ISCC2021 MISC WP

原创

Atkxor 于 2021-05-26 00:06:53 发布 2349 收藏 9

分类专栏： WriteUp CTF 文章标签： ISCC

目录

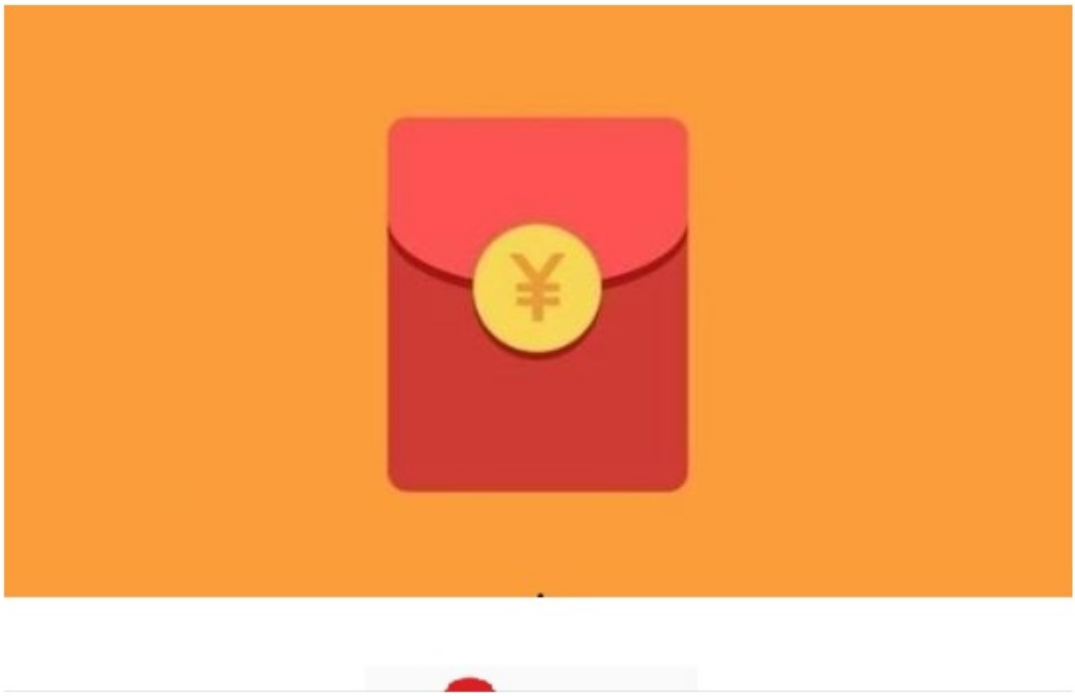
> 来自菜鸡的复现

练武

李华的红包

> 题目描述：
> 大年初一，李华给爸爸拜年，从事计算机行业的父亲发给李华一张图片和一张银行卡。父亲告诉李华密码就藏在图片中，但是李华打开图片后却百思不得其解。你能帮助李华拿到密码吗？

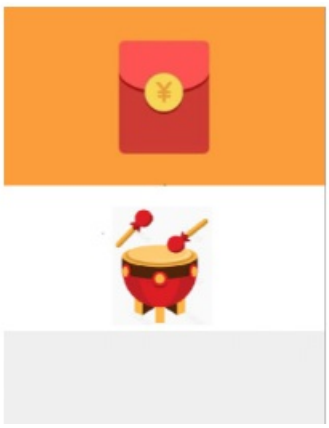
下载附件，



直接binwalk分离得到hongbao.txt

```
24,43,13,13,12,21,43
```

没啥思路，看起来图片不完全，修改图片高度，图片下面有一个鼓，联想到敲击码



对照敲击码表

转换得到`ISCCBFS`，有点坑，说是银行卡密码，其实并不是六位数字，最终flag为

```
ISCC{ISCCBFS}
```

Retrieve the passcode

> 题目描述：
> Scatter说他能解开这个古怪的密码，你呢？来试试吧！ Flag格式：ISCC{XXX}，XXX为小写字符串，不包括空格

xyz.txt

```
1:3:1
1.25:3:1
1.5:3:1
1.75:3:1
2:3:1
2:2.75:1
2:2.5:1
2:2.25:1
```

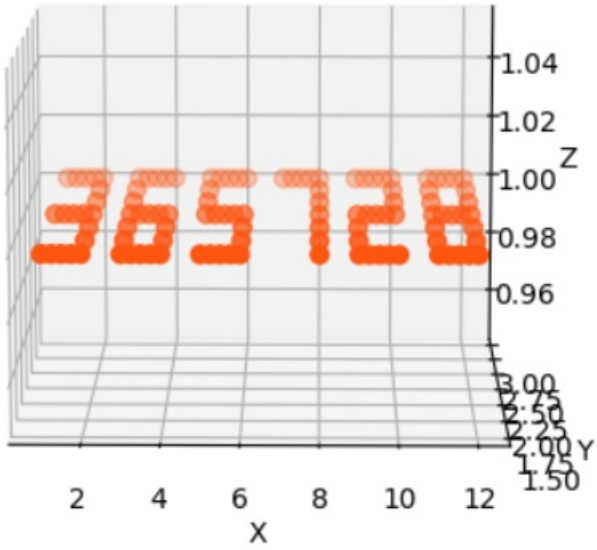
2:2:1
2:1.75:1
2:1.5:1
1:2.25:1
1.25:2.25:1
1.5:2.25:1
1.75:2.25:1
1:1.5:1
1.25:1.5:1
1.5:1.5:1
1.75:1.5:1
3:3:1
3.25:3:1
3.5:3:1
3.75:3:1
4:3:1
3.25:2.25:1
3.5:2.25:1
3.75:2.25:1
4:2.25:1
4:2:1
4:1.75:1
4:1.5:1
3:1.5:1
3.25:1.5:1
3.5:1.5:1
3.75:1.5:1
3:1.75:1
3:2:1
3:2.25:1
3:2.5:1
3:2.75:1
5:3:1
5.25:3:1
5.5:3:1
5.75:3:1
6:3:1
6:2.25:1
6:2:1
6:1.75:1
6:1.5:1
5.75:1.5:1
5.5:1.5:1
5.25:1.5:1
5:1.5:1
5:2.25:1
5.25:2.25:1
5.5:2.25:1
5.75:2.25:1
5:2.5:1
5:2.75:1
7:3:1
7.25:3:1
7.5:3:1
7.75:3:1
8:3:1
8:2.75:1
8:2.5:1
8:2.25:1
8:2:1

```
8:1.75:1
8:1.5:1
9:3:1
9.25:3:1
9.5:3:1
9.75:3:1
10:3:1
10:2.75:1
10:2.5:1
10:2.25:1
9.75:2.25:1
9.5:2.25:1
9.25:2.25:1
9:2.25:1
9:2:1
9:1.75:1
9:1.5:1
9.25:1.5:1
9.5:1.5:1
9.75:1.5:1
10:1.5:1
11:3:1
11.25:3:1
11.5:3:1
11.75:3:1
12:3:1
12:2.75:1
12:2.5:1
12:2.25:1
12:2:1
12:1.75:1
12:1.5:1
11.75:1.5:1
11.5:1.5:1
11.25:1.5:1
11:1.5:1
11:1.75:1
11:2:1
11:2.25:1
11:2.5:1
11:2.75:1
11.25:2.25:1
11.5:2.25:1
11.75:2.25:1
```

百度找到的脚本，修改一下：

```
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure()
ax = fig.gca(projection="3d")


xs, ys ,zs= np.loadtxt('xyz.txt', delimiter=':', unpack=True)
ax.scatter(xs, ys, zs, zdir="z", c="#FF5511", marker="o", s=40)
ax.set(xlabel="X", ylabel="Y", zlabel="Z")

plt.show()
```

