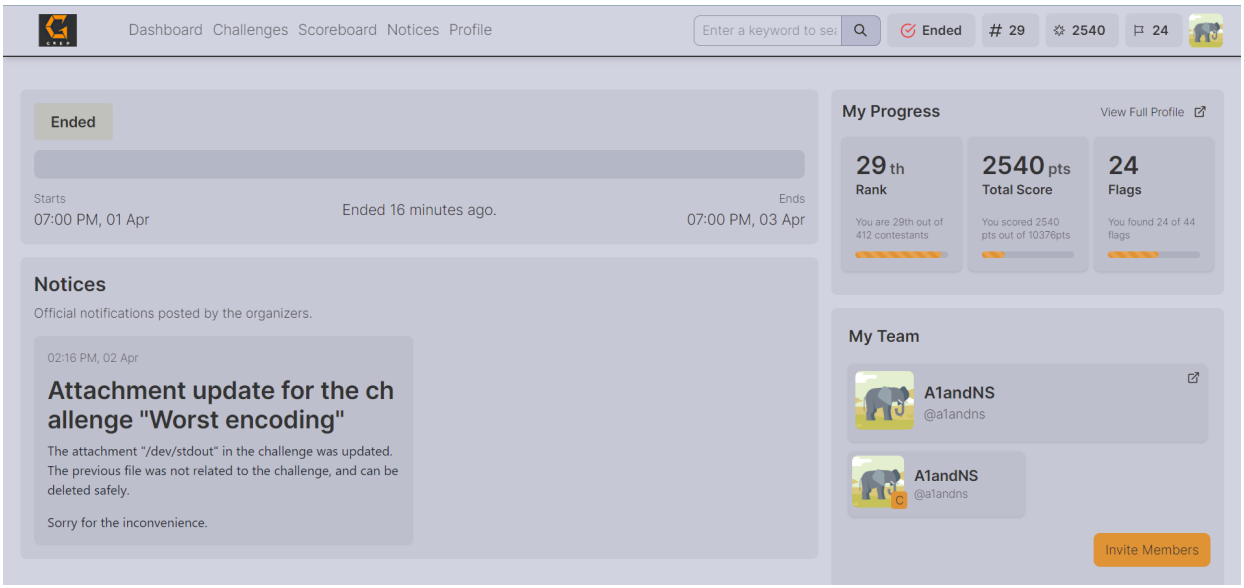
GREP CTF 2023 WP

> Author: A1andNS



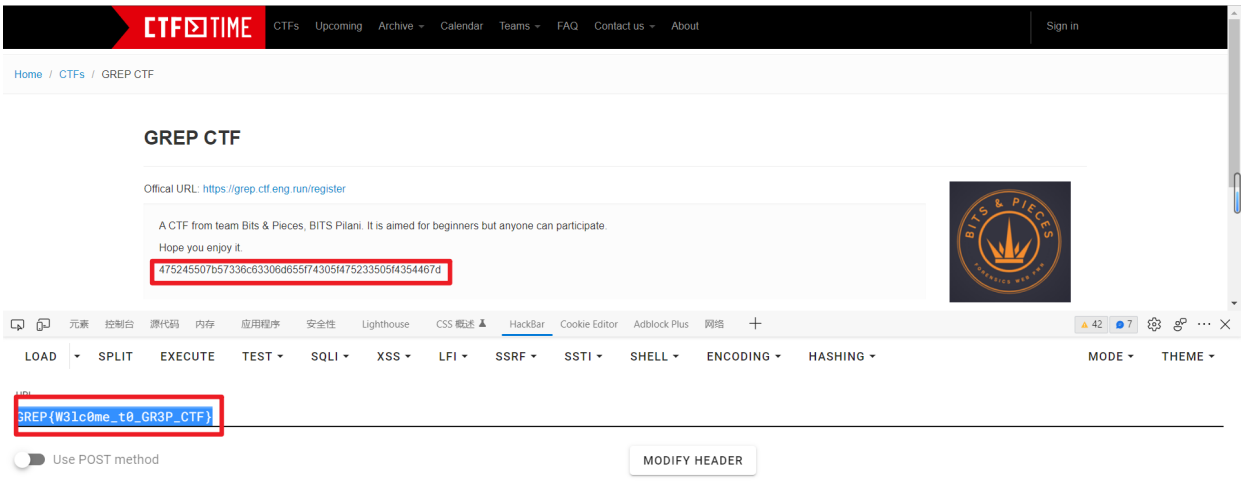
Misc

Approved!

Description: Finally, the clock ticks 3:48 am and I got approved. **TIME** flies when you are playing **CTF**. I welcome you all to GREP CTF, enjoy !! Oops, approved of what and where 😵, I leave that to you, gn....

According to the description of the challenge, We can know the flag is about GREP CTF in CTFTIME. So you can easily find out this CTF in CTFTIME in [CTFtime.org / GREP CTF]

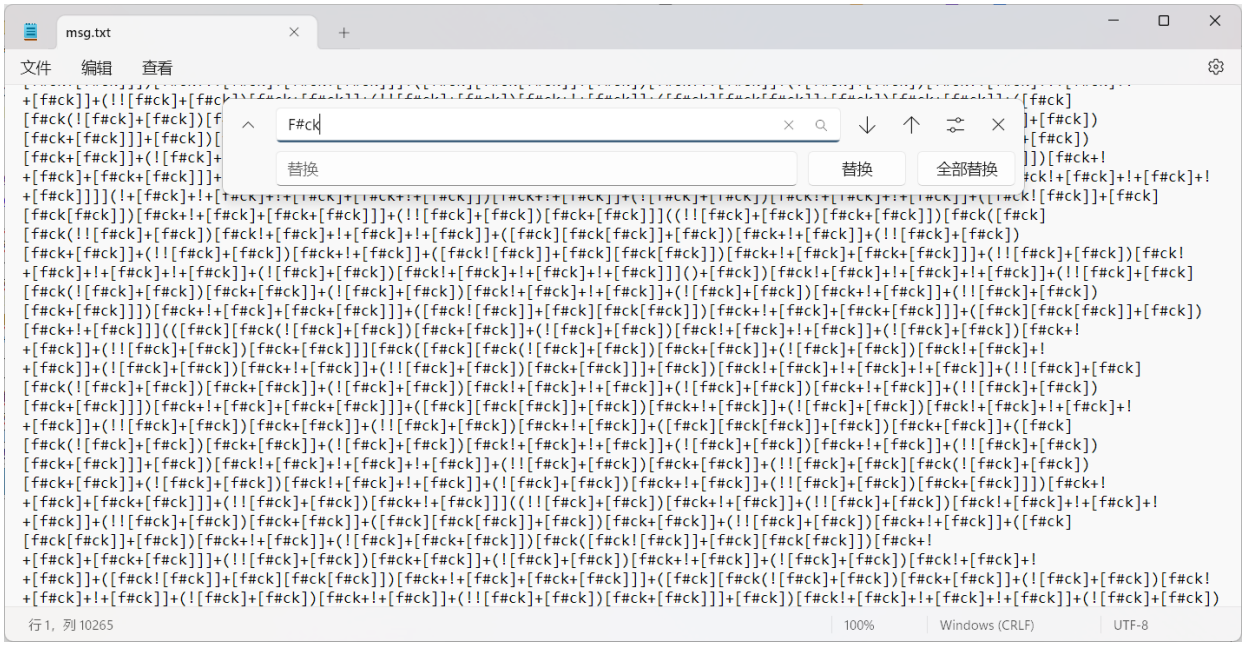
You can get a hexadecimal encode string, just decode it. You can get the flag.

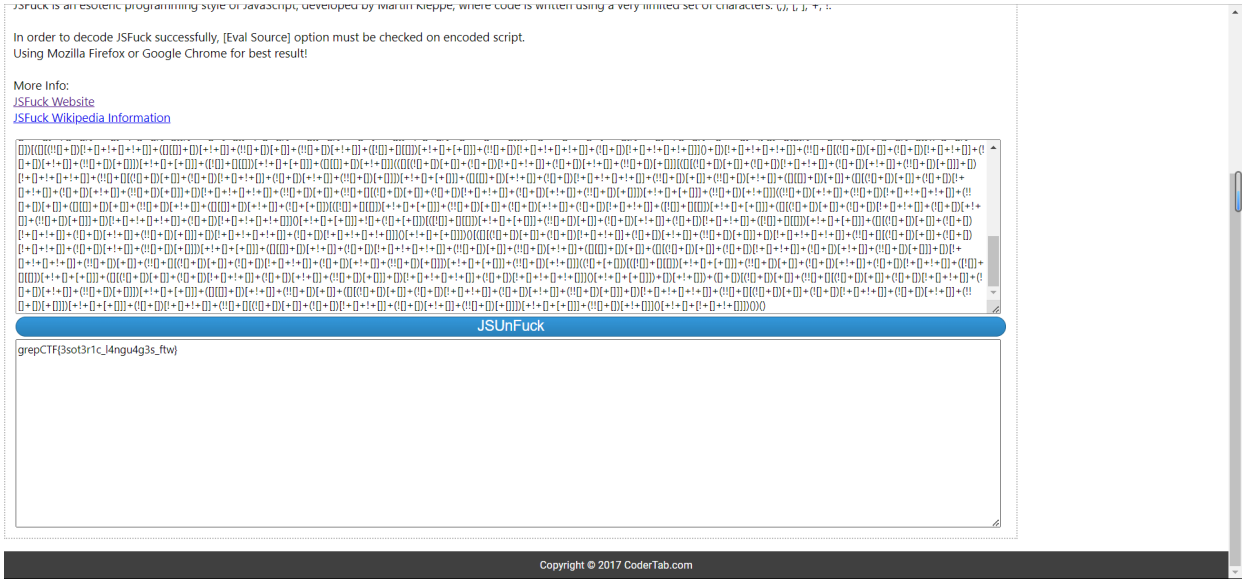


esoF*ck

DESCRIPTION: I've heard about brainf#ck but what the f#ck js this?

