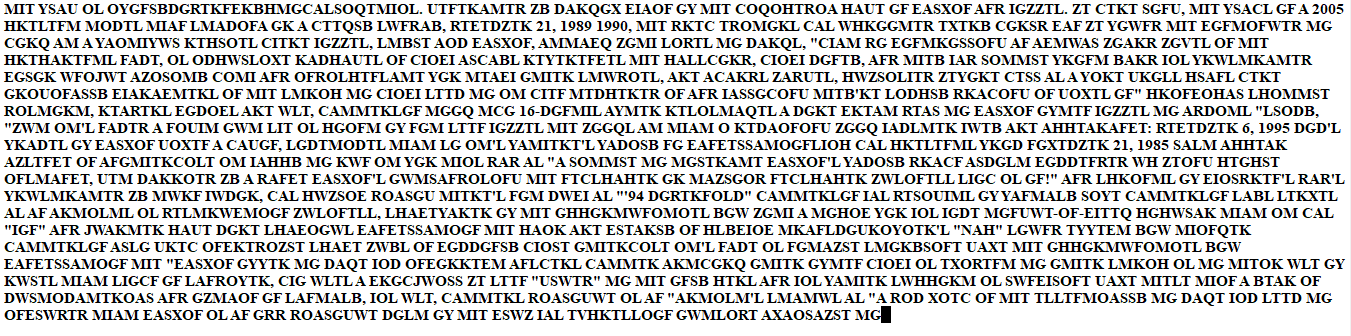


Oke xong

CTFlearn{HERE\_GOES\_THE\_KEY}

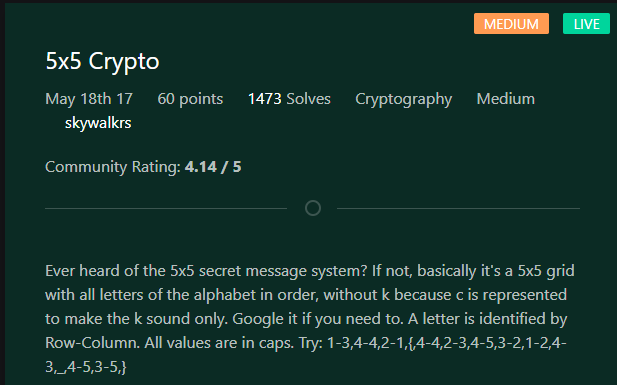
SUBSTITUTION CIPHER



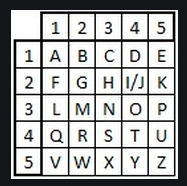


Nhìn đoạn mã hóa trên ta sẽ nghi là mã hóa monoalphabetic, dùng dcode.fr để giải auto.