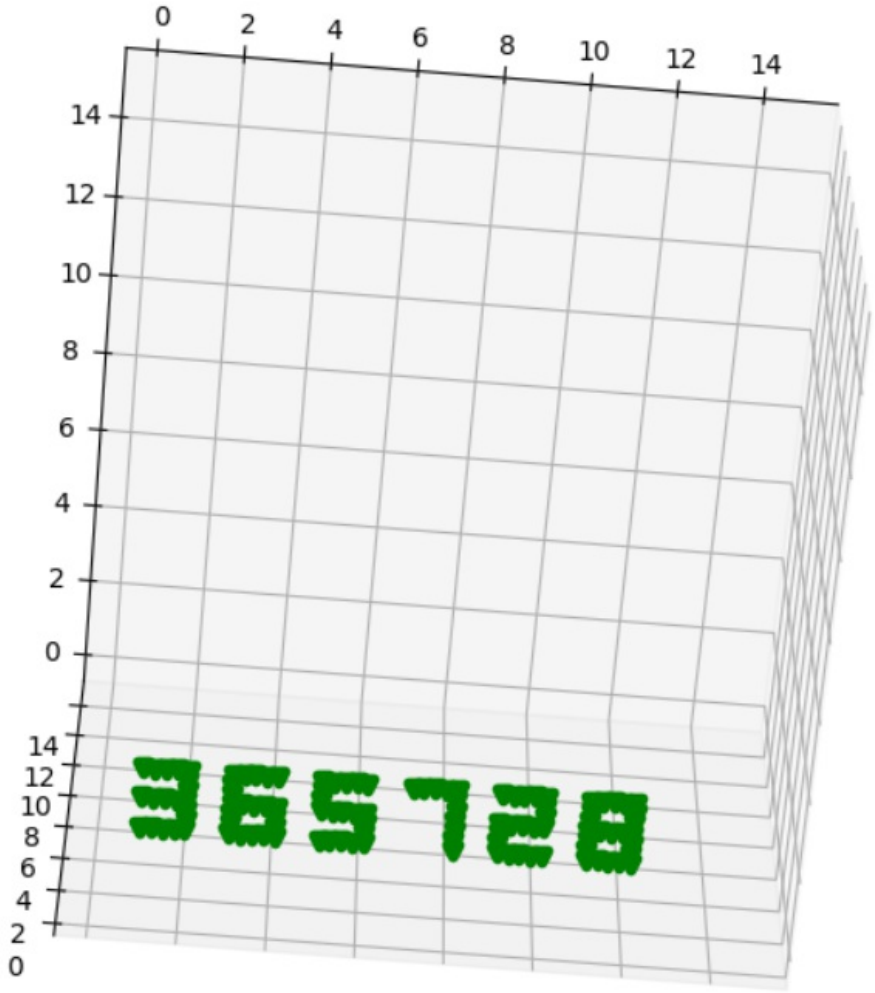
得到



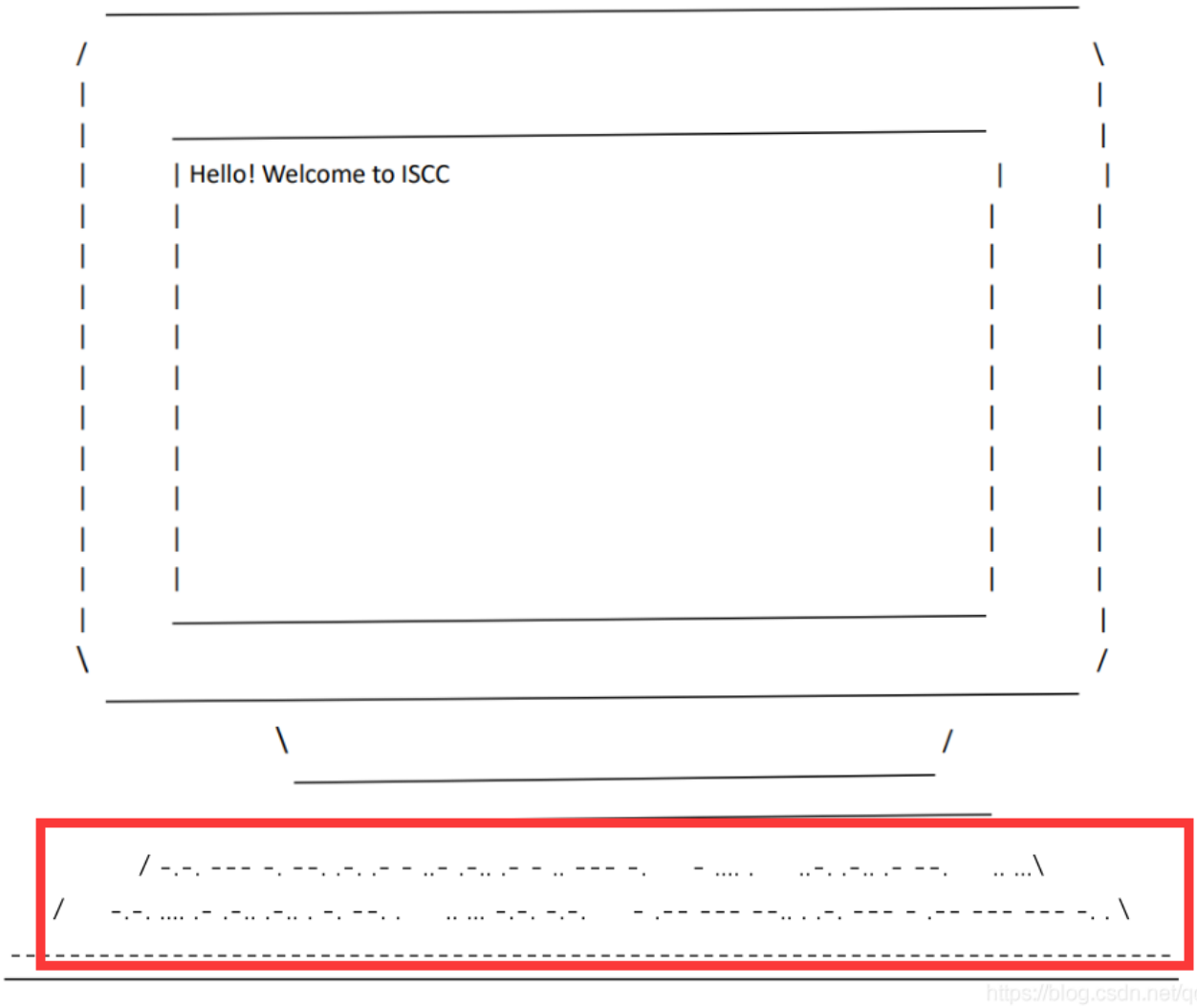
脚本2：

```python
from matplotlib import pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
dot1 = [[1, 3, 1], [1.25, 3, 1], [1.5, 3, 1], [1.75, 3, 1], [2, 3, 1], [2, 2.75, 1], [2, 2.5, 1], [2, 2.25, 1],
[2, 2, 1], [2, 1.75, 1], [2, 1.5, 1], [1, 2.25, 1], [1.25, 2.25, 1], [1.5, 2.25, 1], [1.75, 2.25, 1], [1, 1.5, 1
], [1.25, 1.5, 1], [1.5, 1.5, 1], [1.75, 1.5, 1], [3, 3, 1], [3.25, 3, 1], [3.5, 3, 1], [3.75, 3, 1], [4, 3, 1],
 [3.25, 2.25, 1], [3.5, 2.25, 1], [3.75, 2.25, 1], [4, 2.25, 1], [4, 2, 1], [4, 1.75, 1], [4, 1.5, 1], [3, 1.5,
1], [3.25, 1.5, 1], [3.5, 1.5, 1], [3.75, 1.5, 1], [3, 1.75, 1], [3, 2, 1], [3, 2.25, 1], [3, 2.5, 1], [3, 2.75,
 1], [5, 3, 1], [5.25, 3, 1], [5.5, 3, 1], [5.75, 3, 1], [6, 3, 1], [6, 2.25, 1], [6, 2, 1], [6, 1.75, 1], [6, 1
.5, 1], [5.75, 1.5, 1], [5.5, 1.5, 1], [5.25, 1.5, 1], [5, 1.5, 1], [5, 2.25, 1], [5.25, 2.25, 1], [5.5, 2.25, 1
], [5.75, 2.25, 1], [5, 2.5, 1], [5, 2.75, 1], [7, 3, 1], [7.25, 3, 1], [7.5, 3, 1], [7.75, 3, 1], [8, 3, 1], [8
, 2.75, 1], [8, 2.5, 1], [8, 2.25, 1], [8, 2, 1], [8, 1.75, 1], [8, 1.5, 1], [9, 3, 1], [9.25, 3, 1], [9.5, 3, 1
], [9.75, 3, 1], [10, 3, 1], [10, 2.75, 1], [10, 2.5, 1], [10, 2.25, 1], [9.75, 2.25, 1], [9.5, 2.25, 1], [9.25,
 2.25, 1], [9, 2.25, 1], [9, 2, 1], [9, 1.75, 1], [9, 1.5, 1], [9.25, 1.5, 1], [9.5, 1.5, 1], [9.75, 1.5, 1], [1
0, 1.5, 1], [11, 3, 1], [11.25, 3, 1], [11.5, 3, 1], [11.75, 3, 1], [12, 3, 1], [12, 2.75, 1], [12, 2.5, 1], [12
, 2.25, 1], [12, 2, 1], [12, 1.75, 1], [12, 1.5, 1], [11.75, 1.5, 1], [11.5, 1.5, 1], [11.25, 1.5, 1], [11, 1.5,
 1], [11, 1.75, 1], [11, 2, 1], [11, 2.25, 1], [11, 2.5, 1], [11, 2.75, 1], [11.25, 2.25, 1], [11.5, 2.25, 1], [
11.75, 2.25, 1]]  # 得到五个点
plt.figure()  # 得到画面
ax1 = plt.axes(projection='3d')
ax1.set_xlim(0, 15)  # X轴，横向向右方向
ax1.set_ylim(15, 0)  # Y轴,左向与X,Z轴互为垂直
ax1.set_zlim(0, 15)  # 竖向为Z轴
color1 = ['r', 'g', 'b', 'k', 'm']
marker1 = ['o', 'v', '1', 's', 'H']
i = 0
for x in dot1:
    ax1.scatter(x[0], x[1], x[2], c=color1[1],
                marker=marker1[1], linewidths=4)  # 用散点函数画点
    i += 1
plt.show()
```

运行得到



密码口令是`365728`，解压压缩包，得到pdf文档



摩斯密码解密

```
CONGRATULATIONTHEFLAGICHALLENGEISCCTWOZEROTWOONE
```

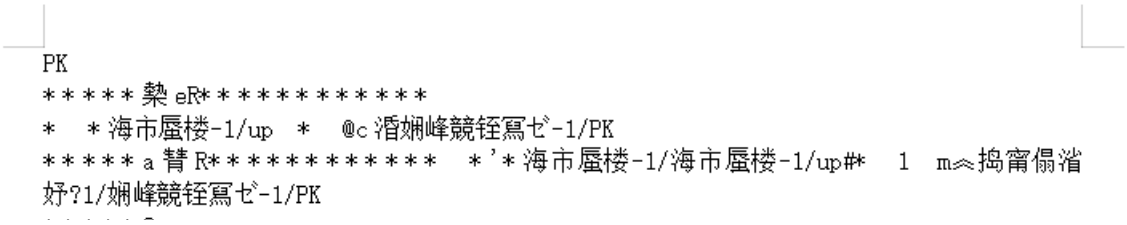
根据提示最终flag为

```
ISCC{congratulationtheflagischallengeiscctwozerotwoone}
```

海市蜃楼-1

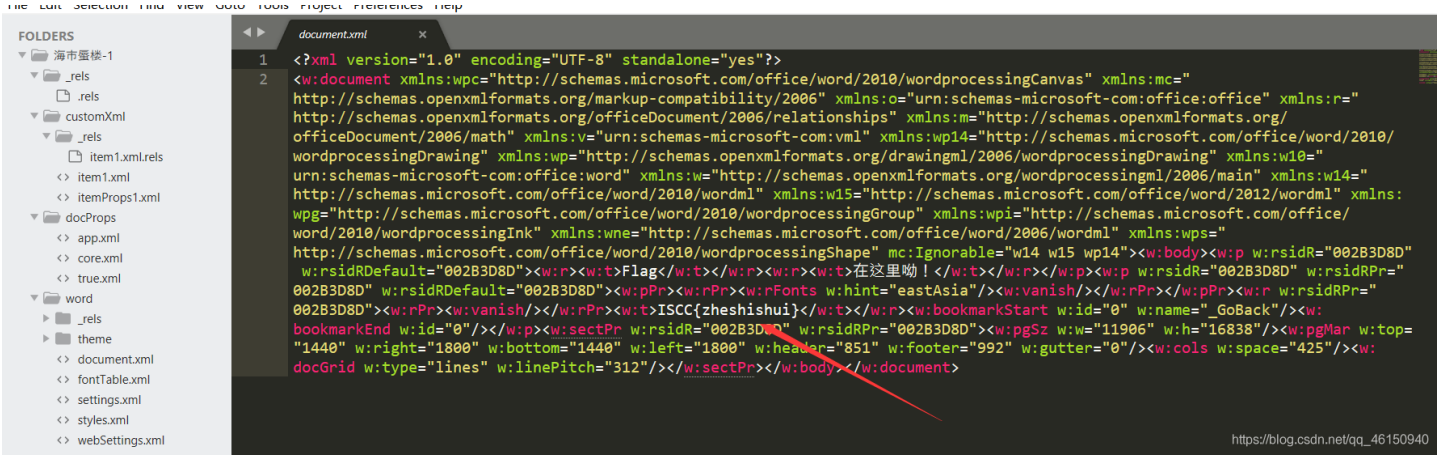
> 题目描述： 或许你看到的只是海市蜃楼...