They are not a standard jsfuck encoded string, so change "F#ck" to null. Then jsunfuck the encoded string get a text via [CoderTab - JSUnFuck - Decode JSFuck Here]

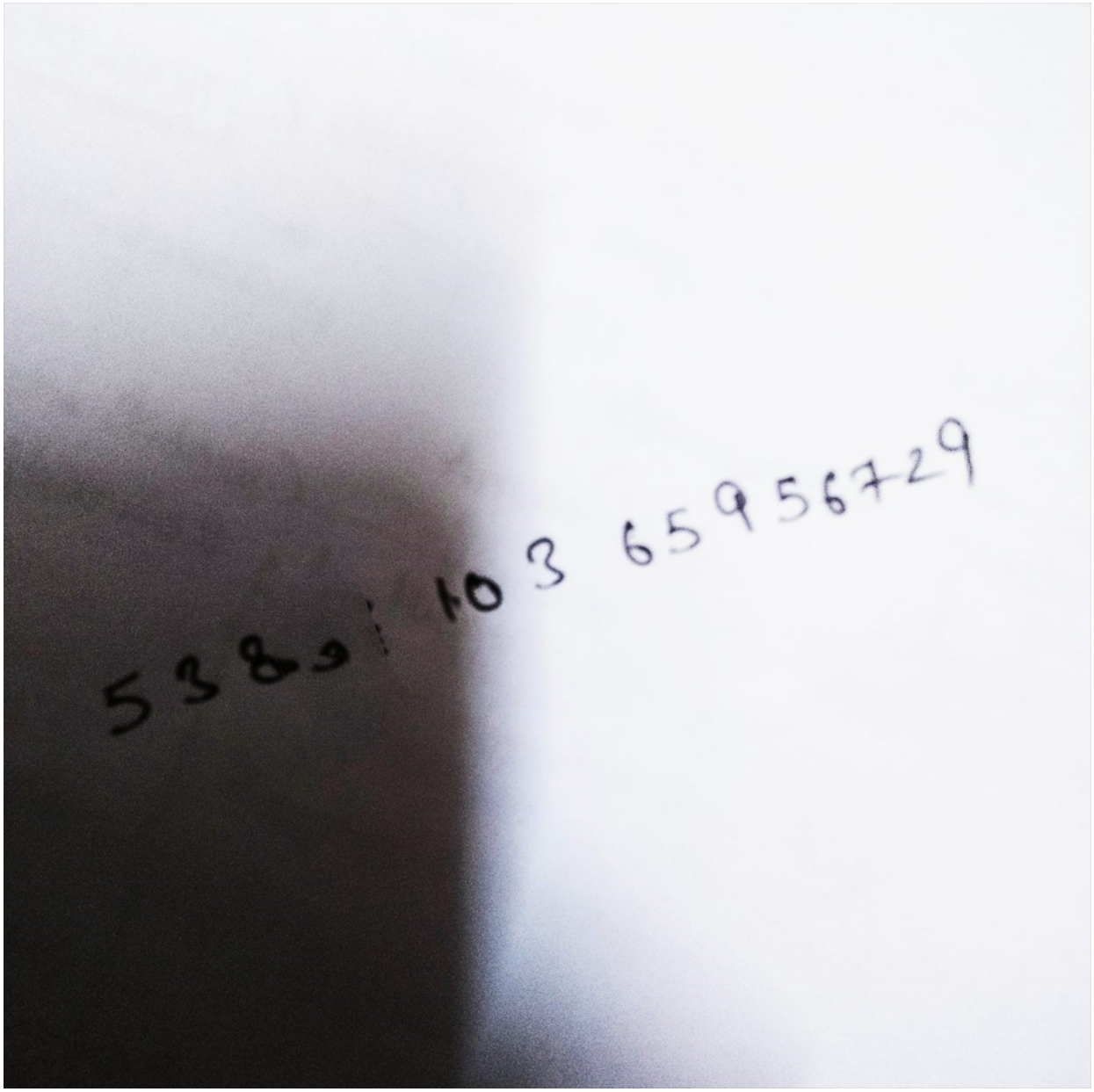




Lost Card

DESCRIPTION: Your card got lost while eating at the ANC. You have written it somewhere, but the two digits are unclear.

Firstly, make a mirror of the picture, we can obtain the result follow the picture. Then analyze the structure of the digits, you will find the No.4 digit is lacking the up part. Then, we will find number maybe is 8,3,6,5 by similar. we can easily find The No.5 digit is 1. So have a try, it's lucky that the number is 8, so we get the flag: GREP{5388110365956729}



Layout

DESCRIPTION: I've got this weird keyboard which has all the keys messed up. It's from someone called the workman.

use the automated cryptogram solver in [quipqiup - cryptoquip and cryptogram solver]

By observing the word "su;erman', I can easily know the ; is p

so flag is **grepCTF{r4pg0d_em1n3m_3256gd62}**

esoF*ck 2

DESCRIPTION: 2 levels of eso should make my message impossible to decipher.

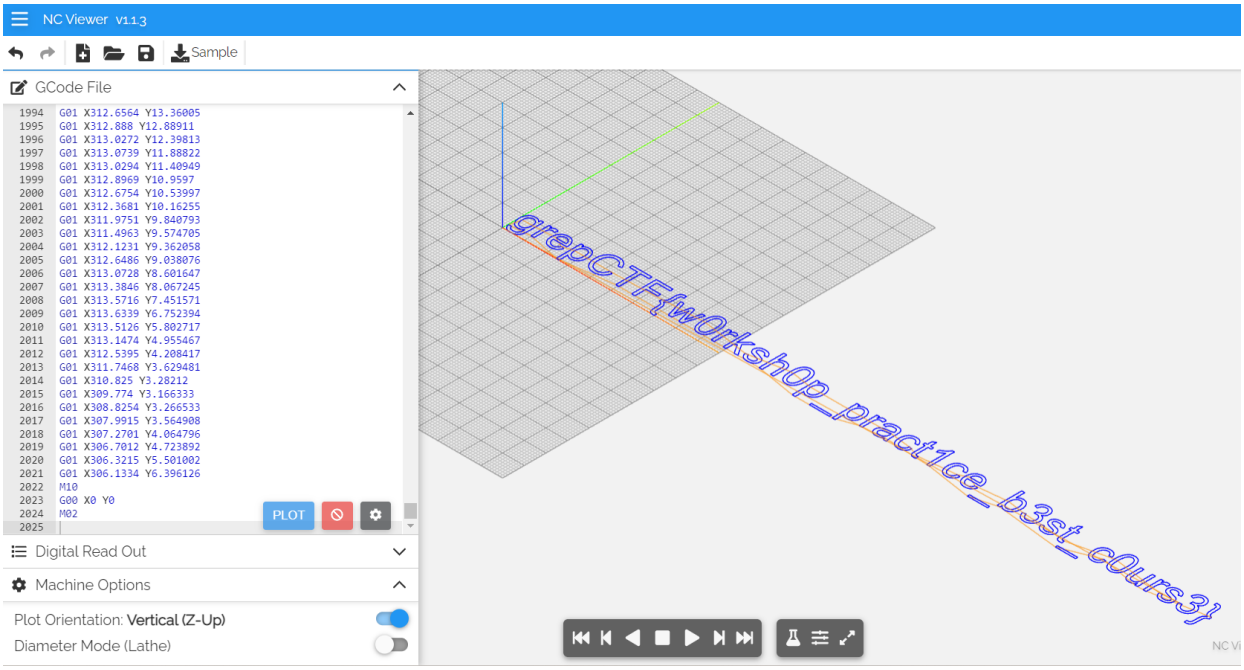
firstly, decode brainfuck then ork decode.https://tool.bugku.com/brainfuck/

grepCTF{3sot3r1c_l4ngu4g3s_4r3_0k!}

Consensual Non Consent

DESCRIPTION: Believe it or not, this is real coding.

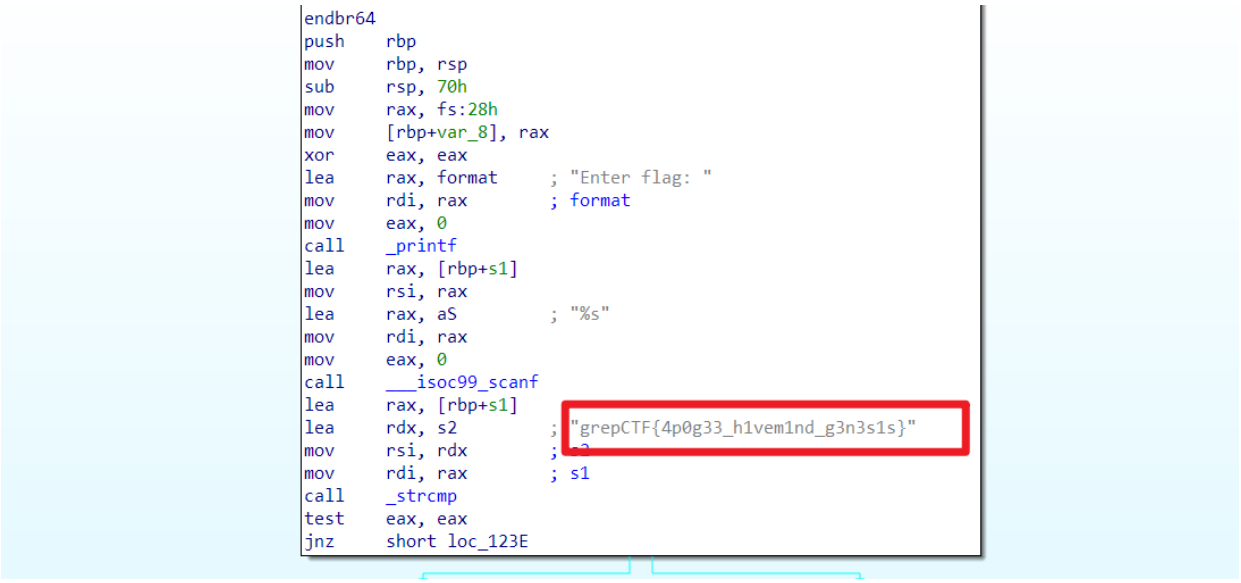
Get a file named wp.txt, We can find it's a G-Code, so run it by a simulator.



Reverse

Simple rev

Download the attachment and open it by IDA Pro. You will find the Flag string directly.

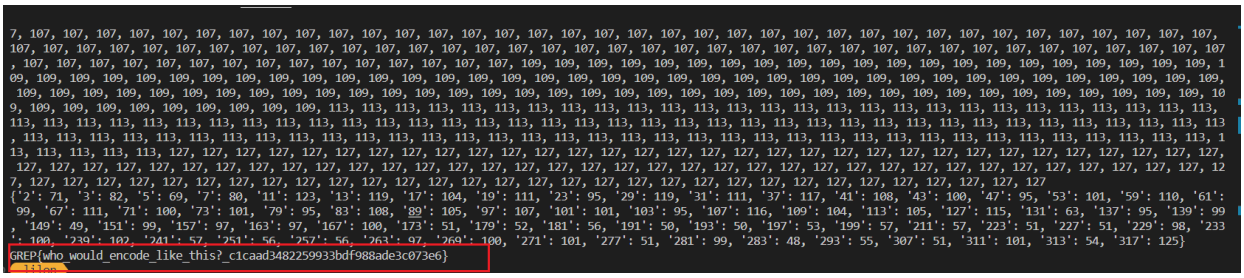


Worst encoding

jadx reverse the .jar file. And read the java code, find its algorithm. so we need to do a prime factoring.

```
try:
    n=int(input("please input integer: "))
    list1=[]
    if n !=1 and type(n)==int and n>1:

        while True:
            for i in range(2,n+1):
                b=n%i
                if b ==0:
                    list1.append(i)
                    n=n//i
                    break
            else:
                break


        print(list1)
        dic = {}
        for l in list1:
            try:
                dic[str(l)] += 1
            except KeyError:
                dic[str(l)] = 1
        print(dic)
        flag = ''
        for i in dic.keys():
```