5x5 crypto



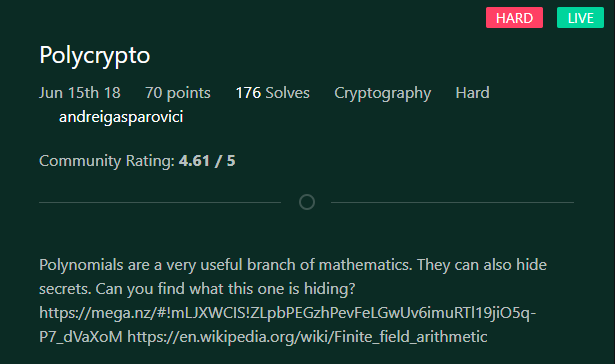
5x5 crypto is matrix 5x5

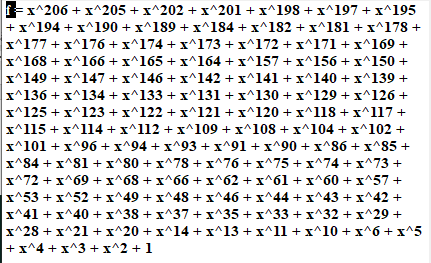


CTF{THUMBS\_UP}

* Flag = CTFlearn{THUMBS\_UP}

POLYCRYPTO

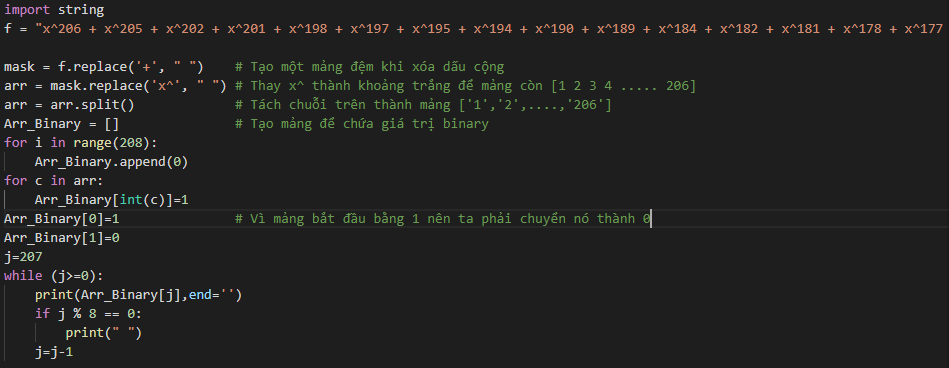




Ở bài này dùng đa thức để biểu hiện số bit

Vd như : x7+x4+x3+x+1 = 10011011

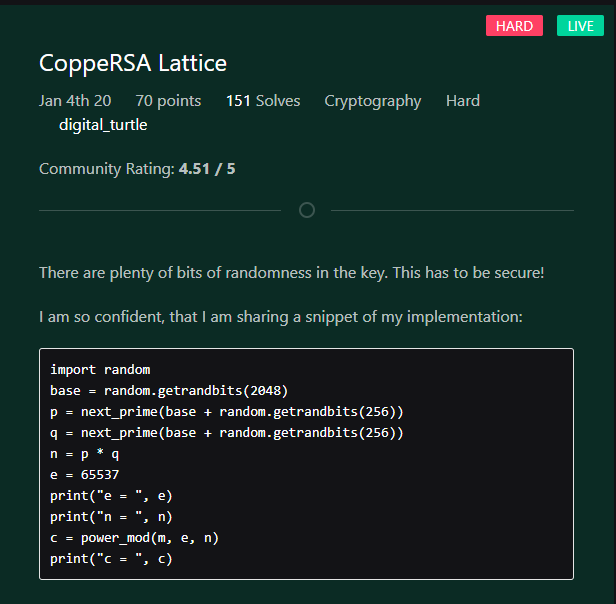
Từ ví dụ trên ta tìm đoạn binary cho hàm trên.



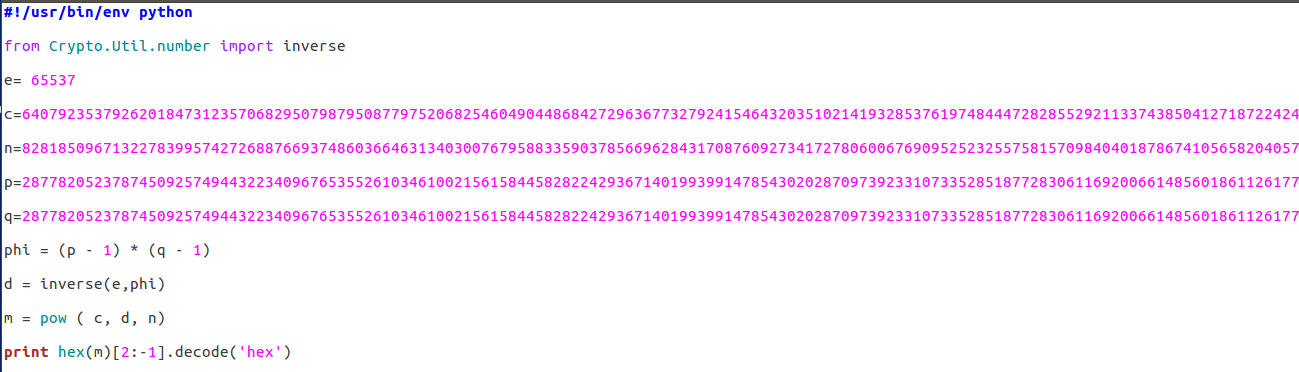
Ban đầu ta tạo một mảng gồm 208 phần tử (vì 208 % 8 =0) ta cho mảng đó bằng không hết, sau đó ta tạo vòng lặp với mỗi vị trí trong mảng nó sẽ là 1 còn lại là 0, và in ra mảng binary