下载附件，是个docx文档，看到了压缩包文件头PK



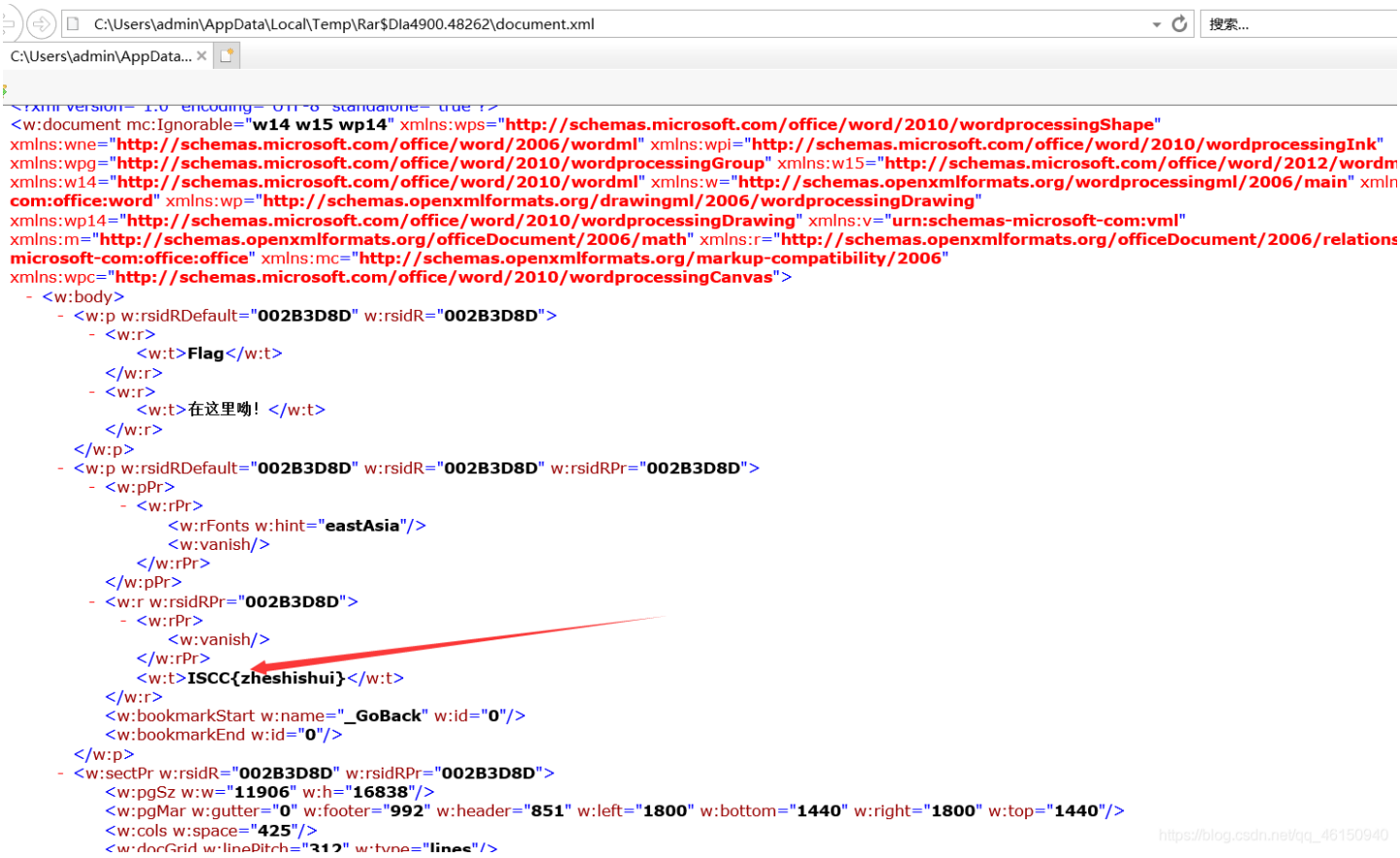
*****?

修改后缀为zip，解压，在document.xml文件中找到flag



方法二：

使用winrar在文件中搜索ISCC字符串



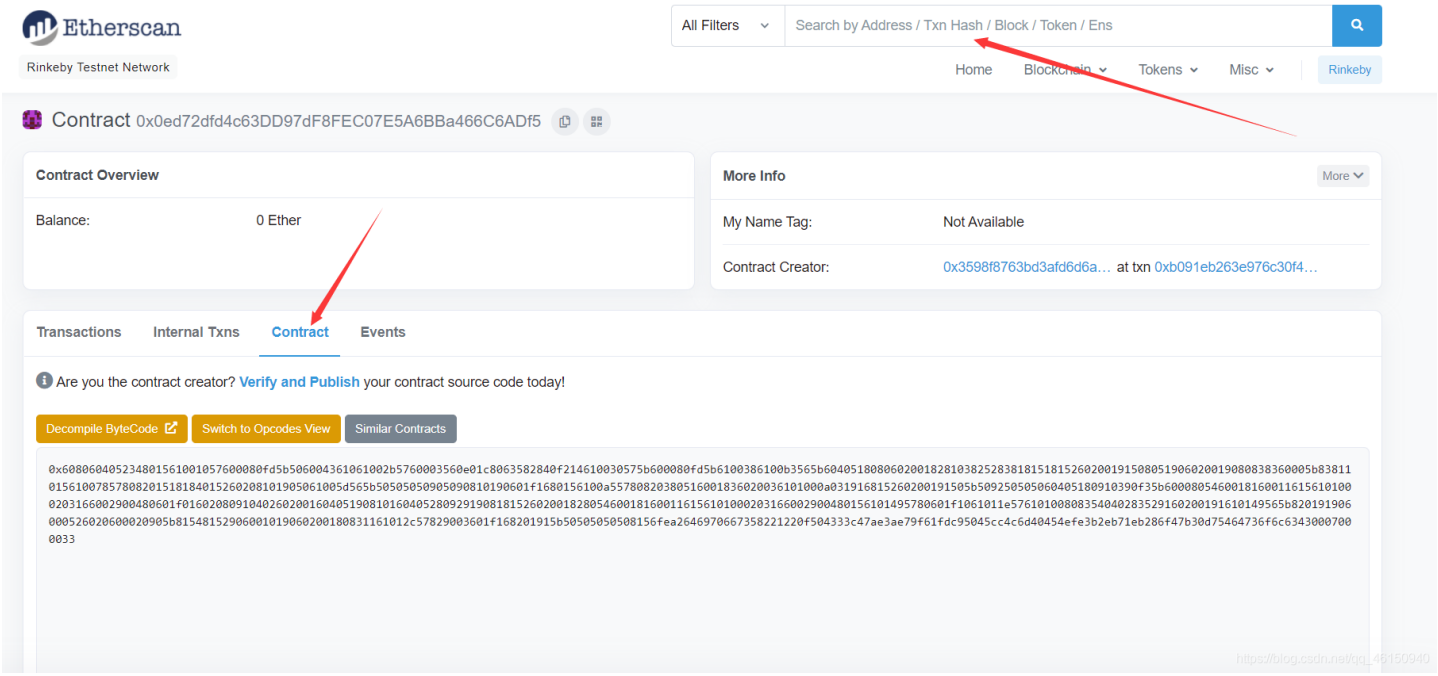
区块链

参考：https://terminalcats.com/0x41414141-ctf-blockchain-sanity-check-400/

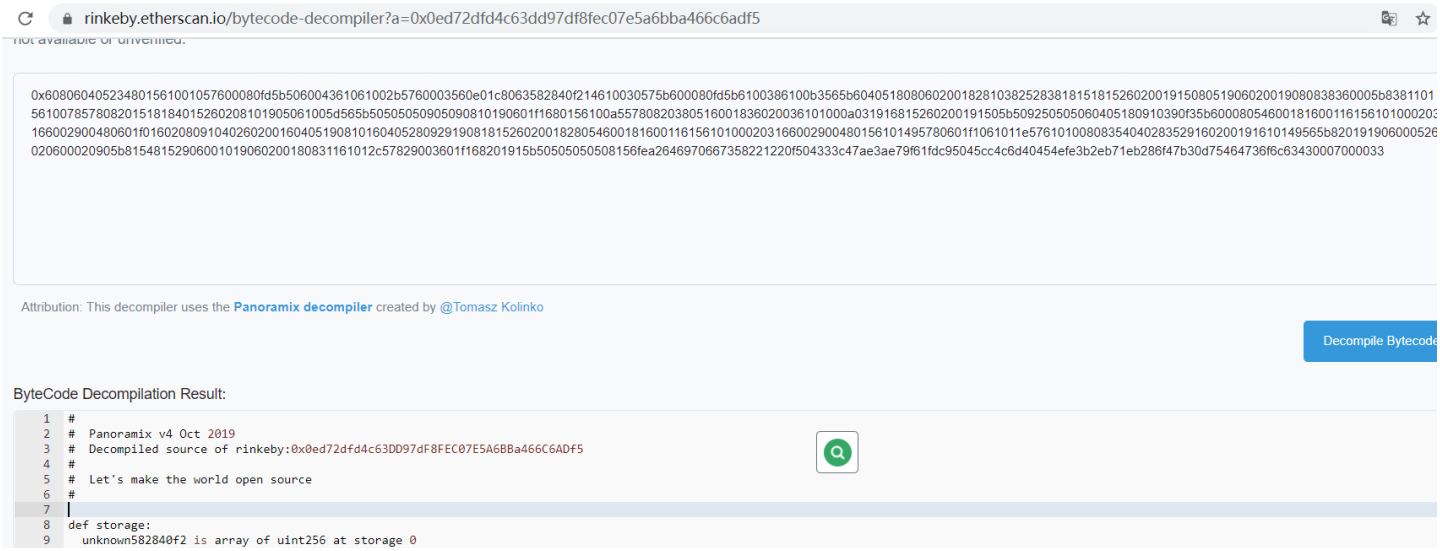
```
pragma solidity ^0.7.0;  // 指定所需的编译器版本
//SPDX-License-Identifier: UNLICENSED

contract look_look {
    function sloc111() public pure returns (string memory){    //public智能合约外部和内部都可使用的方法
        return "flag{}";
    }
}



合约地址：0x0ed72dfd4c63dd97df8fec07e5a6bba466c6adf5
```