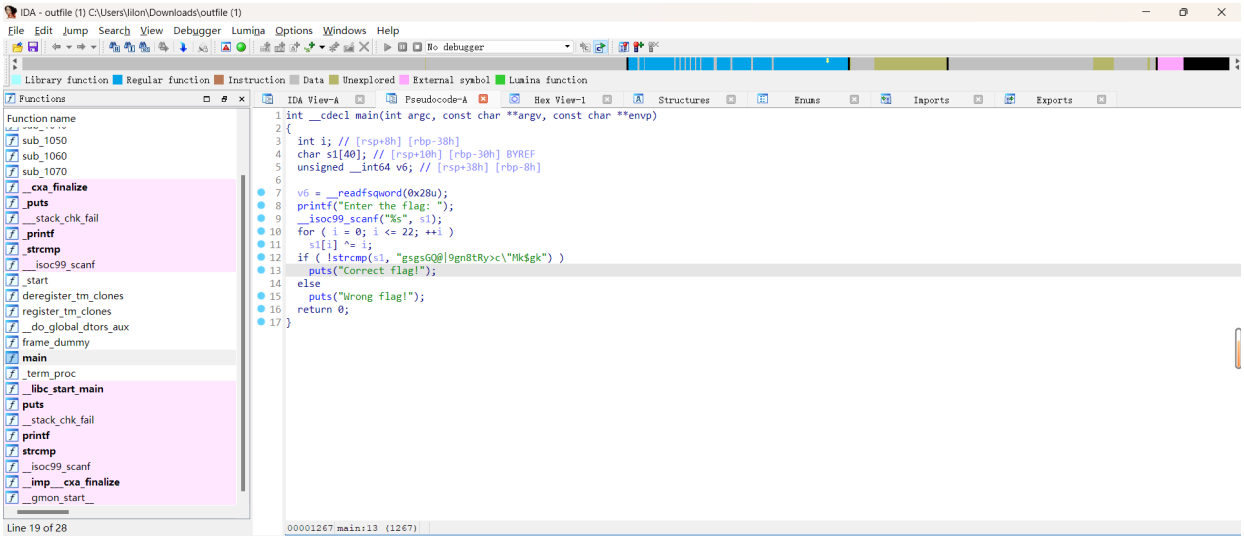
```
            flag += chr(dic[i])
        print(flag)
    elif n==1:
        print("1 can not be process")
    else: print("please input a right integer")

except ValueError:
    print("please input a right integer")
```



GREP{who_would_encode_like_this?_c1caad3482259933bdf988ade3c073e6}

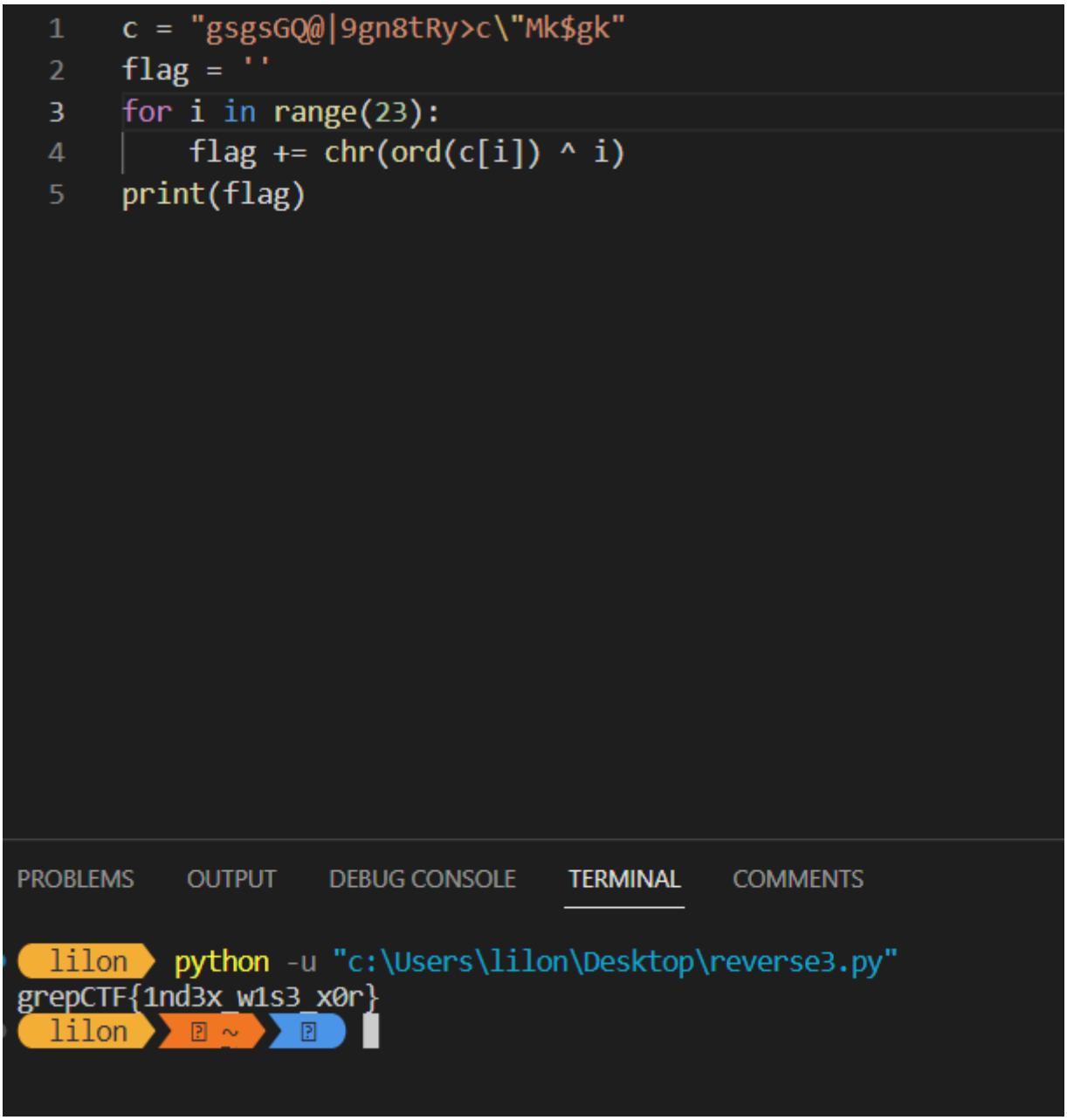
EXORcist

IDA analyze the outfile.



It's a easy XOR, so write a decrypt script.

```
c = "gsgsGQ@|9gn8tRy>c\"Mk$gk"
flag = ''
for i in range(23):
    flag += chr(ord(c[i]) ^ i)
print(flag)
```

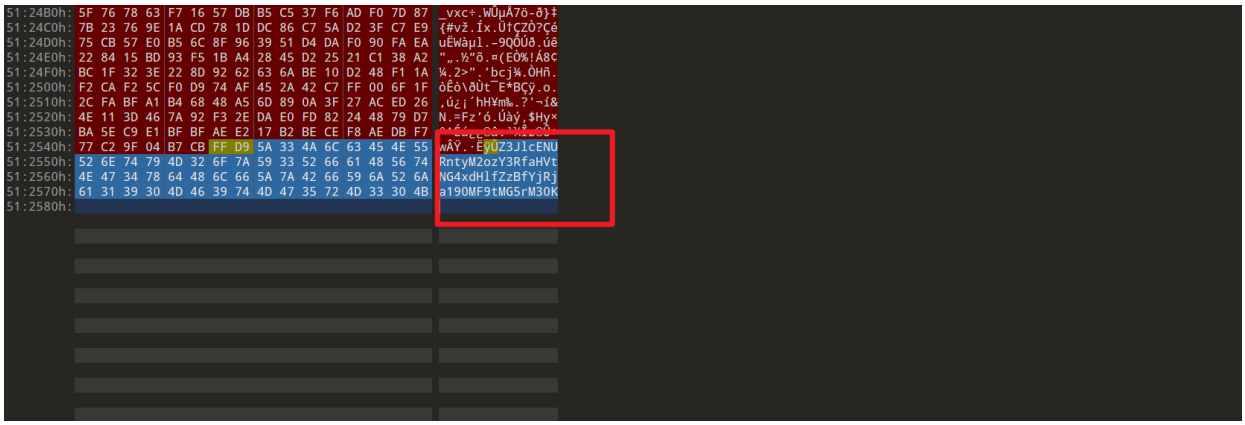


Forensics

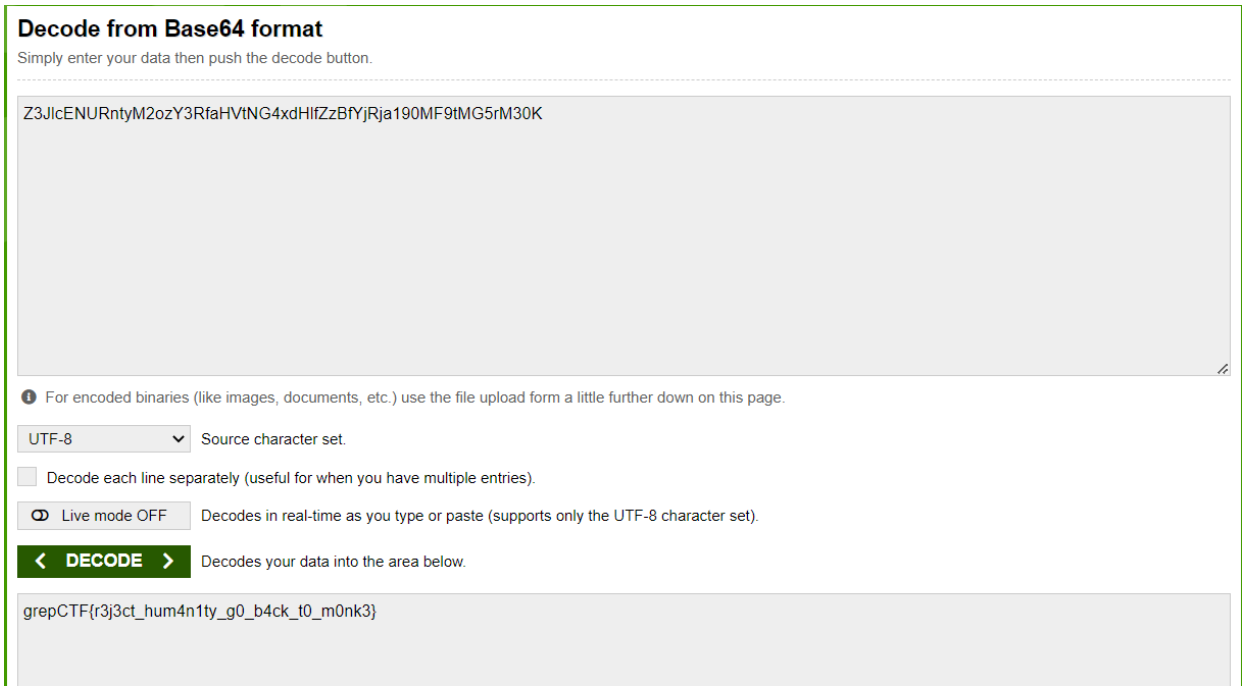
Monke

DESCRIPTION: I was playing guitar and then this monkey came and broke all my strings 😢

You can use 010 Editor to analyze the picture and find the interesting string in the end of the hexadecimal data.