Flag: CTFlearn{p0lynom1als\_4r3\_k00l}

COPPERSA LATTICE

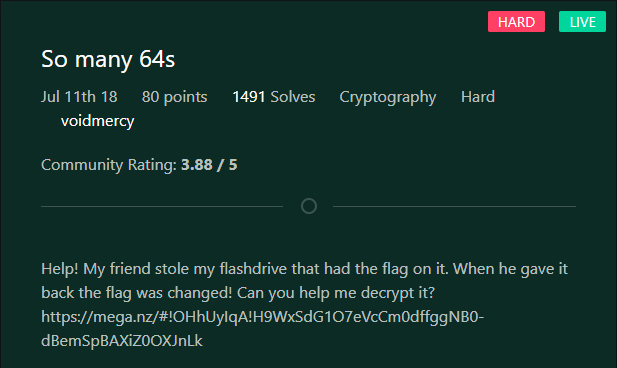


Bài này y chang bài RSA, kéo lên mà xem nhé!

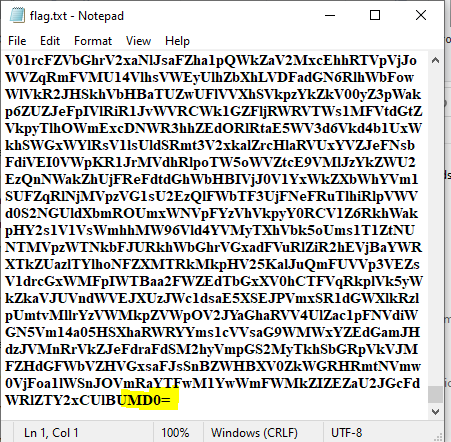


CTFlearn{n0t\_th4t\_s3cur3\_4ft3r\_4ll}

SO MANY 64S

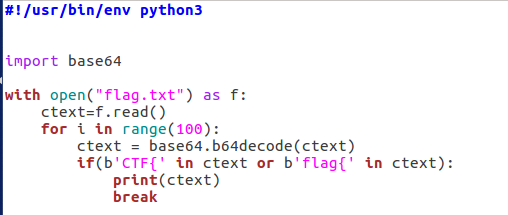


Khi mở file flag.txt ta thấy ở cuối dấu “=”



Nghi nó là base64.

Vì có nhiều kí tự nên ta không thể nào dùng tool trên web, vì nó là một loại mã hóa dễ vcl nên dùng code đi nhé đm,



Dùng vòng lặp lụi chứ hk biết bao nhiêu cho đủ nghen, nên đừng thắc mắc

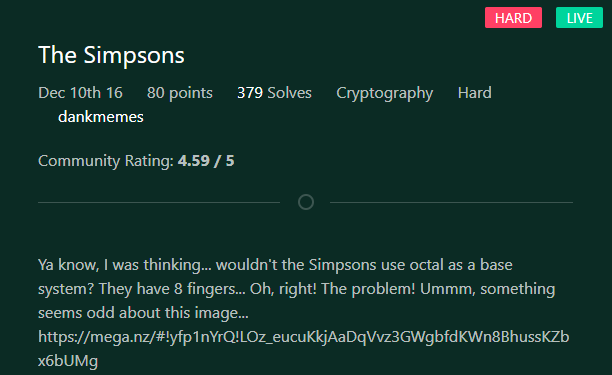
Ta có 2 cách đi một và decode sang file mới rồi Ctrl + F tìm

Hoặc dùng code tìm nhé.