需要挂代理，访问https://rinkeby.etherscan.io/

第一次做智能合约方面的题，输入合约地址，然后点击Contract

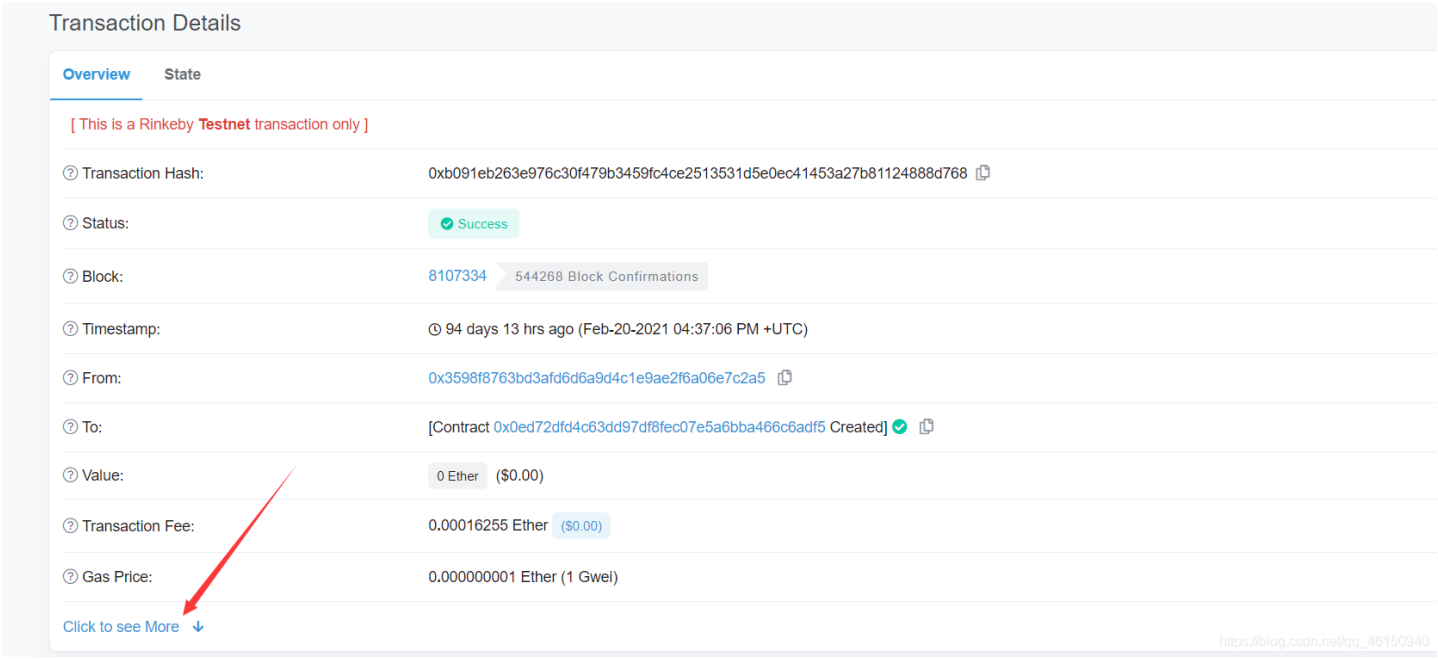


点击Decompile ByteCode，到这就卡住了

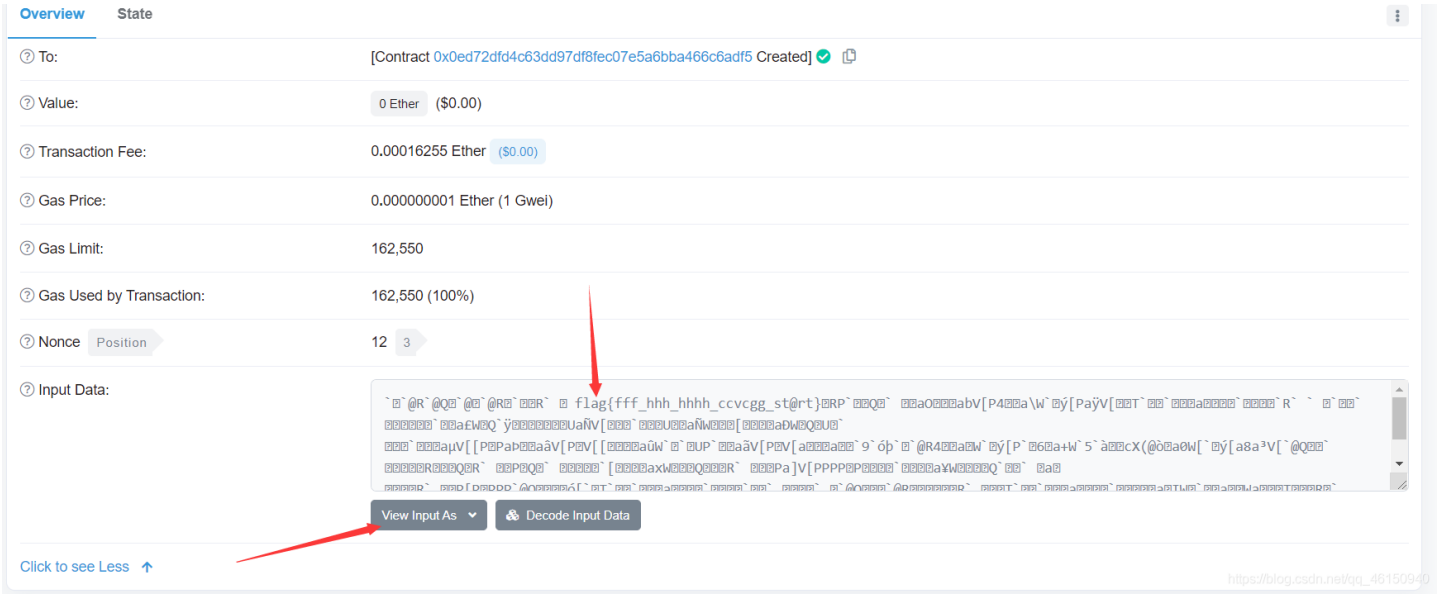


```
10
11  def unknown582840f2() payable:
12    return unknown582840f2[0 len unknown582840f2.length]
13
14  #
15  #  Regular functions
16  #
17
18  def _fallback() payable: # default function
19    revert
20
21
```

星期天打了个比赛，里面也有个区块链题-AZ。公链明文泄露



选择UTF-8 查看,得到flag



美人计

> 题目描述：美人说的话里有解题提示，但是美人的话不能全信。

我记得当时附件文档中有两个二维码，二维码内容还保存着，图片不见了，不知道怎么回事

```
第一个二维码：
U2FsdGVkX1/Ka+sScszwQkwhO+VLiJwV/6IFg5W+TfNHGxG2qZsIr2iwMwb9X9Iu
3GuGWmPOtO27z8vNppD2D50fwsD+8VWhdtW9J4cewYivH/Z/7GoUvcJXJMrvf+vu
+CBqWDGp6HWd0e5whGhuzlK0ZtBcDZdPDSIHA7+GuUlifp8PcFCtJPgiuk143REE
+pKFiSJXolXLR1vJCdGY9w5mXFbiWPrb2U7r/v5noP8=

第二个二维码：
U2FsdGVkX19eOY/pDh8+vPAcvfkLi1XLUneVzjLLOMul53sKK8UpobdCOiPIv4KE
```

图片属性给的提示，AES加密，密钥是ISCC2021

```
Try AES, and you will get the flag. ISCC2021
```

对第一个二维码内容，解了三次AES加密得到

```
y0u_h@ve_fal1en_intO_tHe_tr@p_0f_tHe_be@uty_!
```

第二个二维码内容，进行AES解密发现不行，然后尝试DES解密结果成功了，得到flag

```
ISCC{be@uty_1ike$_Y0u_2O2lISCC}
```

再复现的时候，附件中就剩一张二维码了，把docx后缀转换成zip解压，出题人变狗了，把第二张二维码藏在这了，和上面的方法一样，得到flag

我的折扣是多少

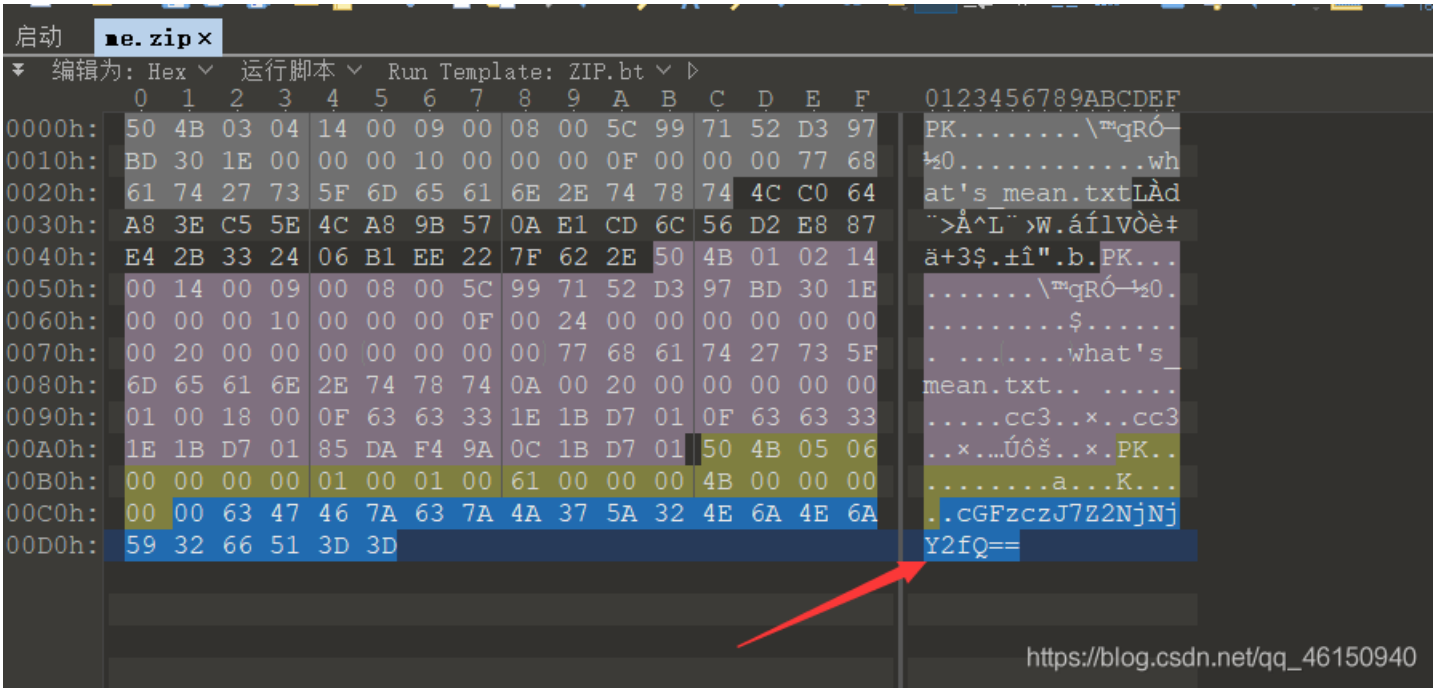
> 题目描述：
> 小c同学去参加音乐会，在官网买票时发现了有提示消息，提供给的有“give_me_discount”的压缩包，好奇的小c下载下来，但却无从下手，为了节省零花钱，你能帮帮他吗？

命令行运行give.exe，得到

```
pass1{\u006b\u0072\u0077}
```

把中间的进行unicode转码得到：pass1{krw}

010打开me.zip，在末尾发现Base编码



base64解码得到：pass2{gcc666}

所以压缩包口令为krwgcc666，解压得到

```
eW91Zm91bmRtZT8=
```

base64解码

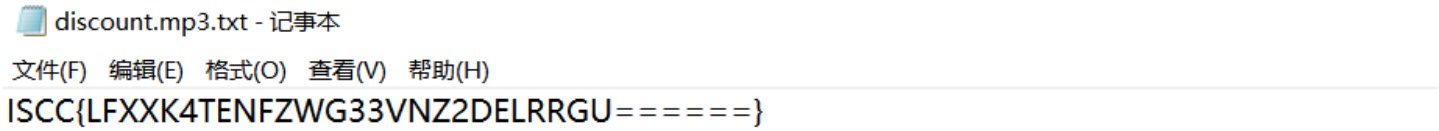
eW91Zm91bmRtZT8=

清空 加密 解密 解密结果以16进制显示

youfoundme?