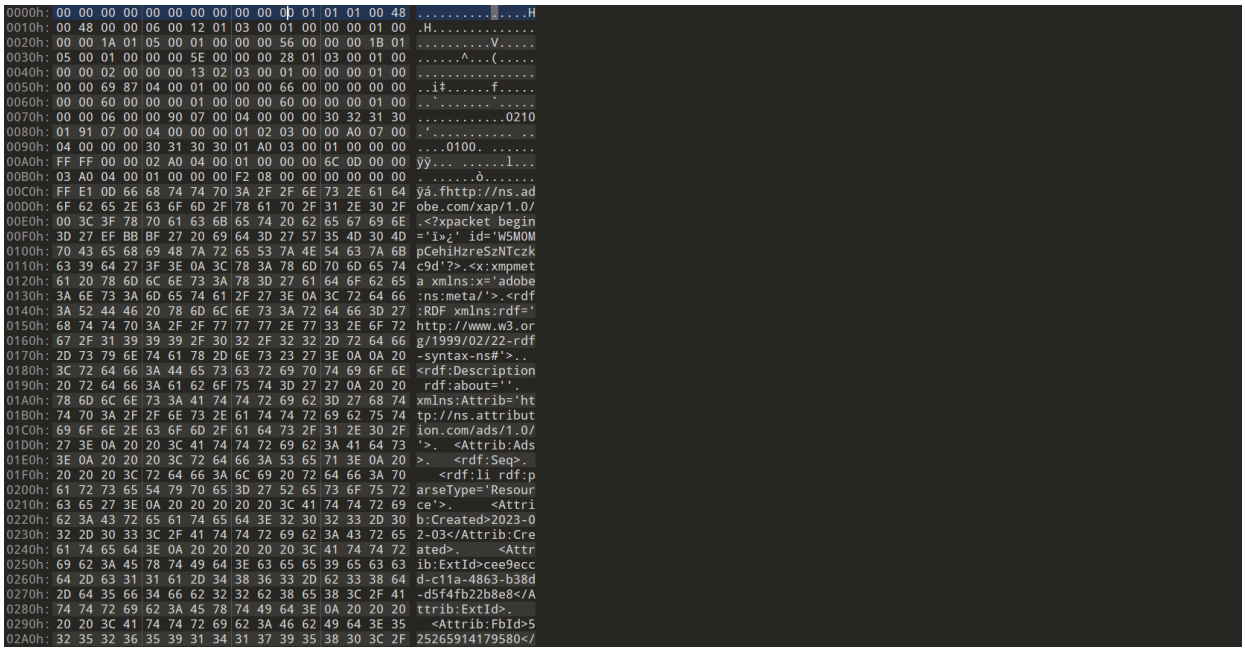
You can easily decode it by base64 format in [Base64 Decode and Encode - Online]

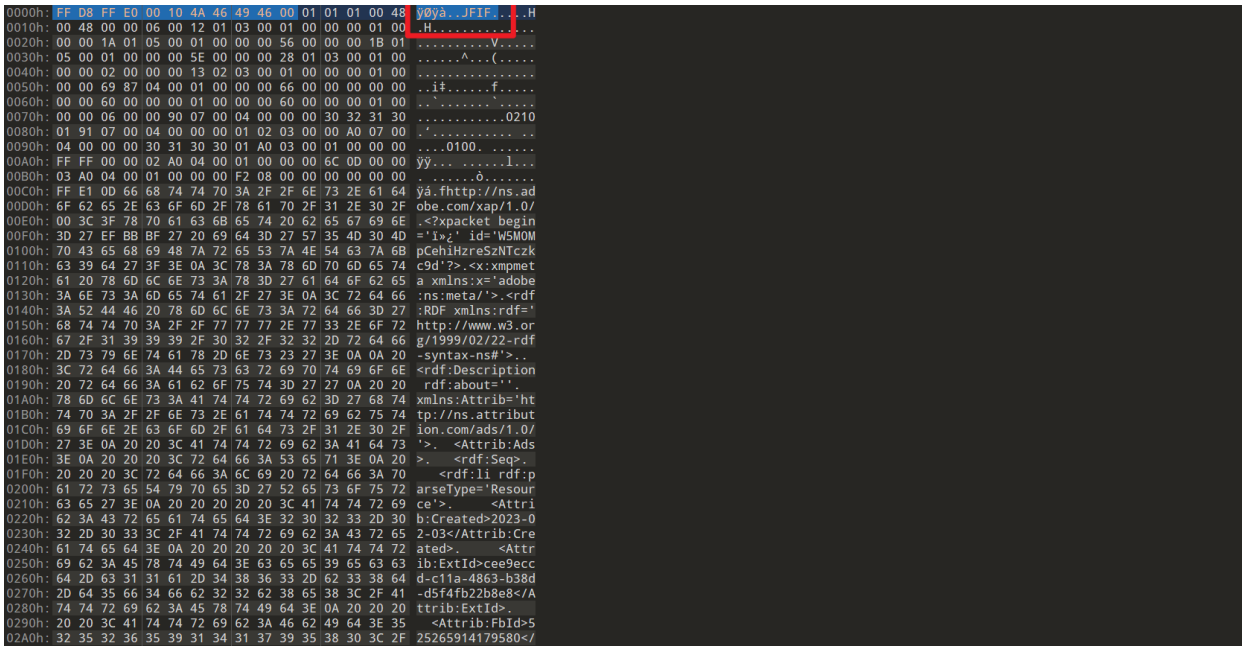


Doctored image

DESCRIPTION: Help, my images are all corrupted!!

You can find this picture lacking a file head of jps in 010 editor, so fixed it.

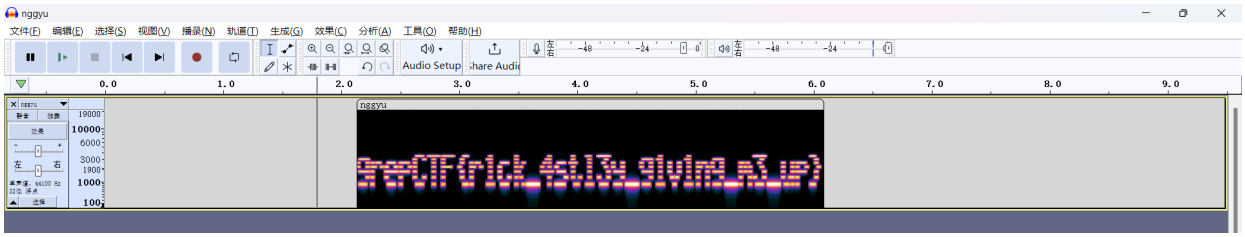

000Oh: 00 00 00 00 00 00 00 00 00 00 0D 01 01 01 00 48 .....H
0010h: 00 48 00 00 06 00 12 01 03 00 01 00 00 00 01 00 .H..............
0020h: 00 00 1A 01 05 00 01 00 00 00 56 00 00 00 1B 01 ....V.....
0030h: 05 00 01 00 00 00 5E 00 00 00 28 01 03 00 01 00 ......^...(.....
0040h: 00 00 02 00 00 00 13 02 03 00 01 00 00 00 01 00 ....
0050h: 00 00 69 87 04 00 01 00 00 00 66 00 00 00 00 00 ..i‡......f.....
0060h: 00 00 60 00 00 00 01 00 00 00 60 00 00 00 01 00 ..`.......`.....
0070h: 00 00 06 00 00 90 07 00 04 00 00 00 30 32 31 30 0210
0080h: 01 91 07 00 04 00 00 00 01 02 03 00 00 A0 07 00 .'.......... ..
0090h: 04 00 00 00 30 31 30 30 01 A0 03 00 01 00 00 00 ....0100. ......
00A0h: FF FF 00 00 02 A0 04 00 01 00 00 00 6C 0D 00 00 ÿÿ... ......l...
00B0h: 03 A0 04 00 01 00 00 00 F2 08 00 00 00 00 00 00 . ...ò.......
00C0h: FF E1 0D 66 68 74 74 70 3A 2F 2F 6E 73 2E 61 64 ÿá.fhttp://ns.ad
00D0h: 6F 62 65 2E 63 6F 6D 2F 78 61 70 2F 31 2E 30 2F obe.com/xap/1.0/
00E0h: 00 3C 3F 78 70 61 63 6B 65 74 20 62 65 67 69 6E .<?xpacket begin
00F0h: 3D 27 EF BB BF 27 20 69 64 3D 27 57 35 4D 30 4D ='ï»¿' id='W5MOM
0100h: 70 43 65 68 69 48 7A 72 65 53 7A 4E 54 63 7A 6B pCehiHzreSzNTczk
0110h: 63 39 64 27 3F 3E 0A 3C 78 3A 78 6D 70 6D 65 74 c9d'?>.<x:xmpmet
0120h: 61 20 78 6D 6C 6E 73 3A 78 3D 27 61 64 6F 62 65 a xmlns:x='adobe
0130h: 3A 6E 73 3A 6D 65 74 61 2F 27 3E 0A 3C 72 64 66 :ns:meta/'>.<rdf
0140h: 3A 52 44 46 20 78 6D 6C 6E 73 3A 72 64 66 3D 27 :RDF xmlns:rdf='
0150h: 68 74 74 70 3A 2F 2F 77 77 77 2E 77 33 2E 6F 72 http://www.w3.or
0160h: 67 2F 31 39 39 39 2F 30 32 2F 32 32 2D 72 64 66 g/1999/02/22-rdf
0170h: 2D 73 79 6E 74 61 78 2D 6E 73 23 27 3E 0A 0A 20 -syntax-ns#'>..
0180h: 3C 72 64 66 3A 44 65 73 63 72 69 70 74 69 6F 6E <rdf:Description
0190h: 20 72 64 66 3A 61 62 6F 75 74 3D 27 27 0A 20 20 rdf:about=''.
01A0h: 78 6D 6C 6E 73 3A 41 74 74 72 69 62 3D 27 68 74 xmlns:Attrib='ht
01B0h: 74 70 3A 2F 2F 6E 73 2E 61 74 74 72 69 62 75 74 tp://ns.attribut
01C0h: 69 6F 6E 2E 63 6F 6D 2F 61 64 73 2F 31 2E 30 2F ion.com/ads/1.0/
01D0h: 27 3E 0A 20 20 3C 41 74 74 72 69 62 3A 41 64 73 '>. <Attrib:Ads
01E0h: 3E 0A 20 20 20 3C 72 64 66 3A 53 65 71 3E 0A 20 >.   <rdf:Seq>.
01F0h: 20 20 20 3C 72 64 66 3A 6C 69 20 72 64 66 3A 70   <rdf:li rdf:p
0200h: 61 72 73 65 54 79 70 65 3D 27 52 65 73 6F 75 72 arseType='Resour
0210h: 63 65 27 3E 0A 20 20 20 20 20 3C 41 74 74 72 69 ce'>.   <Attri
0220h: 62 3A 43 72 65 61 74 65 64 3E 32 30 32 33 2D 30 b:Created>2023-0
0230h: 32 2D 30 33 3C 2F 41 74 74 72 69 62 3A 43 72 65 2-03</Attrib:Cre
0240h: 61 74 65 64 3E 0A 20 20 20 20 20 3C 41 74 74 72 ated>.   <Attr
0250h: 69 62 3A 45 78 74 49 64 3E 63 65 65 39 65 63 63 ib:ExtId>cee9ecc
0260h: 64 2D 63 31 31 61 2D 34 38 36 33 2D 62 33 38 64 d-c11a-4863-b38d
0270h: 2D 64 35 66 34 66 62 32 32 62 38 65 38 3C 2F 41 -d5f4fb22b8e8</A
0280h: 74 74 72 69 62 3A 45 78 74 49 64 3E 0A 20 20 20 ttrib:ExtId>.
0290h: 20 20 3C 41 74 74 72 69 62 3A 46 62 49 64 3E 35   <Attrib:FbId>5
02A0h: 32 35 32 36 35 39 31 34 31 37 39 35 38 30 3C 2F 252659141795 80</