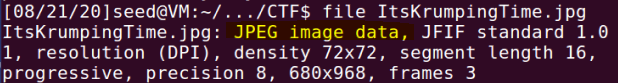
Flag 🡺 CTFlearn{pr3tty\_b4s1c\_r1ght?}

THE SIMPSON



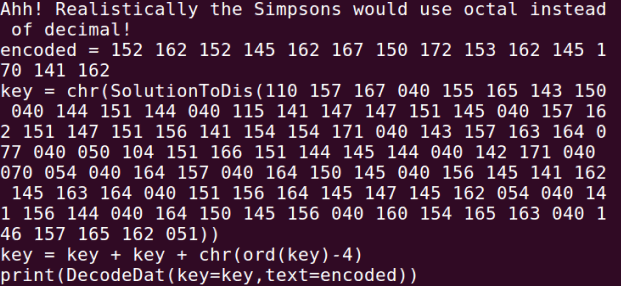


Ảnh trên là ảnh của gia đình simpsion



Ta dùng lệnh strings để xem bên trong ảnh có che dấu gì không

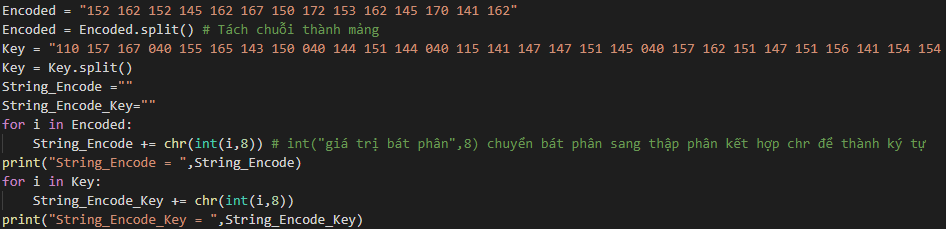
$ strings ItsKrumpingTime.jpg



Theo đề bài thì đây là hệ bát phân (octal) hãy em những con số đó có ý nghĩa gì?

Ta có thể dùng python hay dùng tool trên web,

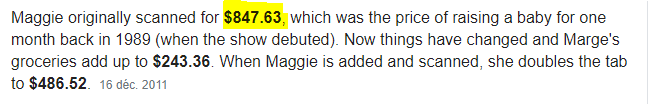
Chỉ cần dùng int(‘đoạn mã đó’, 8)



Code đi nó dễ lắm nghen



Theo như Encode\_key ta xem thử là Maggie bao nhiêu tiền nào. Search gg nhé!



847,63 / 8 =105,95375 ~ 106

106 + 4 =110

Ta xem lại ở phía trên: **Key = key + key + chr(ord(key) -4))**

Key = n + n + j = nnj

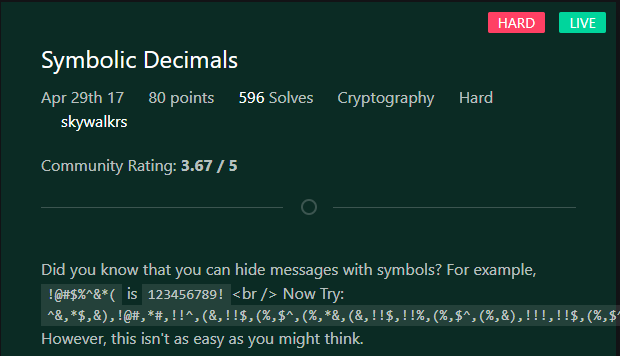
Encode= jrjerwhzkrexar

Oke key và cipher\_text vấn đề đây là mã hóa gì? Theo kinh nghiệp ít ỏi chơi CTF thì đoán đây là vigenere, cái này dùng tool trên web cho nhanh



Flag 🡺 CTFlearn{wearenumberone}

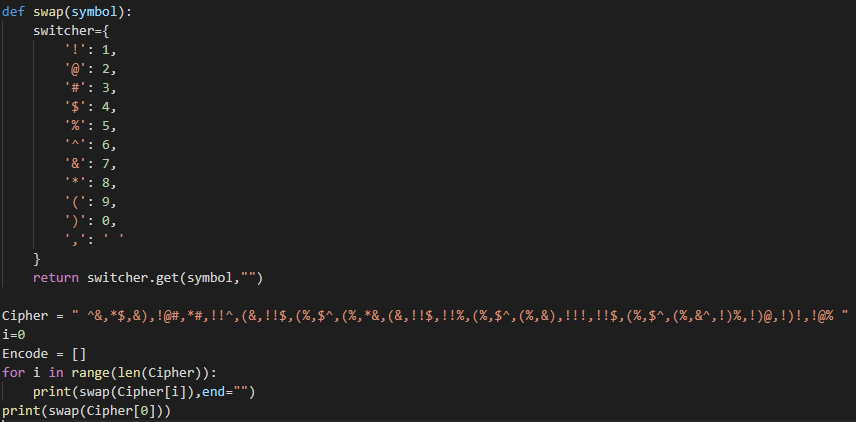
Symbolic Decimals



Đừng có nhìn thấy dòng cuối mà sợ, nó còn dễ hơn bạn nghĩ nữa 😊

!@#$%^&\*( là 123456789 thực chất nó là các kí tự đặc biệt trên bàn phím số khi nhấn shift là sẽ ra nhưng nhậy thì số 0 sẽ là”)”