最后使用MP3Stego

```
decode -X -P youfoundme? discount.mp3
```



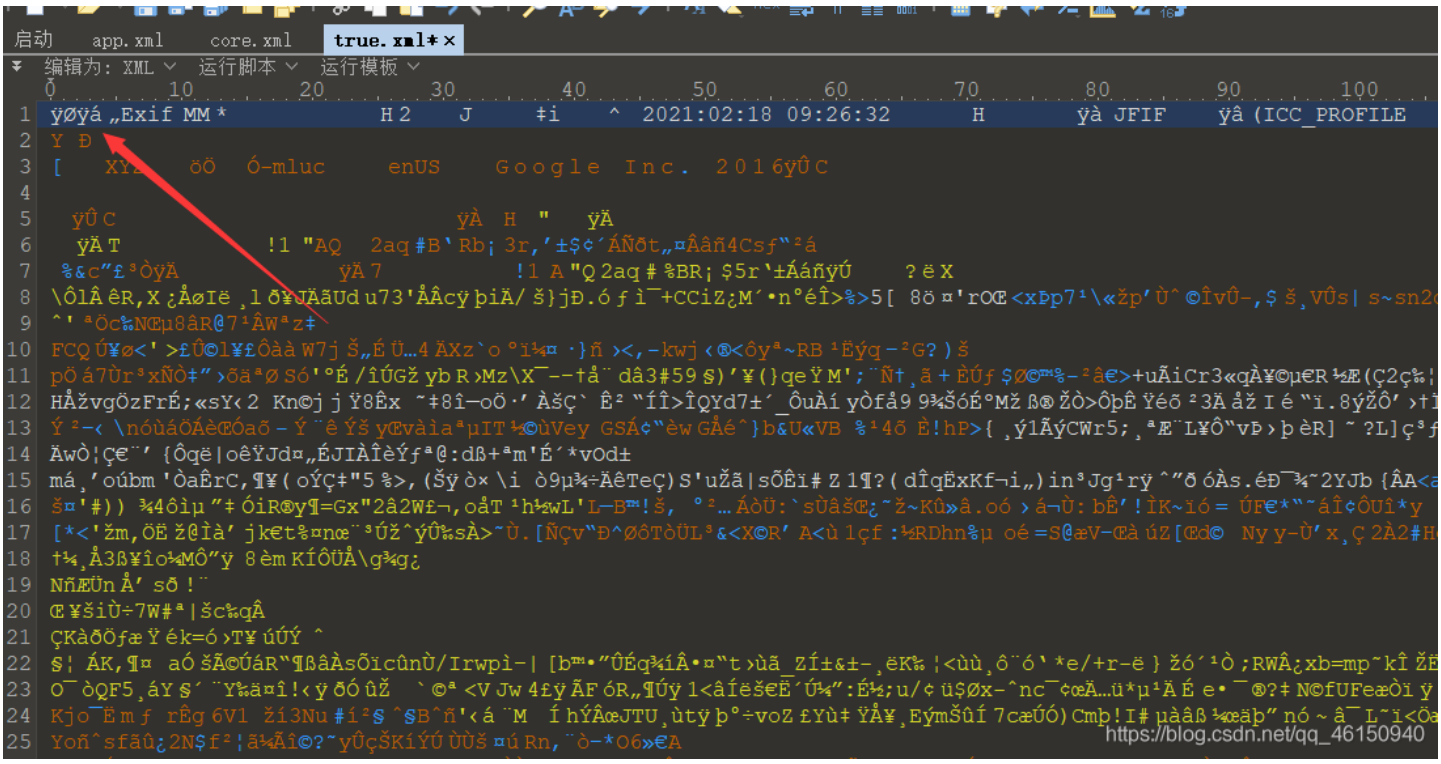
base32解码，最终flag为

```
ISCC{Yourdiscount2.15}
```

海市蜃楼-2

foremost分离图片，得到加密的压缩包

由于这是组合题，所以要在海市蜃楼-1中去找线索。最终发现true.xml有猫腻，原来是jpg文件



修改为jpg

海市蜃楼-1的flag也有猫腻，结合flag和图片，最终得到压缩包的口令为zhongnanshan，解压海市蜃楼-2的压缩包

扫码得到一个网址，内容为`ZWFzeQ==`，base解码后为`easy`

```
https://ctewm.com/aTvQcE/kva11w
```

继续往下走，foremost分离图片得到压缩包，口令为easy

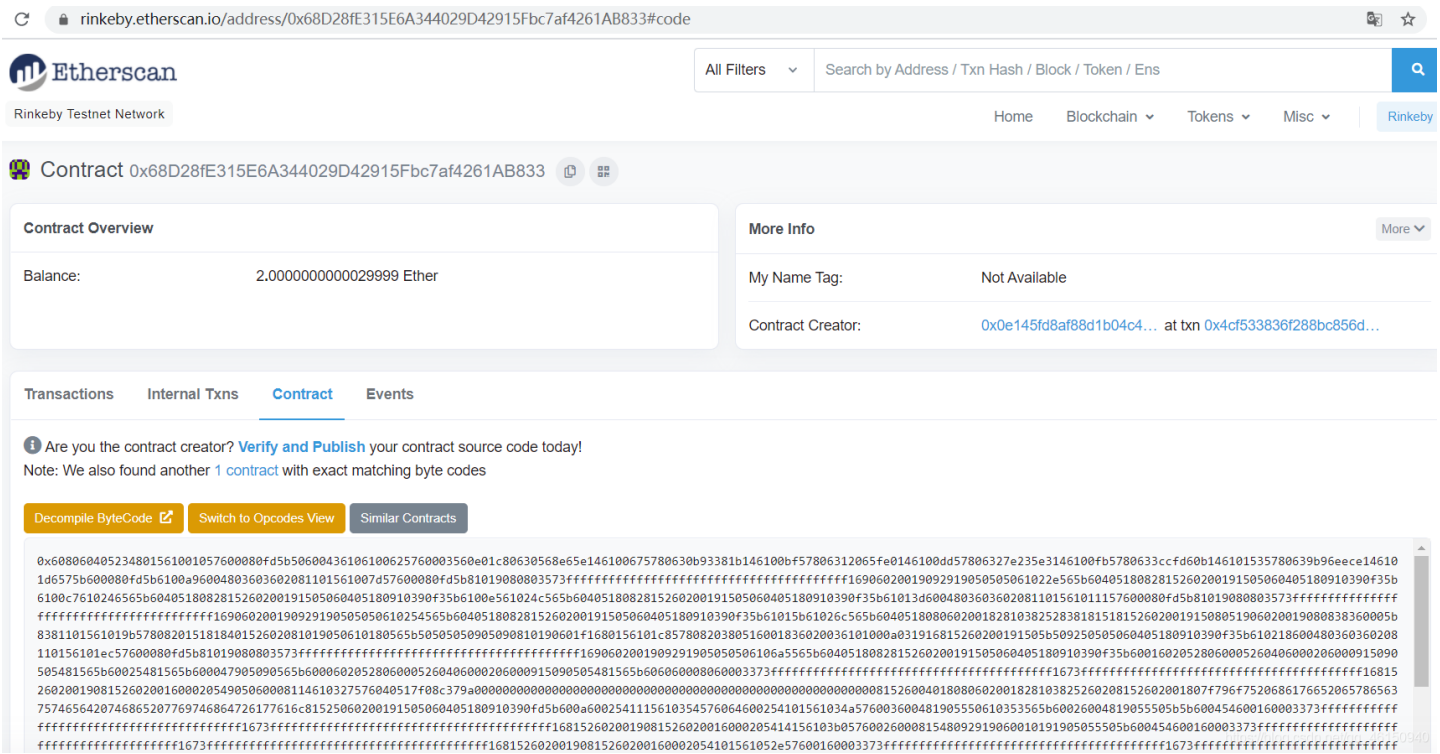
Hack the Victim

题目

```
Victim为含有漏洞的智能合约，在 Rinkeby测试网络的合约地址为：
0x68D28fE315E6A344029D42915Fbc7af4261AB833
接口为：
contract Victim {
function withdraw() public returns (string memory ){
return "ISCC{xxxxx}";
}
}
请编写攻击合约，实现对 Victim 的攻击，获取 flag。
```

合约地址

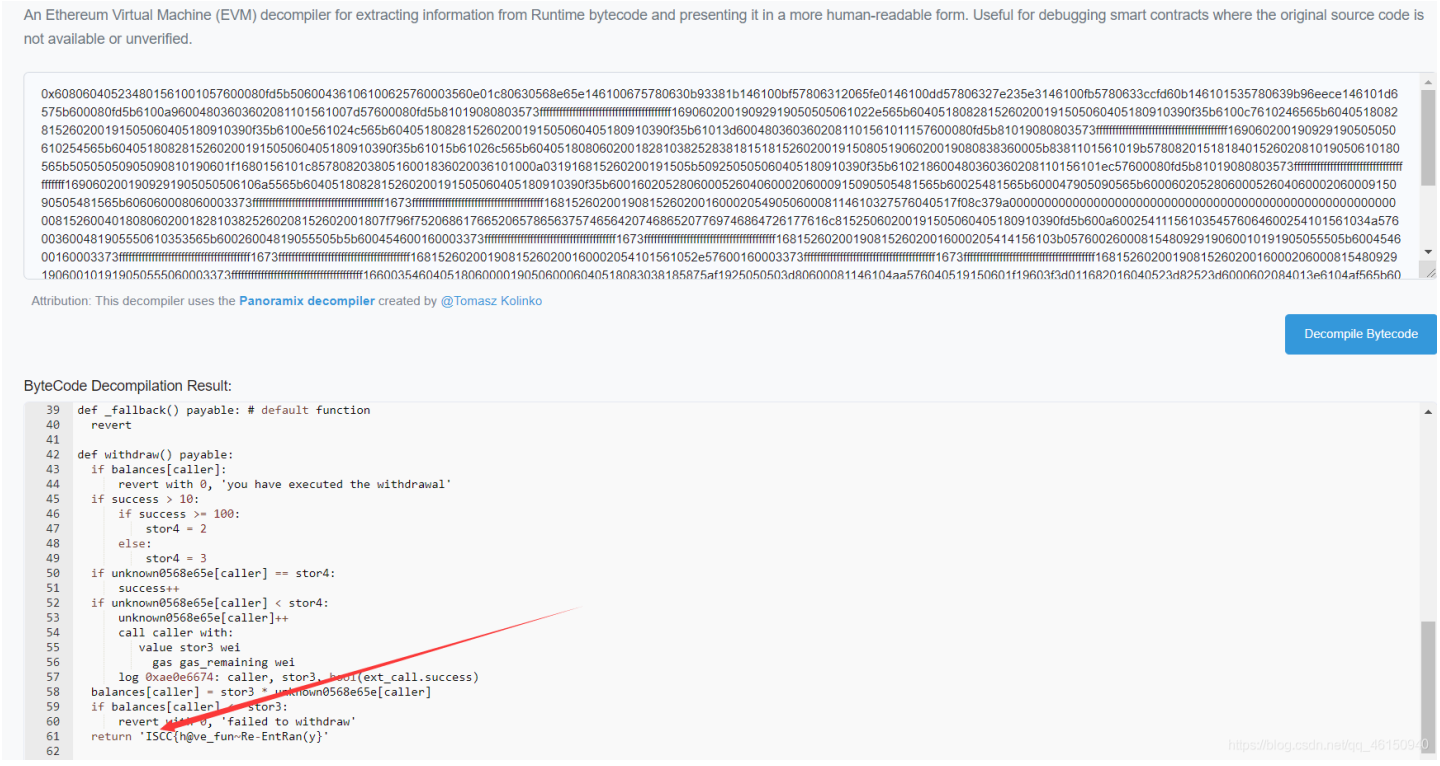
```
https://rinkeby.etherscan.io/address/0x68D28fE315E6A344029D42915Fbc7af4261AB833
```



16进制转字符串

16进制到文本字符串

也可以直接点击Decompile ByteCode进行反编译



检查一下

> 题目描述：你真的了解png文件的格式吗？

用pngcheck检查一下图片，发现最后一个IDAT块异常

C:\Windows\System32\cmd.exe

```
  chunk IDAT at offset 0x742bba, length 8192
  chunk IDAT at offset 0x744bc6, length 8192
  chunk IDAT at offset 0x746bd2, length 8192
  chunk IDAT at offset 0x748bde, length 8192
  chunk IDAT at offset 0x74abea, length 8192
  chunk IDAT at offset 0x74cbf6, length 8192
  chunk IDAT at offset 0x74ec02, length 8192
  chunk IDAT at offset 0x750c0e, length 8192
  chunk IDAT at offset 0x752c1a, length 8192
  chunk IDAT at offset 0x754c26, length 6181
  chunk IDAT at offset 0x756457, length 178
  chunk IEND at offset 0x756515, length 0
No errors detected in 221B.png (942 chunks, 79.0% compression).

D:\CTF\MISC\隐写\pngcheck-3.0.2-win32>
```