000Oh: FF D8 FF E0 00 10 4A 46 49 46 00 01 01 01 00 48 ÿØÿà..JFIF.....H
0010h: 00 48 00 00 06 00 12 01 03 00 01 00 00 00 01 00 .H..............
0020h: 00 00 1A 01 05 00 01 00 00 00 56 00 00 00 1B 01 ....V.....
0030h: 05 00 01 00 00 00 5E 00 00 00 28 01 03 00 01 00 ......^...(.....
0040h: 00 00 02 00 00 00 13 02 03 00 01 00 00 00 01 00 ....
0050h: 00 00 69 87 04 00 01 00 00 00 66 00 00 00 00 00 ..i‡......f.....
0060h: 00 00 60 00 00 00 01 00 00 00 60 00 00 00 01 00 ..`.......`.....
0070h: 00 00 06 00 00 90 07 00 04 00 00 00 30 32 31 30 0210
0080h: 01 91 07 00 04 00 00 00 01 02 03 00 00 A0 07 00 .'.......... ..
0090h: 04 00 00 00 30 31 30 30 01 A0 03 00 01 00 00 00 ....0100. ......
00A0h: FF FF 00 00 02 A0 04 00 01 00 00 00 6C 0D 00 00 ÿÿ... ......l...
00B0h: 03 A0 04 00 01 00 00 00 F2 08 00 00 00 00 00 00 . ...ò.......
00C0h: FF E1 0D 66 68 74 74 70 3A 2F 2F 6E 73 2E 61 64 ÿá.fhttp://ns.ad
00D0h: 6F 62 65 2E 63 6F 6D 2F 78 61 70 2F 31 2E 30 2F obe.com/xap/1.0/
00E0h: 00 3C 3F 78 70 61 63 6B 65 74 20 62 65 67 69 6E .<?xpacket begin
00F0h: 3D 27 EF BB BF 27 20 69 64 3D 27 57 35 4D 30 4D ='ï»¿' id='W5MOM
0100h: 70 43 65 68 69 48 7A 72 65 53 7A 4E 54 63 7A 6B pCehiHzreSzNTczk
0110h: 63 39 64 27 3F 3E 0A 3C 78 3A 78 6D 70 6D 65 74 c9d'?>.<x:xmpmet
0120h: 61 20 78 6D 6C 6E 73 3A 78 3D 27 61 64 6F 62 65 a xmlns:x='adobe
0130h: 3A 6E 73 3A 6D 65 74 61 2F 27 3E 0A 3C 72 64 66 :ns:meta/'>.<rdf
0140h: 3A 52 44 46 20 78 6D 6C 6E 73 3A 72 64 66 3D 27 :RDF xmlns:rdf='
0150h: 68 74 74 70 3A 2F 2F 77 77 77 2E 77 33 2E 6F 72 http://www.w3.or
0160h: 67 2F 31 39 39 39 2F 30 32 2F 32 32 2D 72 64 66 g/1999/02/22-rdf
0170h: 2D 73 79 6E 74 61 78 2D 6E 73 23 27 3E 0A 0A 20 -syntax-ns#'>..
0180h: 3C 72 64 66 3A 44 65 73 63 72 69 70 74 69 6F 6E <rdf:Description
0190h: 20 72 64 66 3A 61 62 6F 75 74 3D 27 27 0A 20 20 rdf:about=''.
01A0h: 78 6D 6C 6E 73 3A 41 74 74 72 69 62 3D 27 68 74 xmlns:Attrib='ht
01B0h: 74 70 3A 2F 2F 6E 73 2E 61 74 74 72 69 62 75 74 tp://ns.attribut
01C0h: 69 6F 6E 2E 63 6F 6D 2F 61 64 73 2F 31 2E 30 2F ion.com/ads/1.0/
01D0h: 27 3E 0A 20 20 3C 41 74 74 72 69 62 3A 41 64 73 '>. <Attrib:Ads
01E0h: 3E 0A 20 20 20 3C 72 64 66 3A 53 65 71 3E 0A 20 >.   <rdf:Seq>.
01F0h: 20 20 20 3C 72 64 66 3A 6C 69 20 72 64 66 3A 70   <rdf:li rdf:p
0200h: 61 72 73 65 54 79 70 65 3D 27 52 65 73 6F 75 72 arseType='Resour
0210h: 63 65 27 3E 0A 20 20 20 20 20 3C 41 74 74 72 69 ce'>.   <Attri
0220h: 62 3A 43 72 65 61 74 65 64 3E 32 30 32 33 2D 30 b:Created>2023-0
0230h: 32 2D 30 33 3C 2F 41 74 74 72 69 62 3A 43 72 65 2-03</Attrib:Cre
0240h: 61 74 65 64 3E 0A 20 20 20 20 20 3C 41 74 74 72 ated>.   <Attr
0250h: 69 62 3A 45 78 74 49 64 3E 63 65 65 39 65 63 63 ib:ExtId>cee9ecc
0260h: 64 2D 63 31 31 61 2D 34 38 36 33 2D 62 33 38 64 d-c11a-4863-b38d
0270h: 2D 64 35 66 34 66 62 32 32 62 38 65 38 3C 2F 41 -d5f4fb22b8e8</A
0280h: 74 74 72 69 62 3A 45 78 74 49 64 3E 0A 20 20 20 ttrib:ExtId>.
0290h: 20 20 3C 41 74 74 72 69 62 3A 46 62 49 64 3E 35   <Attrib:FbId>5
02A0h: 32 35 32 36 35 39 31 34 31 37 39 35 38 30 3C 2F 252659141795 80</

NGGYU

DESCRIPTION: .

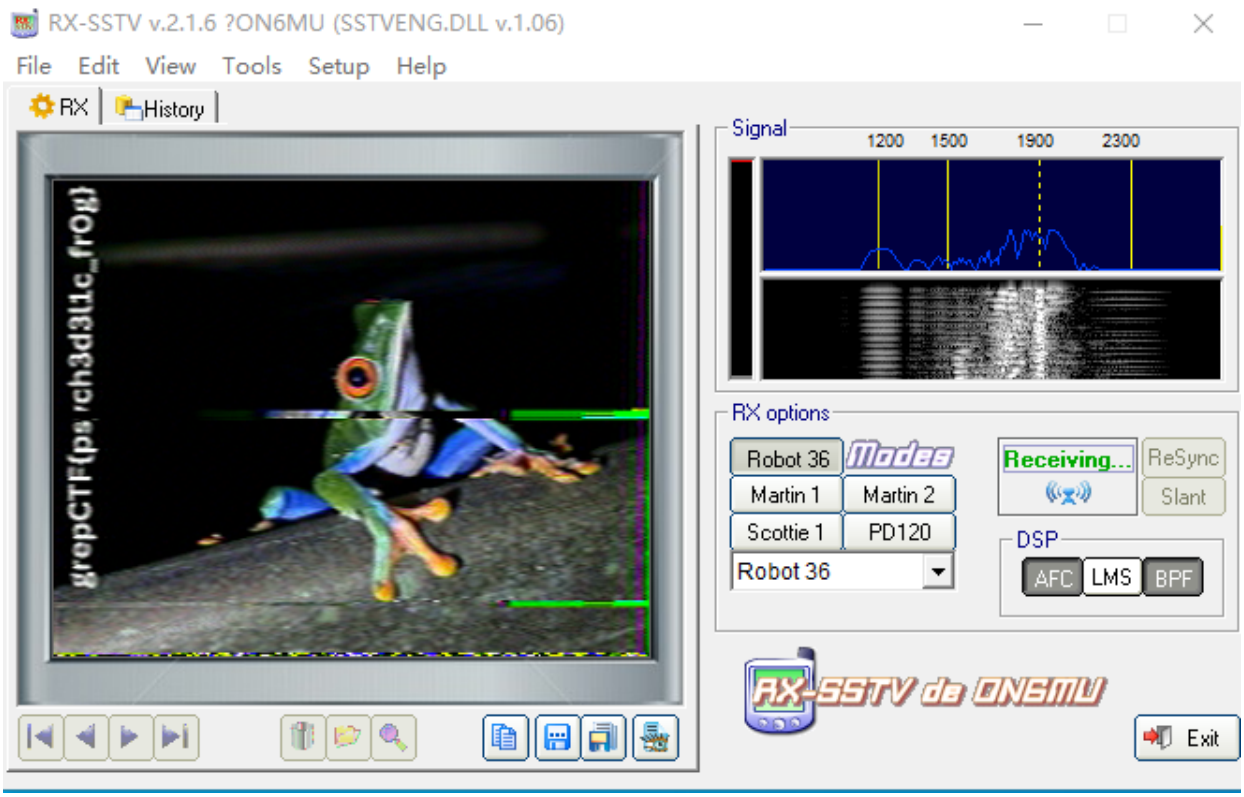
Open the audio file by audacity and change to see the spectrogram.



grepCTF{r1ck_4stl3y_g1v1ng_m3_up}

R36

use RX-SSTV to process the .wav file in Mode robot36.



grepCTF{psych3d3l1c_fr0g}

IronMan

The attachment is a picture of IronMan. Nice! Analyze it by zsteg.

```
zsteg ironman.png -a | grep CTF
```

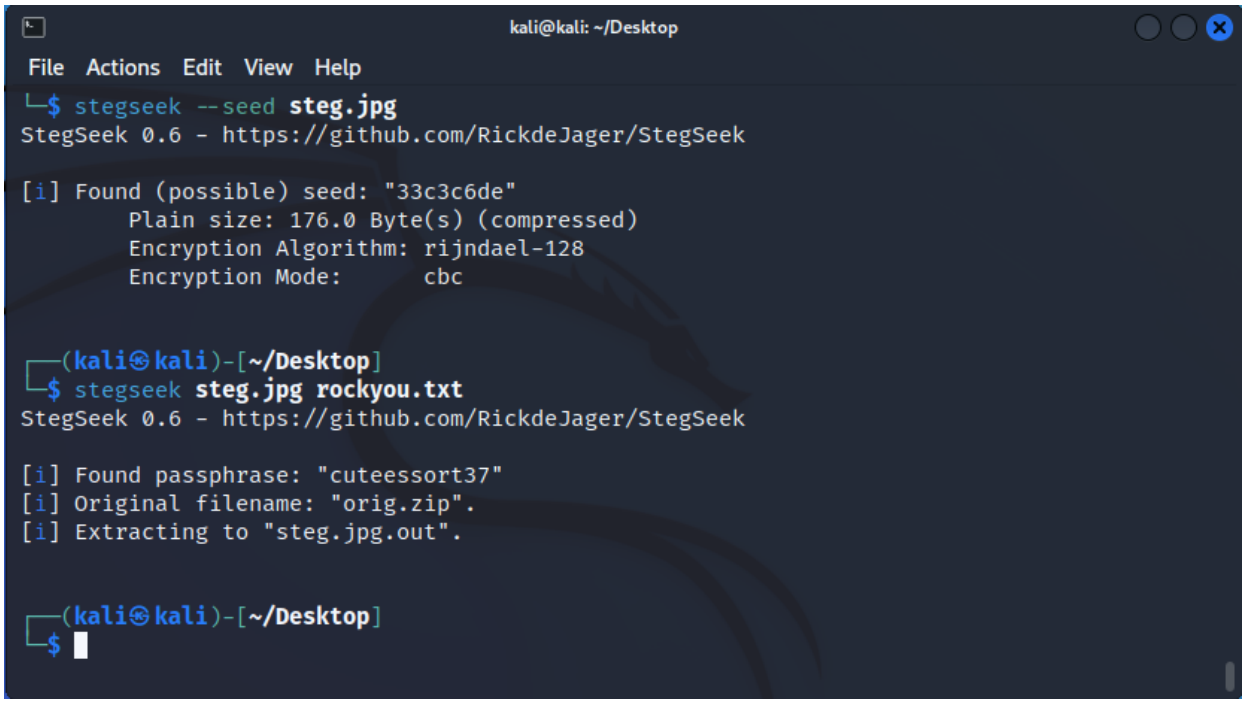
```
┌──(kali㉿kali)-[~/Desktop]
└─$ zsteg ironman.png -a | grep CTF
b1,rgb,lsb,yx       .. text: "grepCTF{i_d0n't_f3el_s0_g00d}\n"
```

Royal Steg

DESCRIPTION: Then Jesus turned, and seeing them following, said to them, 'what do you SEEK?