Cũng chỉ là hoán đổi thôi mà,viết code đi ae

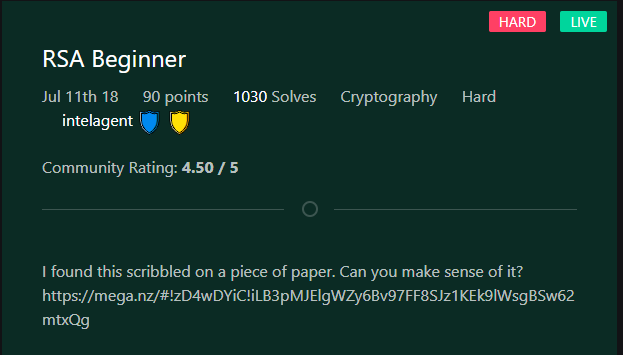




Nhìn phát biết ngay là dec nên chúng ta **convert dec to ascii** nào

Flag 🡺 CTFlearn{Star\_ . \_Wars\_ . \_For\_ . \_Life}

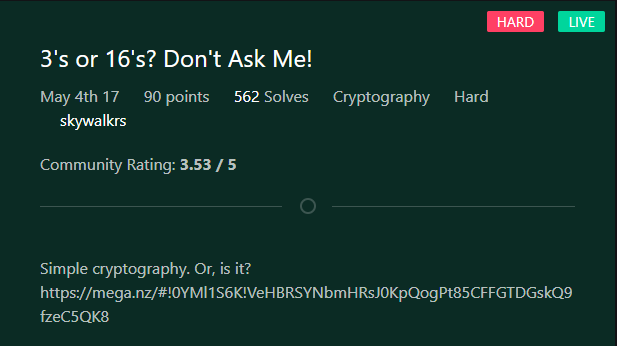
RSA BEGINNER

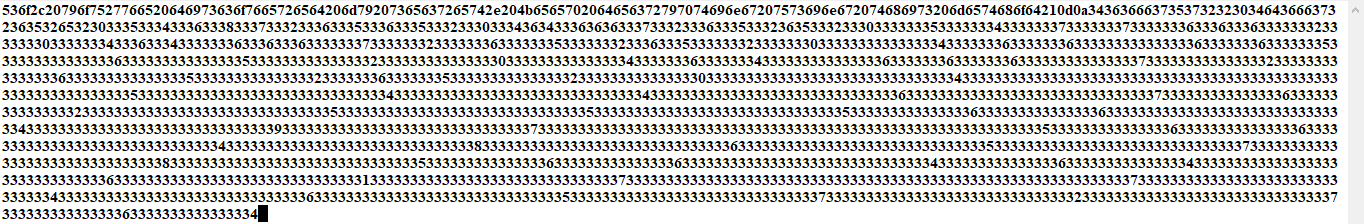


Một cái RSA mà 3 bài giống nhau, nên hk có write up nghen

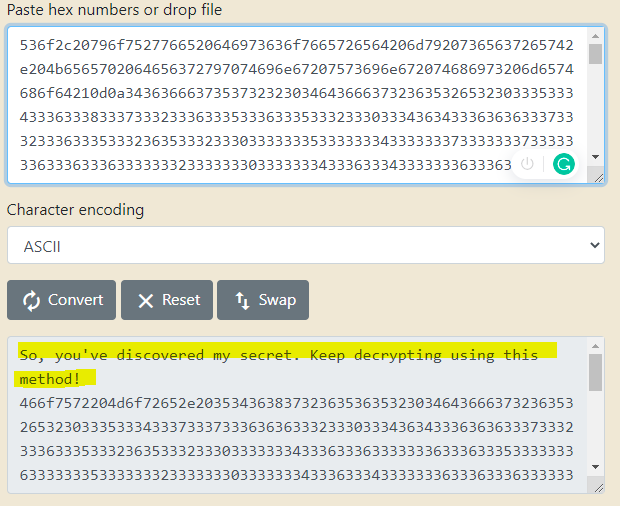
Flag 🡺 CTFlearn{rs4\_is\_aw3s0m3}

3’s or 16’s? Don’t ask me!