010复制出来

```
75:6430h:  99 A7 60 C4 6C 4F A8 86 E9 C1 0B 69 A4 00 7D 45   ™§`ÄlO¨†éÁ.i¤.}E
75:6440h:  42 FF D2 E1 17 20 F6 A9 FC 7F 01 17 2D 1B 01 E6   BÿÒá. ö©ü...-..æ
75:6450h:  2F 0F 17 00 00 00 B2 49 44 41 54 78 9C 5D 52 09   /.....²IDATxœ]R.
75:6460h:  0E C0 20 0C FA 12 FC FF 73 CB 6A 39 54 97 4C 7B   .À .ú.üÿsËj9T—L{
75:6470h:  41 2B E4 2C 00 3C DF 7F 04 69 B3 AC E7 CA D9 27   A+ä,.<ß..i³¬çÊÙ'
75:6480h:  E4 37 1D 27 15 33 01 E3 91 75 AB 9D F4 D7 29 F3   ä7.'.3.ã'u«.ô×)ó
75:6490h:  E2 AF 13 75 21 CC AC 69 3D BB C8 0A 2F 8C E0 26   â¯.u!Ì¬i=»È./Œà&
75:64A0h:  D0 25 F9 F4 62 A2 08 C5 0A A0 69 90 35 0D 34 E1   Ð%ùôb¢.Å. i.5.4á
75:64B0h:  39 8B 8B 2A 5D A0 CA DE B4 FE 23 61 59 CB 60 3B   9‹‹*] ÊÞ´þ#aYË`;
75:64C0h:  EE DA 82 5E 2E 1E E2 46 15 D8 8D AE 46 33 54 95   îÚ‚^..âF.Ø.®F3T•
75:64D0h:  23 8A AD 82 FD C0 F5 CC 86 2C 76 69 1F 34 74 8D   #Š-‚ýÀõÌ†,vi.4t.
75:64E0h:  5F 22 70 2B 86 8A CC FA 69 EA 6D A4 12 DC 99 A5   _"p+†ŠÌúiêm¤.Ü™¥
75:64F0h:  21 08 B9 27 5F F3 37 40 A4 8C AA E7 1A A5 BE F4   !.¹'_ó7@¤Œªç.¥¾ô
75:6500h:  AB E2 A5 B3 68 21 23 C6 07 2E D4 9F 5D E1 77 9E   «â¥³h!#Æ..ÔŸ]áwž
75:6510h:  9A 00 00 00 00 49 45 4E 44 AE 42 60 82            š....IEND®B`‚
```

查找结果

地址	值
746BD2h	IDAT
748BDEh	IDAT
74ABEAh	IDAT
74CBF6h	IDAT
74EC02h	IDAT
750C0Eh	IDAT
752C1Ah	IDAT
754C26h	IDAT
939 756457h	IDAT

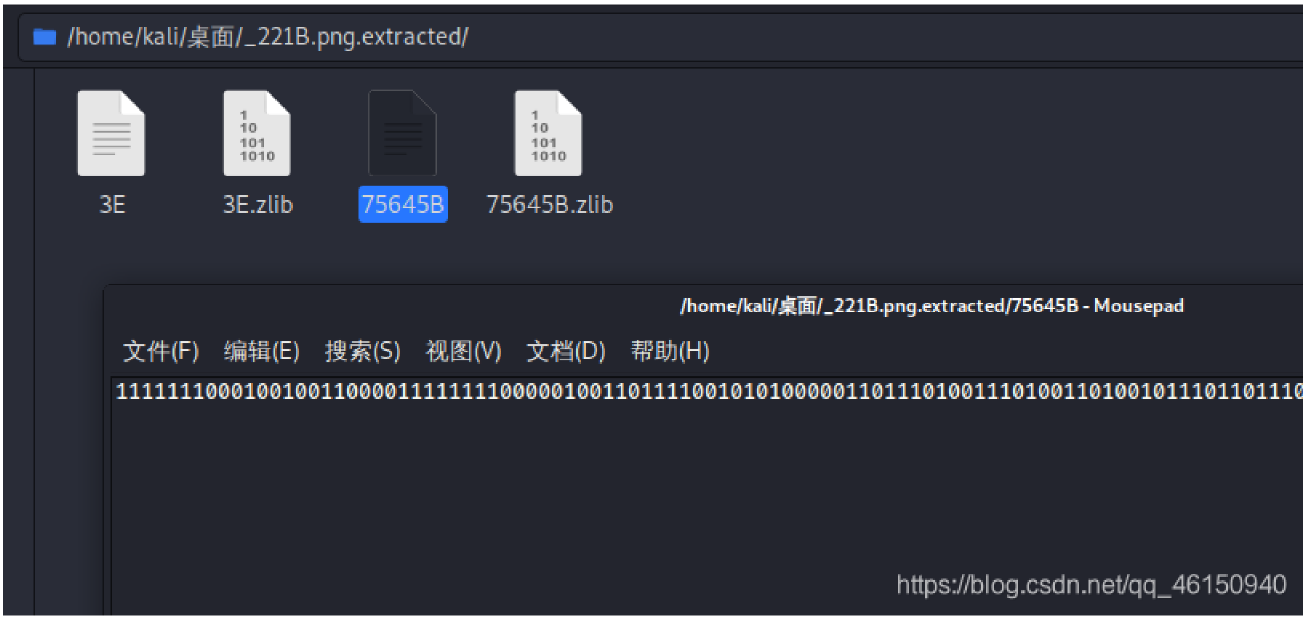
使用脚本解压zlib

```python
import zlib
s = '''
78 9C 5D 52 09 0E C0 20 0C FA 12 FC FF 73 CB 6A
39 54 97 4C 7B 41 2B E4 2C 00 3C DF 7F 04 69 B3
AC E7 CA D9 27 E4 37 1D 27 15 33 01 E3 91 75 AB
9D F4 D7 29 F3 E2 AF 13 75 21 CC AC 69 3D BB C8
0A 2F 8C E0 26 D0 25 F9 F4 62 A2 08 C5 0A A0 69
90 35 0D 34 E1 39 8B 8B 2A 5D A0 CA DE B4 FE 23
61 59 CB 60 3B EE DA 82 5E 2E 1E E2 46 15 D8 8D
AE 46 33 54 95 23 8A AD 82 FD C0 F5 CC 86 2C 76
69 1F 34 74 8D 5F 22 70 2B 86 8A CC FA 69 EA 6D
A4 12 DC 99 A5 21 08 B9 27 5F F3 37 40 A4 8C AA
E7 1A A5 BE F4 AB E2 A5 B3 68 21 23 C6 07 2E D4
9F 5D E1 77 9E 9A
'''
s = s.replace(' ','').replace('\n','')
b = bytes.fromhex(s)
flag = zlib.decompress(b)
print(flag)
```

得到

```
1111111000100100110000111111110000010011011110010101000001101110100111010011010010111011011101000000111011110101
1101101110100101111011100010111011000001011101110000100100000111111101010101010101011111110000000001110010011101
0000000010010110111000010101010100000110100001110101101101110010100000111001001000111101111001000110000010110110
0111111001010110101101110000100011110100110001001100010010101110000111111111100101001101010011111100101100011100
1101111101110001100111100100011110011100001000110100001101001000000001011001010101101110100011011010011100011101 0
0111101111110001010011011011011001010100011111010000000000111011011010100010110111111100000111100001010110101000 0
0101011110010101000101101011101001010011001011111001010111010110101111000001101001101110100101010000010100111011
0000010000111010000110010010111111101001001000000111100110
```

直接binwalk图片，也能得到同样的结果



一共841位，猜测是29*29的二维码，直接使用脚本

```
from PIL import Image

MAX = 29
pic = Image.new("RGB",(MAX, MAX))
str = "11111110001001001100001111111100000100110111100101010000011011101001110100110100101110110111010000001110111101110100101111011100010111011000001011101111010000111111110101010101010101111111000000000111001001101000000001001011011100001010101010000011010000111010110110111001010000011100100100011110111100100011000011101100111111100101011010110111000010001111010011000100110001001010111000011111111111001010011010100111111001011000110111111011100011001111001000111100111000010001101000011010010000000101100101010110111010001101101011010000111110100000000011101101101010001011011111110000011110000101011010000010101111001010100010111010111010010100110010111110010101110101101011110000011010011011101001010100000101011110010101000101110101110100101001100101111100101011101011010111100000110100110111010010101000001010011100001000011101000011001001011111110100100100000111100110"
i=0
for y in range (0,MAX):
    for x in range (0,MAX):
        if(str[i] == '1'):
            pic.putpixel([x,y],(0, 0, 0))
        else:
            pic.putpixel([x,y],(841,841,841))
        i = i+1
pic.show()
pic.save("flag.png")
```