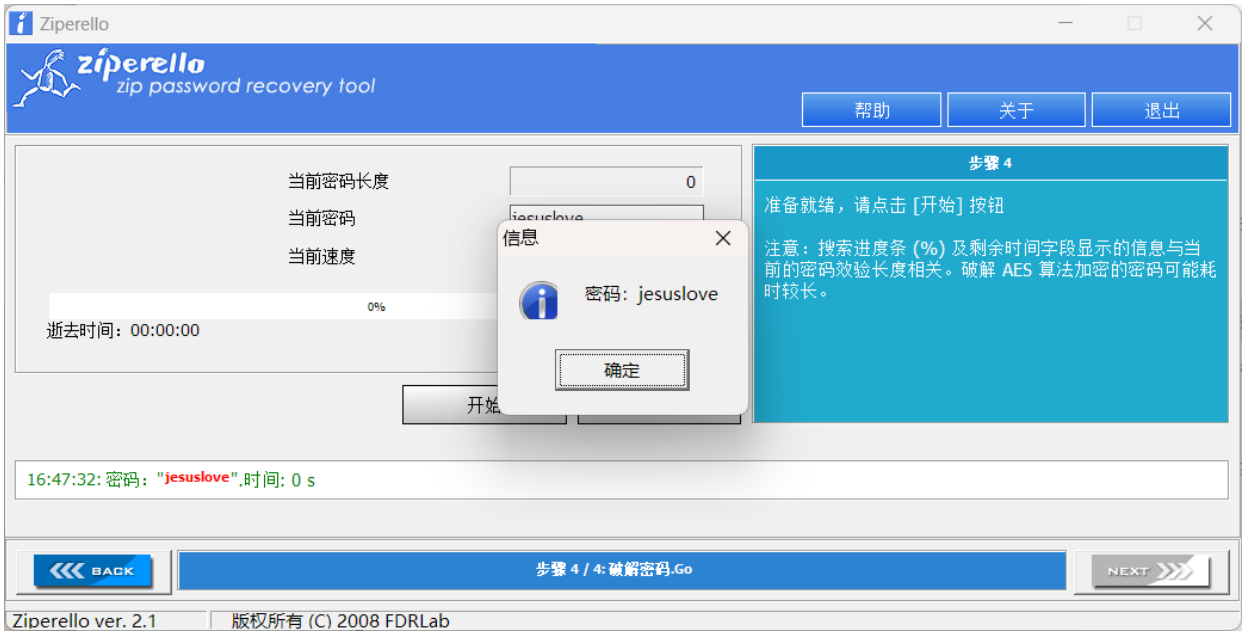
- JOHN 1:38

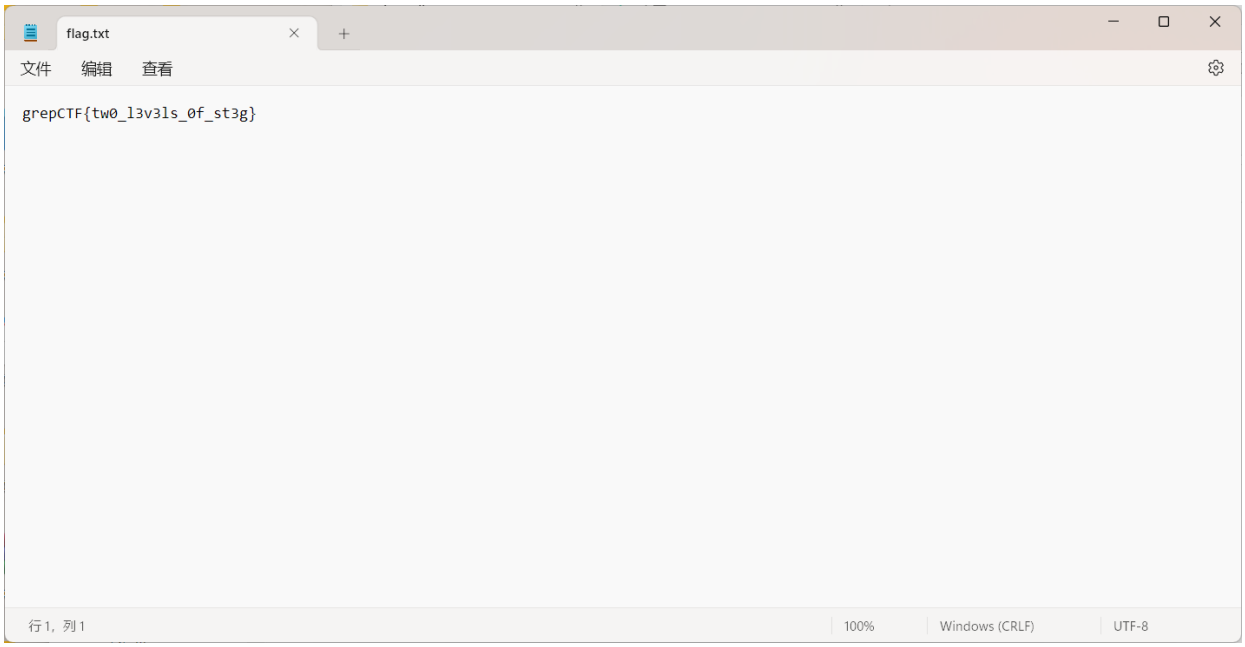
do you see the SEEK, so try stegseek.



get a orig.zip and the flag.txt is in it. However, the zip looks encrypted.

use a zip burster to burst the password of the zip file with the rockyou.txt. the password is jesuslove.





get flag : grepCTF{tw0_l3v3ls_0f_st3g}

Cryptography

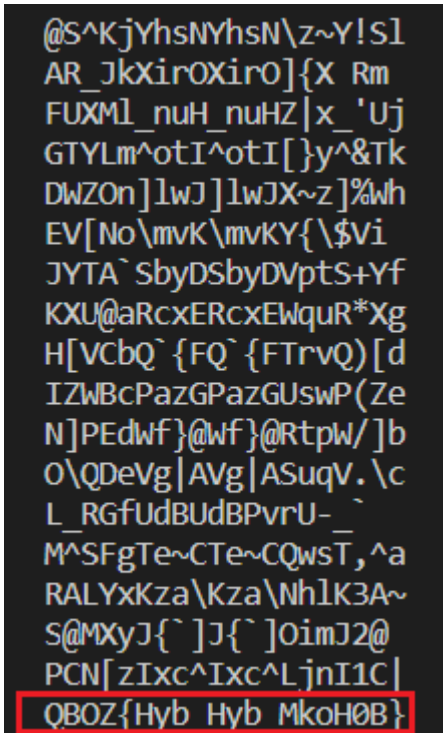
CaeXOR

DESCRIPTION：I pressed shift key 10 times and lost the flag. Can you find my flag.

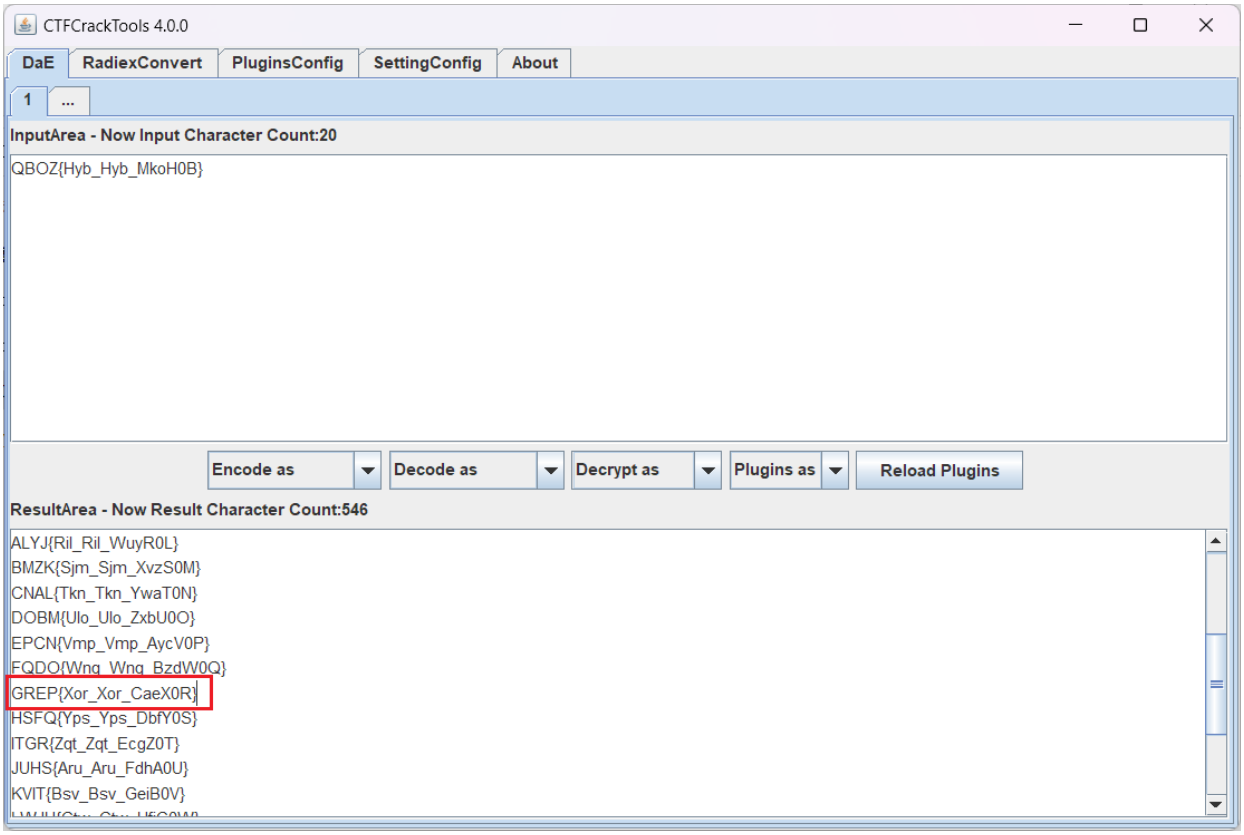
```
#enc.py
from random import *
flag="REDACTED"
a=randint(1,1000)
c=[]
for f in flag:
   c.append(str(ord(f)^a))
print(c)
print(a)

#c=['162', '177', '188', '169', '136', '187', '138', '145', '172', '187', '138',
'145', '172', '190', '152', '156', '187', '195', '177', '142']
#a=REDACTED
```

solution script:

```
f = open('test.txt','w',encoding='utf-8')
c=['162', '177', '188', '169', '136', '187', '138', '145', '172', '187', '138',
'145', '172', '190', '152', '156', '187', '195', '177', '142']
for i in range(1,1001):
   flag = ''
   for j in c:
      flag += chr(int(j)^i)
   print(flag)
   f.write(flag+"\n")
f.close()
```



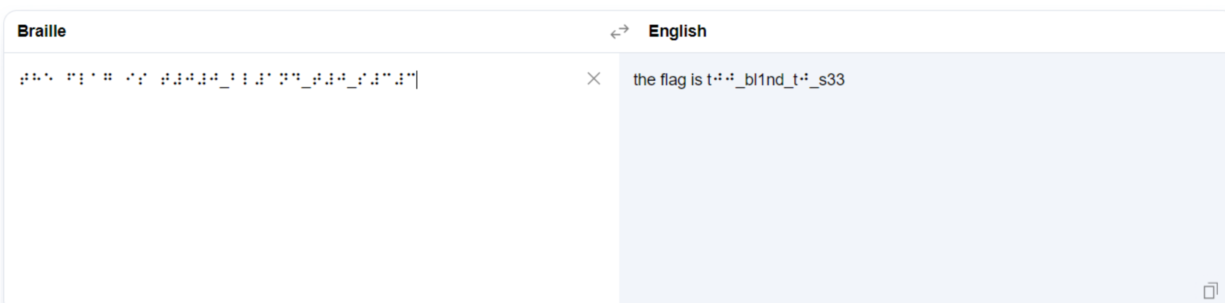
I find a string result "QBOZ{Hyb_Hyb_MkoH0B}" is like a flag format, but it is not GREP, so do a caesarcode decryption. Get the flag.