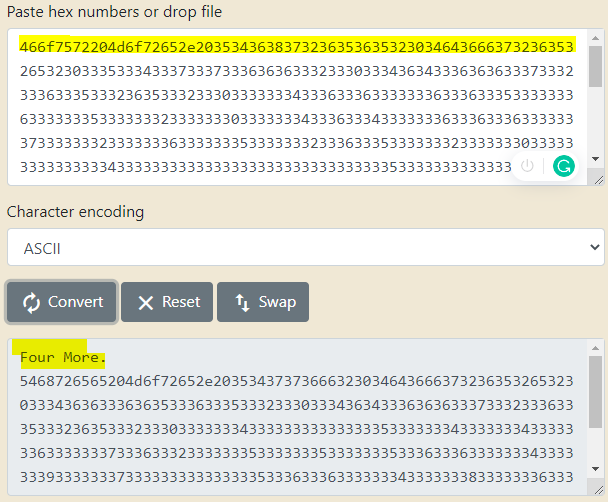




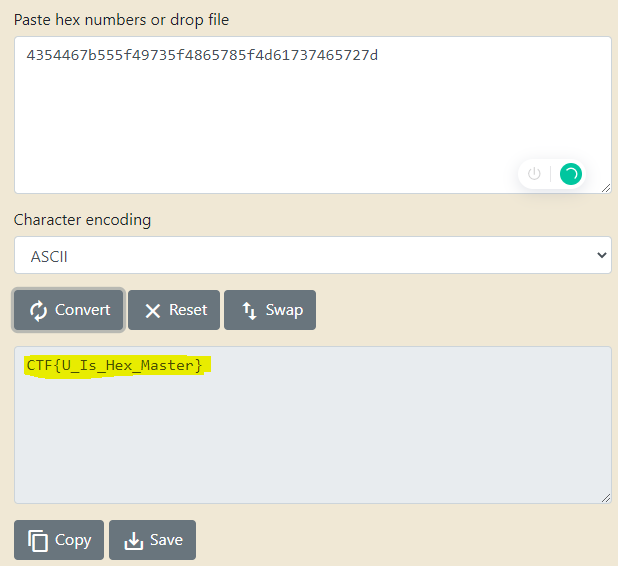
Nhìn vào dòng đầu tiên, la nghi đây là mã hex, vì chứa 1-9,a-f, thử chuyển dạng hex sang ascii, nếu viết code thì cũng đc dùng tool trên mạng cũng oke,



Tiếp tục dữ cách làm này, tiếp tục giải mã tiếp



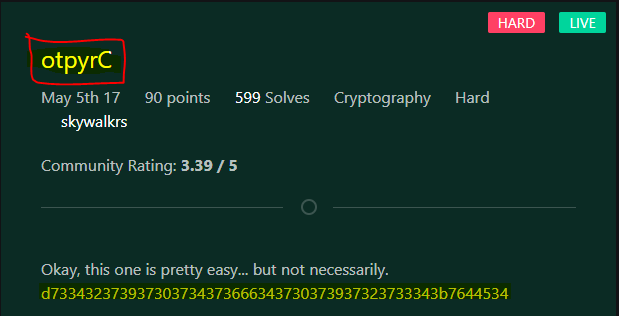
…….



Quá dễ đúng k?

Flag: CTFlearn{U\_Is\_Hex\_Master}

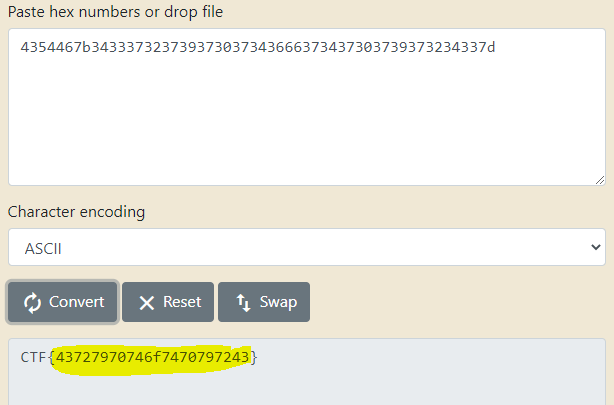
OTPYRC



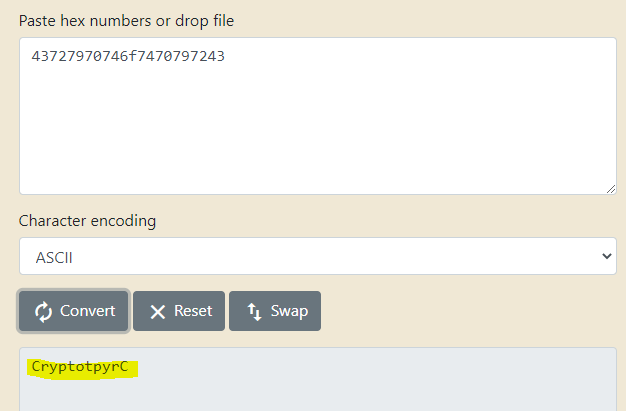
Bài này hk để ý được tiêu đề là sẽ toang