扫码得到flag

变异的SM2

附件

```
#server.py
from gmssl import func, sm2
# from flag import FLAG
FLAG="{testFlag}"

sm2p256v1_ecc_table = {
    'n': 'FFFFFFFEFFFFFFFFFFFFFFFFFFFFFFFF7203DF6B21C6052B53BBF40939D54123',
    'p': 'FFFFFFFEFFFFFFFFFFFFFFFFFFFFFFFF00000000FFFFFFFFFFFFFFFF',
    'g': '32c4ae2c1f1981195f9904466a39c9948fe30bbff2660be1715a4589334c74c7' +
         'bc3736a2f4f6779c59bdcee36b692153d0a9877cc62a474002df32e52139f0a0',
    'a': 'FFFFFFFEFFFFFFFFFFFFFFFFFFFFFFFF00000000FFFFFFFFFFFFFFFC',
    'b': '28E9FA9E9D9F5E344D5A9E4BCF6509A7F39789F515AB8F92DDBCBD414D940E93',
}
n = 'FFFFFFFEFFFFFFFFFFFFFFFFFFFFFFFF7203DF6B21C6052B53BBF40939D54123'
G = '32c4ae2c1f1981195f9904466a39c9948fe30bbff2660be1715a4589334c74c7' \
    'bc3736a2f4f6779c59bdcee36b692153d0a9877cc62a474002df32e52139f0a0'

def sign(tsm2):
    data = func.random_hex(len(n))
    k1_str = func.random_hex(len(n))
    print(tsm2.send_p1(data, k1_str))
    backdoor = input('backdoor:').strip()
    result = tsm2.output_p1(k1_str, backdoor)
    print(result)

def verify(tsm2):
    message = input('msg:').strip().encode().strip(b'\x00')
    sign = input('sign:').strip().encode().strip(b'\x00')
    check = tsm2.verify(sign, message)
    if check is True and message == b'Hello, Welcome to ISCC2021!':
        print(FLAG)
```

```
        print(FLAG)
    else:
        print(check)

class TSM2(object):
    def __init__(self, sk):
        ecc_table = sm2p256v1_ecc_table
        self.ecc_table = ecc_table
        self.n = int(ecc_table['n'], 16)
        self.para_len = len(ecc_table['n'])
        self.ecc_a3 = (int(ecc_table['a'], base=16) + 3) % int(ecc_table['p'], base=16)

        self.sk = int(sk, 16)
        self.pk = self._kg(self.sk, ecc_table['g'])

        self.sks = int(func.random_hex(self.para_len), 16)
        self.pks = pow((self.sk + 1) * self.sks, self.n - 2, self.n) % self.n

    def send_p1(self, data, k1_str):
        e = int(data, 16)
        k1 = int(k1_str, 16)
        k1 = k1 % self.n
        R1 = self._kg(k1, self.ecc_table['g'])
        return '%064x%0128s' % (e, R1)

    def output_p1(self, k1_str, r_s2_s3):
        r = int(r_s2_s3[0:self.para_len], 16)
        s2 = int(r_s2_s3[self.para_len:2 * self.para_len], 16)
        s3 = int(r_s2_s3[2 * self.para_len:], 16)

        k1 = int(k1_str, 16)
        d1 = self.sks
        s = (d1 * k1 * s2 + d1 * s3 - r) % self.n
        if s == 0 or s == (self.n - r):
            return None
        return '%064x%064x' % (r, s)

    def verify(self, Sign, data):
        r = int(Sign[0:self.para_len], 16)
        s = int(Sign[self.para_len:2 * self.para_len], 16)
        e = int(data.hex(), 16)
        t = (r + s) % self.n
        if t == 0:
            return 0

        P1 = self._kg(s, self.ecc_table['g'])
        P2 = self._kg(t, self.pk)

        if P1 == P2:
            P1 = '%s%s' % (P1, 1)
            P1 = self._double_point(P1)
        else:
            P1 = '%s%s' % (P1, 1)
            P1 = self._add_point(P1, P2)
            P1 = self._convert_jacb_to_nor(P1)

        x = int(P1[0:self.para_len], 16)
        return r == ((e + x) % self.n)

    def _kg(self, k, Point):
```

```
        if (k % self.n) == 0:
            return '0' * 128
        Point = '%s%s' % (Point, '1')
        mask_str = '8'
        for i in range(self.para_len - 1):
            mask_str += '0'
        mask = int(mask_str, 16)
        Temp = Point
        flag = False
        for n in range(self.para_len * 4):
            if flag:
                Temp = self._double_point(Temp)
            if (k & mask) != 0:
                if flag:
                    Temp = self._add_point(Temp, Point)
                else:
                    flag = True
                    Temp = Point
            k = k << 1
        return self._convert_jacb_to_nor(Temp)

    def _double_point(self, Point):
        l = len(Point)
        len_2 = 2 * self.para_len
        if l < self.para_len * 2:
            return None
        else:
            x1 = int(Point[0:self.para_len], 16)
            y1 = int(Point[self.para_len:len_2], 16)
            if l == len_2:
                z1 = 1
            else:
                z1 = int(Point[len_2:], 16)

            T6 = (z1 * z1) % int(self.ecc_table['p'], base=16)
            T2 = (y1 * y1) % int(self.ecc_table['p'], base=16)
            T3 = (x1 + T6) % int(self.ecc_table['p'], base=16)
            T4 = (x1 - T6) % int(self.ecc_table['p'], base=16)
            T1 = (T3 * T4) % int(self.ecc_table['p'], base=16)
            T3 = (y1 * z1) % int(self.ecc_table['p'], base=16)
            T4 = (T2 * 8) % int(self.ecc_table['p'], base=16)
            T5 = (x1 * T4) % int(self.ecc_table['p'], base=16)
            T1 = (T1 * 3) % int(self.ecc_table['p'], base=16)
            T6 = (T6 * T6) % int(self.ecc_table['p'], base=16)
            T6 = (self.ecc_a3 * T6) % int(self.ecc_table['p'], base=16)
            T1 = (T1 + T6) % int(self.ecc_table['p'], base=16)
            z3 = (T3 + T3) % int(self.ecc_table['p'], base=16)
            T3 = (T1 * T1) % int(self.ecc_table['p'], base=16)
            T2 = (T2 * T4) % int(self.ecc_table['p'], base=16)
            x3 = (T3 - T5) % int(self.ecc_table['p'], base=16)

            if (T5 % 2) == 1:
                T4 = (T5 + ((T5 + int(self.ecc_table['p'], base=16)) >> 1) - T3) % int(self.ecc_table['p'], base
=16)
            else:
                T4 = (T5 + (T5 >> 1) - T3) % int(self.ecc_table['p'], base=16)

            T1 = (T1 * T4) % int(self.ecc_table['p'], base=16)
            y3 = (T1 - T2) % int(self.ecc_table['p'], base=16)
```

```
        form = '%%0%dx' % self.para_len
        form = form * 3
        return form % (x3, y3, z3)

    def _add_point(self, P1, P2):
        if P1 == '0' * 128:
            return '%s%s' % (P2, '1')
        if P2 == '0' * 128:
            return '%s%s' % (P1, '1')
        len_2 = 2 * self.para_len
        l1 = len(P1)
        l2 = len(P2)
        if (l1 < len_2) or (l2 < len_2):
            return None
        else:
            X1 = int(P1[0:self.para_len], 16)
            Y1 = int(P1[self.para_len:len_2], 16)
            if l1 == len_2:
                Z1 = 1
            else:
                Z1 = int(P1[len_2:], 16)
            x2 = int(P2[0:self.para_len], 16)
            y2 = int(P2[self.para_len:len_2], 16)

            T1 = (Z1 * Z1) % int(self.ecc_table['p'], base=16)
            T2 = (y2 * Z1) % int(self.ecc_table['p'], base=16)
            T3 = (x2 * T1) % int(self.ecc_table['p'], base=16)
            T1 = (T1 * T2) % int(self.ecc_table['p'], base=16)
            T2 = (T3 - X1) % int(self.ecc_table['p'], base=16)
            T3 = (T3 + X1) % int(self.ecc_table['p'], base=16)
            T4 = (T2 * T2) % int(self.ecc_table['p'], base=16)
            T1 = (T1 - Y1) % int(self.ecc_table['p'], base=16)
            Z3 = (Z1 * T2) % int(self.ecc_table['p'], base=16)
            T2 = (T2 * T4) % int(self.ecc_table['p'], base=16)
            T3 = (T3 * T4) % int(self.ecc_table['p'], base=16)
            T5 = (T1 * T1) % int(self.ecc_table['p'], base=16)
            T4 = (X1 * T4) % int(self.ecc_table['p'], base=16)
            X3 = (T5 - T3) % int(self.ecc_table['p'], base=16)
            T2 = (Y1 * T2) % int(self.ecc_table['p'], base=16)
            T3 = (T4 - X3) % int(self.ecc_table['p'], base=16)
            T1 = (T1 * T3) % int(self.ecc_table['p'], base=16)
            Y3 = (T1 - T2) % int(self.ecc_table['p'], base=16)

            form = '%%0%dx' % self.para_len
            form = form * 3
            return form % (X3, Y3, Z3)

    def _convert_jacb_to_nor(self, Point):
        len_2 = 2 * self.para_len
        x = int(Point[0:self.para_len], 16)
        y = int(Point[self.para_len:len_2], 16)
        z = int(Point[len_2:], 16)
        z_inv = pow(z, int(self.ecc_table['p'], base=16) - 2, int(self.ecc_table['p'], base=16))
        z_invSquar = (z_inv * z_inv) % int(self.ecc_table['p'], base=16)
        z_invQube = (z_invSquar * z_inv) % int(self.ecc_table['p'], base=16)
        x_new = (x * z_invSquar) % int(self.ecc_table['p'], base=16)
        y_new = (y * z_invQube) % int(self.ecc_table['p'], base=16)
        z_new = (z * z_inv) % int(self.ecc_table['p'], base=16)
        if z_new == 1:
```

```
            form = '%%0%dx' % self.para_len
            form = form * 2
            return form % (x_new, y_new)
        else:
            return None


if __name__ == '__main__':
    sk = func.random_hex(len(sm2p256v1_ecc_table['n']))
    tsm2 = TSM2(sk)
    print('pk:%s'   %tsm2.pk)
    print('pks:%064x'%tsm2.pks)
    for i in range(10):
        op = input('op: ').strip()
        if op == 'sign':
            sign(tsm2)
        elif op == 'verify':
            verify(tsm2)
        else:
            print("""sign: sign message
verify: verify message""")
```

暂时没做出来

混乱的音频

> 题目描述：
> 小明正在整理包含bed、bird、cat、dog、down、eight的音频文件数据，可小明不小心把分类好的一部分数据集弄混了。文件名非常混乱，无法判别出那些文件属于哪个单词类别。碰巧小红来问小明考试复习资料的密码。小明说：“我记得这个密码都是大写的英文字母，并且密码和我的爱好息息相关，具体的我也记不清了。你要是能帮我把这些音频文件分好类，你应该就会发现密码。”小红了解到小明是个计算机专业的学生，平时经常搜集一些与信息技术有关的新闻资讯。他经常参加ctf比赛，对密码学十分熟悉，对数字非常敏感。为了小红能得到考试复习资料，你能帮助小红将混乱的音频文件分好类并获取密码吗？
> 提示：比例 数据下载链接：https://pan.baidu.com/s/1r8C1FByHpgNZJsaUkjASOw 提取码：p1yh

小明的宠物兔

> 题目描述：
> 小明的宠物兔总是发出一些神秘的声音，小明很想知道兔兔在干什么，你能帮他翻译一下吗？

一张图片rabbit.txt，提示碰撞，应该是CRC32碰撞，foremost分离出压缩包

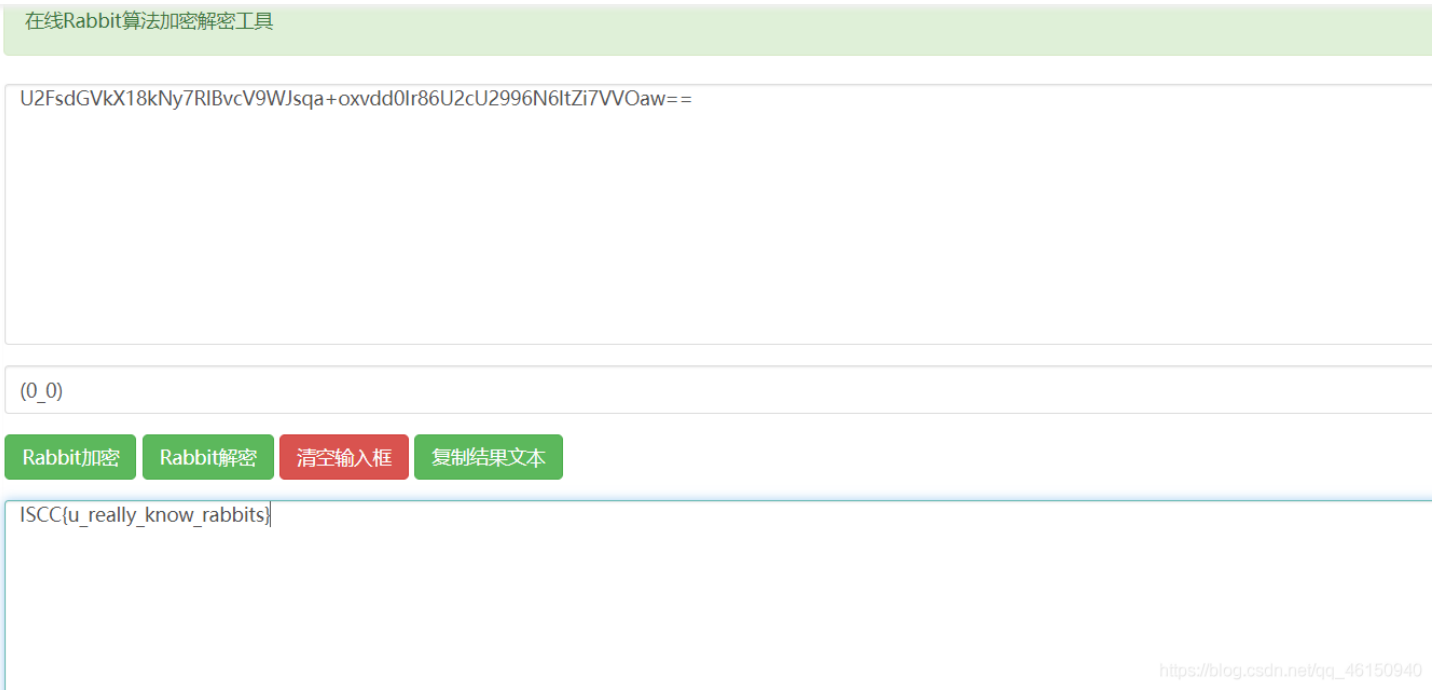
flag.txt内容，加salt，应该是rabbit加密

```
U2FsdGVkX18kNy7RlBvcV9WJsqa+oxvdd0Ir86U2cU2996N6ltZi7VVOaw==
```