Blind

⠞⠓⠑ ⠋⠇⠁⠛ ⠊⠎ ⠞⠼⠚⠼⠚⠃⠇⠼⠁⠝⠙⠞⠼⠚_⠎⠼⠉⠼⠉

exchange it to English in [盲文翻译器：转换器和解码器 - SYMBL (◕‿◕)]

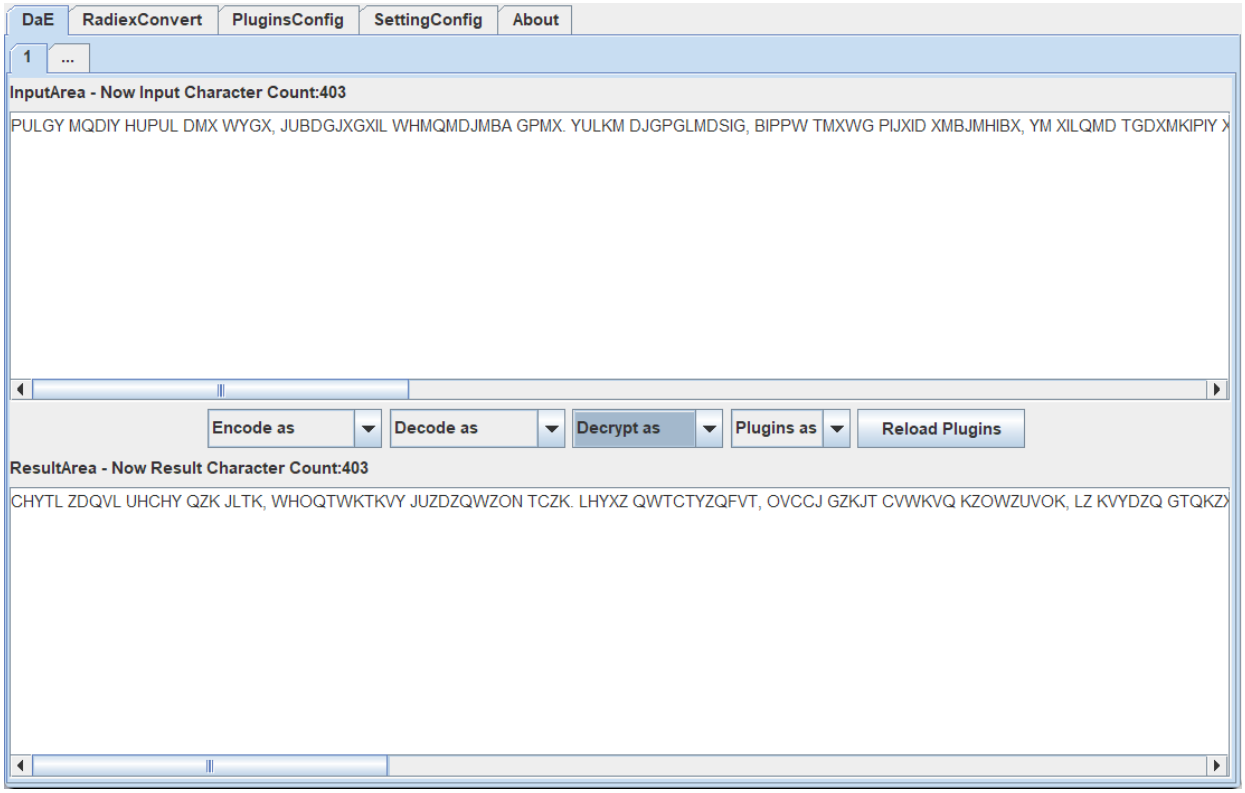


This website can't translate ⠼⠚

⠼⠚ is 0, so flag is **grepCTF{t00_bl1nd_t0_s33}**

NOT 13

rot13 decrypted the string.



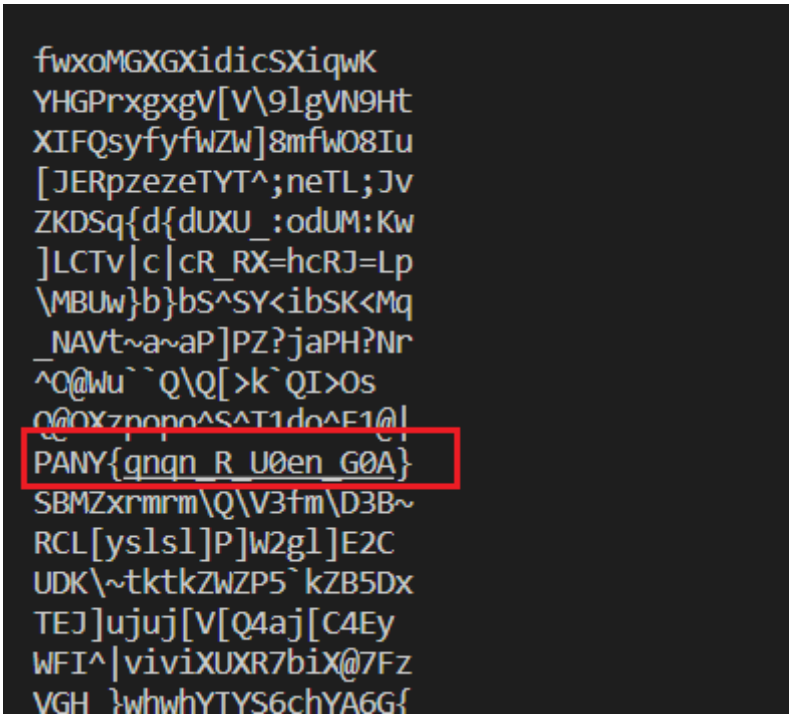
then, analyze it in quipqiup.

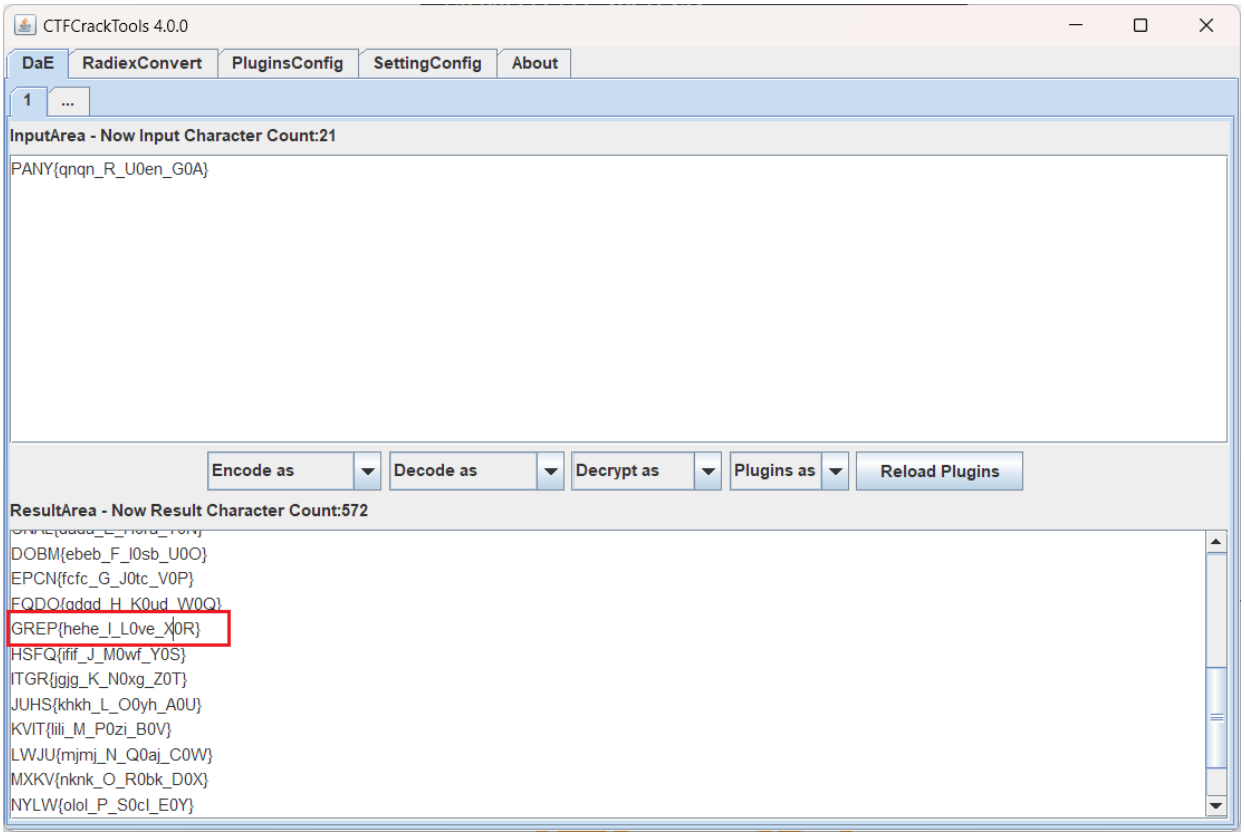
CaeX0R 2

DESCRIPTION: Ooops, i forgot the shift this time. Can you still figure out my flag.

```python
#enc.py
from random import *
flag="REDACTED"
a=randint(1,1000)
c=[]
for f in flag:
   c.append(str(ord(f)^a))
print(c)
print(a)

#c=['313', '296', '295', '304', '274', '280', '263', '280', '263', '310', '315',
'310', '316', '345', '268', '263', '310', '302', '345', '296', '276']
#a=REDACTED
```

```python
f = open('test.txt','w',encoding='utf-8')
c=['313', '296', '295', '304', '274', '280', '263', '280', '263', '310', '315',
'310', '316', '345', '268', '263', '310', '302', '345', '296', '276']
for i in range(1,1001):
   flag = ''
   for j in c:
      flag += chr(int(j)^i)
   print(flag)
   f.write(flag+"\n")
f.close()
```





like CaeXOR, just decrypt by a script and caesar decrypt.