Bạn biết đấy, chữ Crypto bị đảo ngược thì nó gợi cho ta biết cái dòng dưới sẽ đảo được, dùng tool online reverse string: “4354467b343337323739373037343666373437303739373234337d”

Có được mã trên ta đem đi convert hex to ascii:



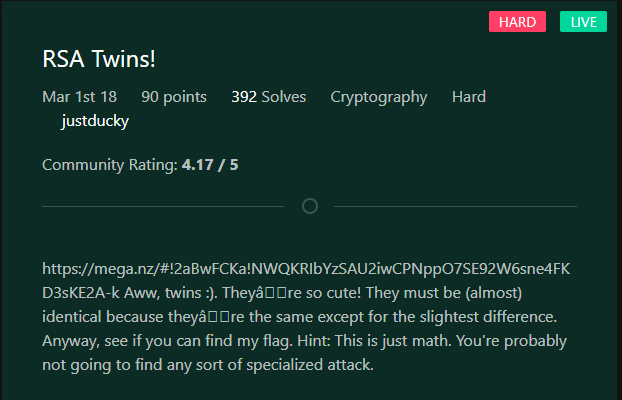
Chưa xong đâu cha, còn 1 lần nữa :v



Xong

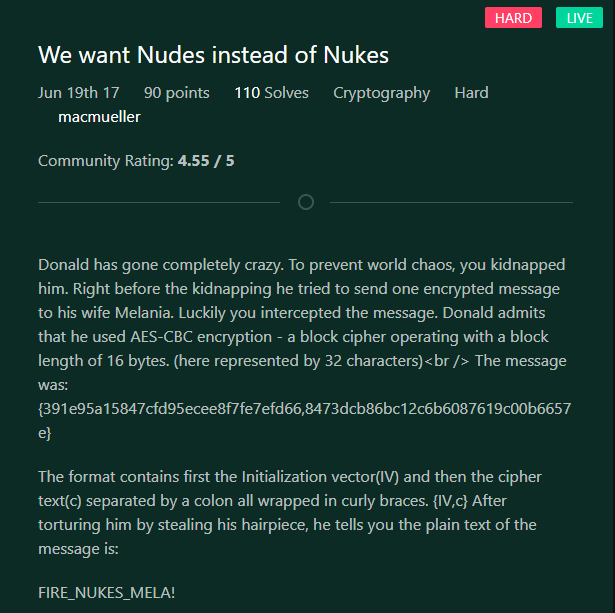
Flag 🡺 CTFlearn{CryptotpyrC}

RSA TWINS

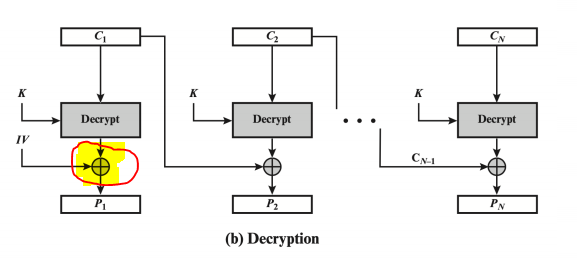


Flag 🡺 CTFlearn{i\_l0v3\_tw1N\_pr1m3s}

We want Nudes instead of Nukes



Ý tưởng chính của bài này IV, sau khi quá trình Decrypt vs Key thì sẽ xor với IV để lấy plantext. Chúng ta hãy dựa vào tính chất này để tấn công thay đổi plantext



Solution

Đây là điểm chúng ta tấn công. Sau khi thực hiện giải mã với Key thì ta sẽ thay đổi IV (Alter IV) sao cho Alter IV Xor Output = “SEND\_NUDES\_MELA!”

Hướng giải quyết

Để thực hiện chúng ta nhìn thấy được phép Xor ở đó, thì dựa vào tính chất đặc biệt của xor rồi chiến thôi.

A ^ B = C

A ^ C = B

B ^ C = A

Giải sử đoặt Output là kết quả khi Cipher text thực hiện Decrypt vs Key, theo bình thường thì Output = Plantext ^ IV đó là khi chưa có gì xảy ra.

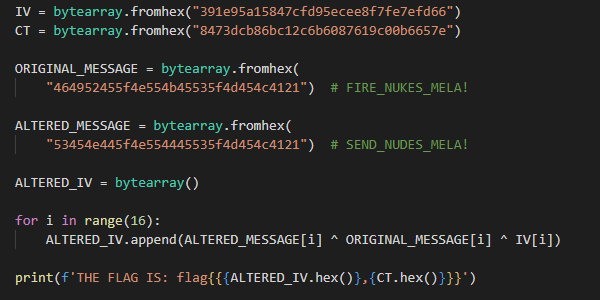
Chúng ta nhớ rằng A ^ A= 0.

Thay đổi IV dựa vào thính chất trên Alter IV = IV ^ Plantext ^ Alter Plantext

Giải thích Output sẽ chứa Plantext ^ IV vì thế khi Output ^ Alter IV thì sẽ chỉ còn Alter Plantext

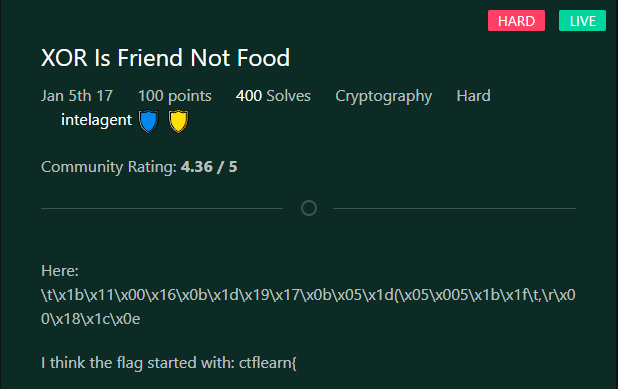
Dễ mà đúng không

Thực hiện trên code trên python

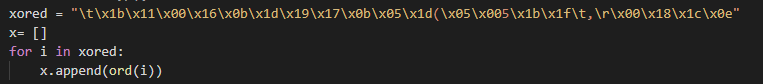


Flag 🡺 CTFlearn{2c1289a05847cfd65ecee8f7fe7efd66,8473dcb86bc12c6b6087619c00b6657e}

XOR IS FRIEND NOT FOOD



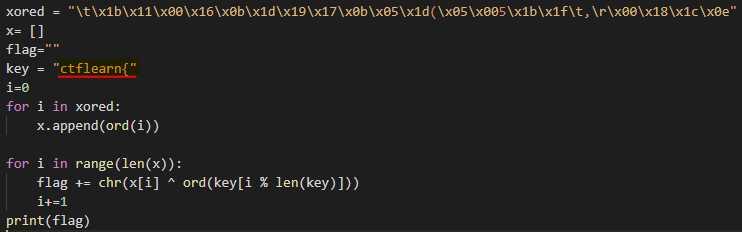
Bài trên thấy được format của xor ‘**/x<hex>**’ việc đầu tiên ta nên convert nó sang dạng mảng và đưa về bảng unicode cho dễ nhìn và tiện mắt



Và ta có được mảng mới.



Theo đề bài thì các kí tự đầu tiên sẽ là ‘ctflearn{‘ dựa vào đặc tính đặc biệt của xor thì ta tìm key

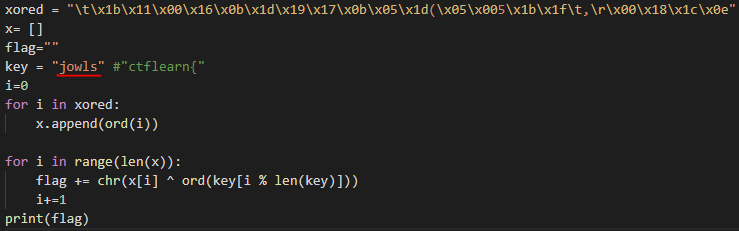


Và kết quả



Ta thấy jowls và sau đó lặp lại nghi đây là key ta thử xem sao

Đổi key = “jowls”



Flag 🡺 CTFlearn{xor\_is\_the\_goop}

Lagrange Polynomial

Tìm thấy được manh mối từ: <https://pdfs.semanticscholar.org/83e9/07d0df7064c04feea746af4937023721ea9c.pdf>

<http://vuontoanblog.blogspot.com/2012/10/polynomial-interpolation-lagrange.html>

<https://vi.wikipedia.org/wiki/M%C3%A3_h%C3%B3a_Huffman>