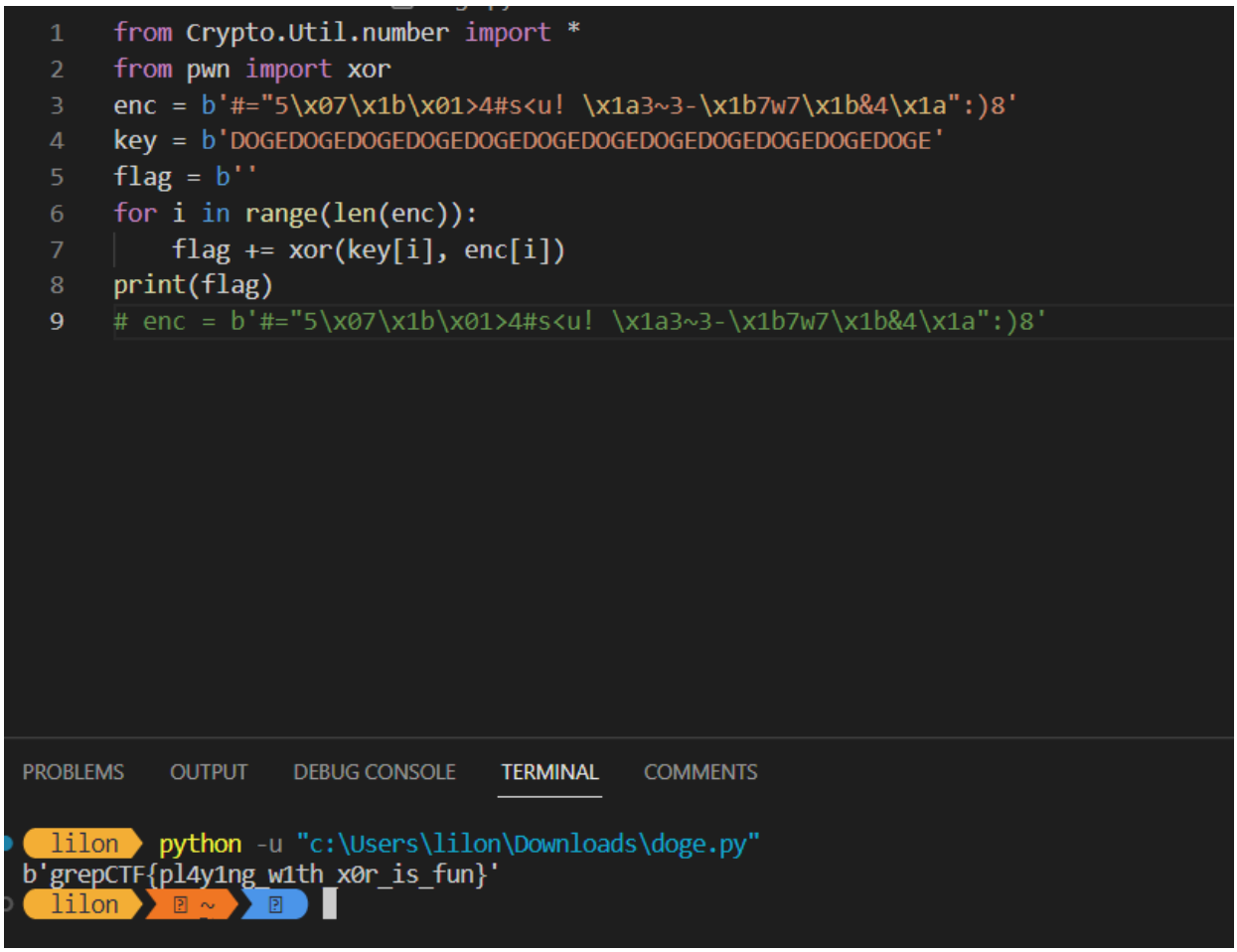
DOGE DOGE DOGE

challenge's code:

```
from Crypto.Util.number import *
from pwn import xor
flag = b'REDACTED'
key = b'REDACTED'
enc = b''
for i in range(len(flag)):
    enc += xor(key[i], flag[i])
print(enc)
# enc = b'#="5\x07\x1b\x01>4#s<u! \x1a3~3-\x1b7w7\x1b&4\x1a":)8'
```

According to the title, I guess the key is many DOGE.

```
from Crypto.Util.number import *
from pwn import xor
enc = b'#="5\x07\x1b\x01>4#s<u! \x1a3~3-\x1b7w7\x1b&4\x1a":)8'
key = b'DOGEDOGEDOGEDOGEDOGEDOGEDOGEDOGEDOGEDOGEDOGEDOGE'
flag = b''
for i in range(len(enc)):
    flag += xor(key[i], enc[i])
print(flag)
# enc = b'#="5\x07\x1b\x01>4#s<u! \x1a3~3-\x1b7w7\x1b&4\x1a":)8'
```



Bird Seed

```
#encrypt.py
import random
flag = open('flag.txt').read()

rand_seed = random.randint(0, 999)
random.seed(rand_seed)
encrypted = ''

for chr in flag:
    encrypted += f'{(ord(chr) ^ random.randint(0, 255)):02x}'

with open('out.txt', 'w') as f:
    f.write(encrypted)

#output
encrypted = a282b415279f5aa08cd4649515268910b8968a1eabda7c1bb2898c
```

1byte = 2 hex number. so change the encrypted result to one decimal number per two hex number.

```python
import random

Deci_Encrypted = [162, 130, 180, 21, 39, 159, 90, 160, 140, 212, 100, 149, 21, 38,
137, 16, 184, 150, 138, 30, 171, 218, 124, 27, 178, 137, 140]

for rand_seed in range(0, 999):
    random.seed(rand_seed)
    flag = ''
    for char in Deci_Encrypted:
        flag += chr(char ^ random.randint(0, 255))
    if "grep" in flag:
        print(flag)
```

```
lilon python -u "c:\Users\lilon\Downloads\encrypt.py"
grepCTF{n3v3r_tru1y_r4nd0m}
lilon ~
```

Uneasy Alliance

DESCRIPTION: You have seen people giving out p, q values and asking you to decrypt the cipher text. This time, you only have the cipher text. Good luck decrypting !

```python
from Crypto.Util.number import *
import math
import time
from random import Random

seed = math.floor(time.time())
rnd = Random(seed)

rand_fn = lambda n: long_to_bytes(rnd.getrandbits(n))
p = getPrime(128, randfunc=rand_fn)
q = getPrime(128, randfunc=rand_fn)
e = 65537
n = p * q

assert p != q

m = bytes_to_long(b"GREP{REDACTED}")
ct = pow(m, e, n)
print("Cipher text:", ct)
# Cipher text:
98987174569511481337499571065760296598797367073497107705609508485036141198 28
# Seed: REDACTED
```

because the seed is the timstamp, so burst way to try.

```python
from Crypto.Util.number import *
import math
```

```
import time
from random import Random

for seed in range(0,999999):
    rnd = Random(seed)

    rand_fn = lambda n: long_to_bytes(rnd.getrandbits(n))
    p = getPrime(128, randfunc=rand_fn)
    q = getPrime(128, randfunc=rand_fn)
    e = 65537

 Cipher=98987174569511481337499571065760296598797367073497107705609508485036141198 28
    n = p * q
    phi = (p-1)*(q-1)
    d = inverse(e,phi)
    plain=str(long_to_bytes(pow(Cipher,d,n)))
    if 'GREP' in plain:
        print(plain)
```

b'GREP{Brut3D_M3!_f0r_l1f3}'

OSINT

Sherlock Exhausted

DESCRIPTION: Holmes has reached 221B Baker Street after an exhausting day. A murder has happened, but there is no clue of the name of the murderer. Your task is to help Sherlock figure out the first name of the murderer.

I can know it's a logo of john GBA lite via google picture search, so the answer is **GREP{john}**

apogtspi

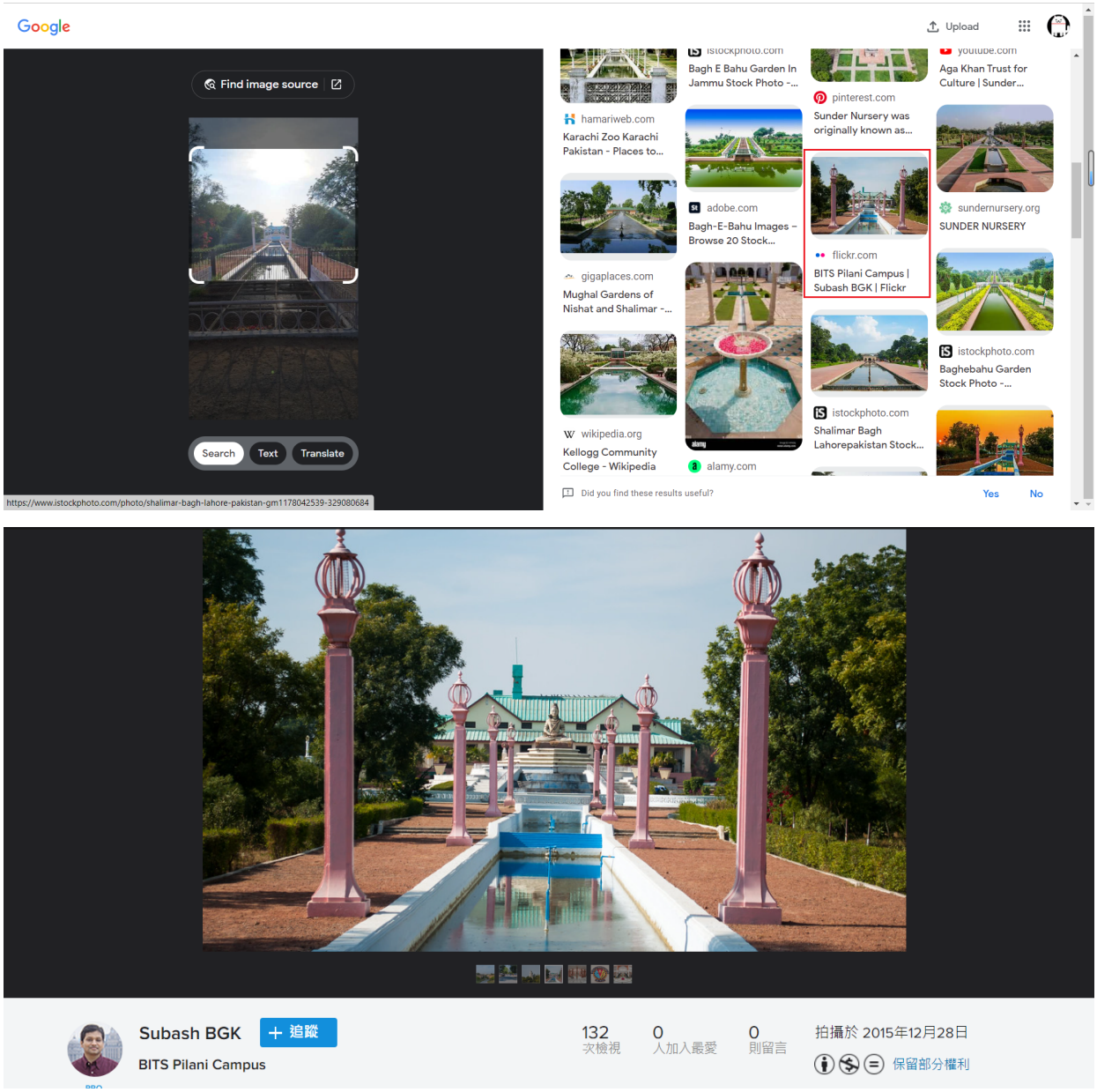
I think some times, and try to find some result via search engine. However I am fail. So I see the hint, Hint: Always remember the organizers. It's key. According to the hint, I focus on the organizers.

BITS PILANI & APOGEE 23'

So I try the flag: GREP{apogeebitspilani}, it's the right flag.

Don't Deviate

search this photo by google image search. Then find some photo in the internet is similar to the one I obtain.



But it's always wrong, (28.3560,75.5883)

I don't know what is the flag, but I think I have closed to it.