CRC32碰撞，得到加密压缩包key.zip内容，

```
┌──(kali㉿kali)-[~/桌面/Python/CRC32]
└─$ python3 crc32.py reverse 0x3dacac6b
4 bytes: {0x47, 0x18, 0x87, 0xce}
verification checksum: 0x3dacac6b (OK)
5 bytes: (0_0) (OK)
5 bytes: DCr4m (OK)
6 bytes: 1QhloU (OK)
6 bytes: 3mmr6H (OK)
6 bytes: 49Gqqk (OK)
6 bytes: 5Uumn6 (OK)
6 bytes: 7ips7+ (OK)
6 bytes: 8Gpbyp (OK)
6 bytes: 9G1Sbi (OK)
6 bytes: EHZxWz (OK)
6 bytes: F93jxv (OK)
6 bytes: I6mk_a (OK)
6 bytes: J+wTt) (OK)
6 bytes: K+6eo0 (OK)
6 bytes: KGEHkt (OK)
6 bytes: N/jUuJ (OK)
6 bytes: O/+dnS (OK)
6 bytes: O3d8oG (OK)
6 bytes: TX.K94 (OK)
6 bytes: Uy1jKa (OK)
6 bytes: XJju5k (OK)
6 bytes: YJ+D.r (OK)
6 bytes: Zvoklv (OK)
6 bytes: a3H1hL (OK)
6 bytes: dG(pwf (OK)
6 bytes: e7U0i/ (OK)
6 bytes: fgb2/o (OK)
6 bytes: g6AbXj (OK)
6 bytes: kHvqPq (OK)
6 bytes: kT9-Qe (OK)
6 bytes: lQqOzZ (OK)
6 bytes: vwW0Z8 (OK)
```

5字节，最终密钥为`(0_0)`，Rabbit解密得到flag



擂台

小明的表情包

> 放假期间小红被亲戚叫去帮店里帮忙，店里忙极了导致小红没有时间写代码。小红苦恼极了，她突然想起来小明有一张非常适合描述她此时心情的表情包。于是，小红让小明把表情包分享给她。小明说如果你记得我的出生的日月年，我就交给你。小明的生日年份隐藏在这串凯撒密码“AVARGRRA
> AVARGL AVAR”中，你能帮小红得到小明的表情包吗？

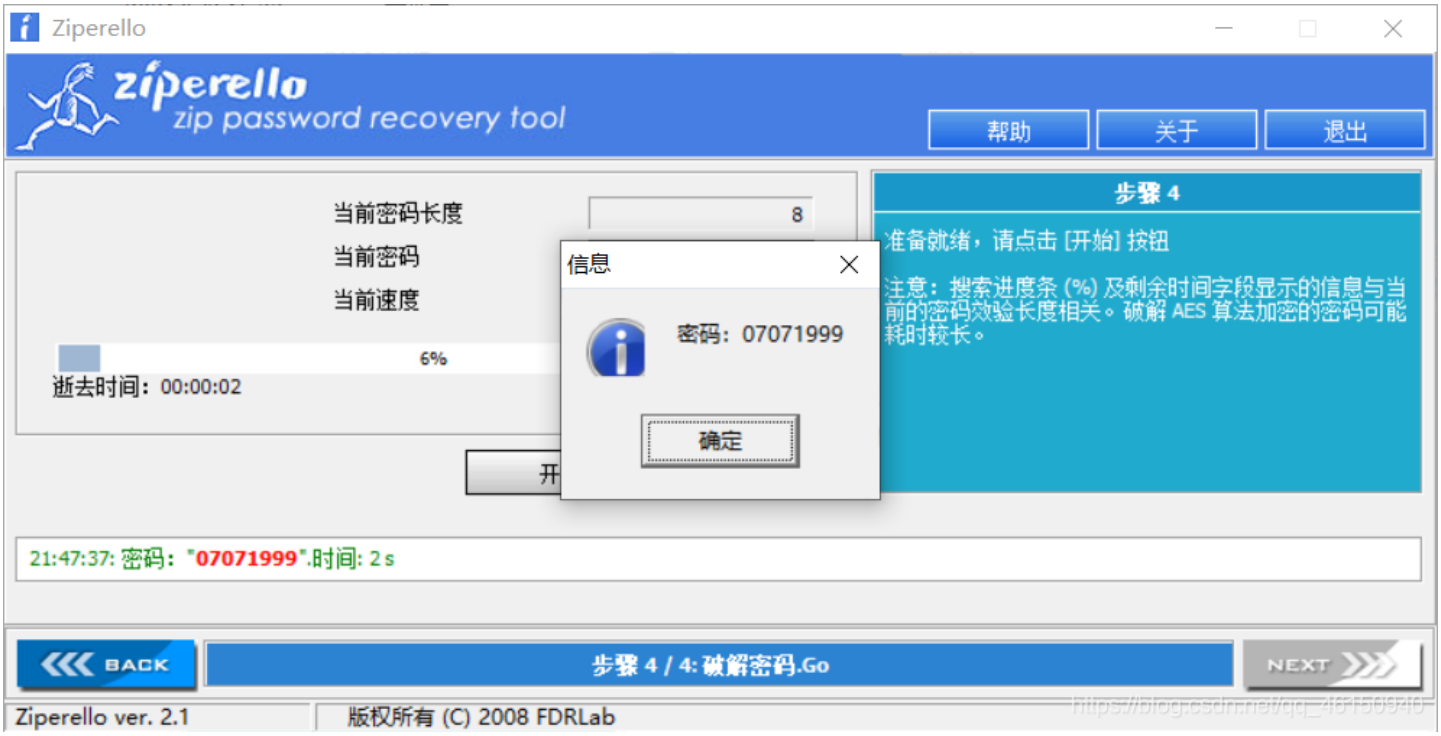
编写脚本进行凯撒密码爆破：

```python
s = "AVARGRRA AVARGL AVAR"

def kaisa(k):
    t = ""
    for c in s:
        if 'a' <= c <= 'z':
            t += chr(ord('a') + ((ord(c) - ord('a')) + int(k)) % 26)
        elif 'A' <= c <= 'Z':
            t += chr(ord('A') + ((ord(c) - ord('A')) + int(k)) % 26)
        else:
            t += c
    print(t)

for i in range(0,26):
    kaisa(i)
```

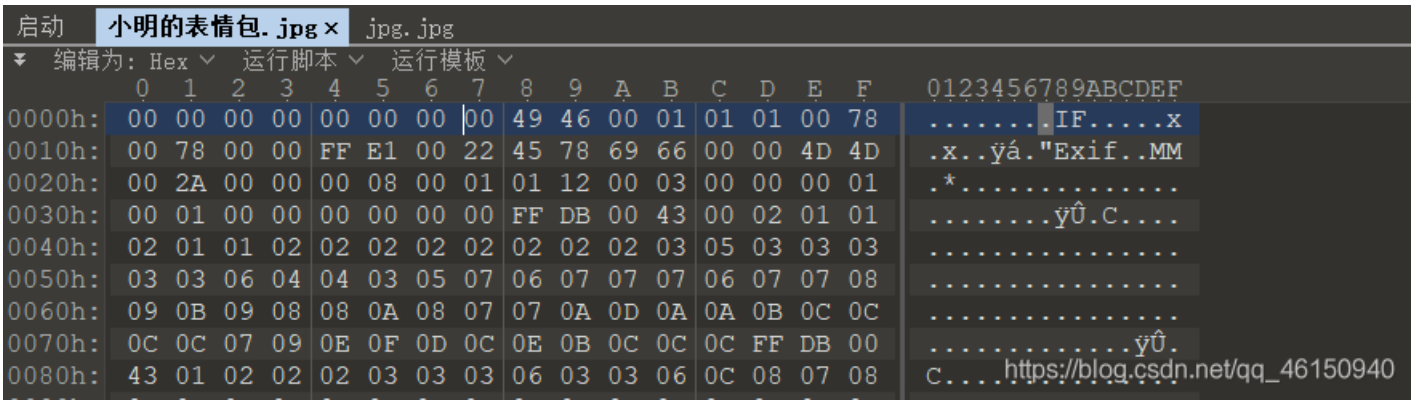
看到有含义的英文：NINETEEN NINETY NINE

```
HCHYNYYH HCHYNS HCHY
IDIZOZZI IDIZOT IDIZ
JEJAPAAJ JEJAPU JEJA
KFKBQBBK KFKBQV KFKB
LGLCRCCL LGLCRW LGLC
MHMDSDDM MHMDSX MHMD
NINETEEN NINETY NINE
OJOFUFFO OJOFUZ OJOF
PKPGVGGP PKPGVA PKPG
```

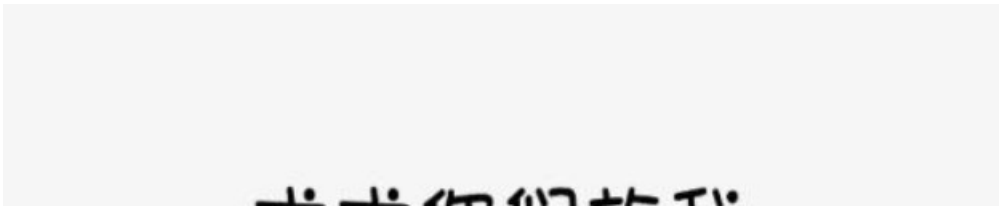

猜测是901909，然而并不是压缩包口令，根据题目提示密码是生日年份，直接爆破8位数字密码



得到口令07071999，解压压缩包，得到的jpg文件无法打开，拖入010 editor中，与正常的jpg文件比较，发现缺少JPG文件头



将前八位改为FF D8 FF E0 00 10 4A 46，保存得到flag



Base小偷

> 被凯撒小猫偷走的等号1/3
> Flag格式：flag{XXX}

密文

```
trefy2k2ov2lig2gqd2eqakoxjqcw4lztnfli
```

使用上面的凯撒密码爆破脚本，修改一下脚本，写入crack.txt文件中：

```
s = "trefy2k2ov2lig2gqd2eqakoxjqcw4lztnfli"
f = open("crack.txt", "w")
def kaisa(k):
    t = ""
    for c in s:
        if 'a' <= c <= 'z':
            t += chr(ord('a') + ((ord(c) - ord('a')) + int(k)) % 26)
        elif 'A' <= c <= 'Z':
            t += chr(ord('A') + ((ord(c) - ord('A')) + int(k)) % 26)
        else:
            t += c
    print(t)
    f.write(t+"\n")

for i in range(0,26):
    kaisa(i)
```

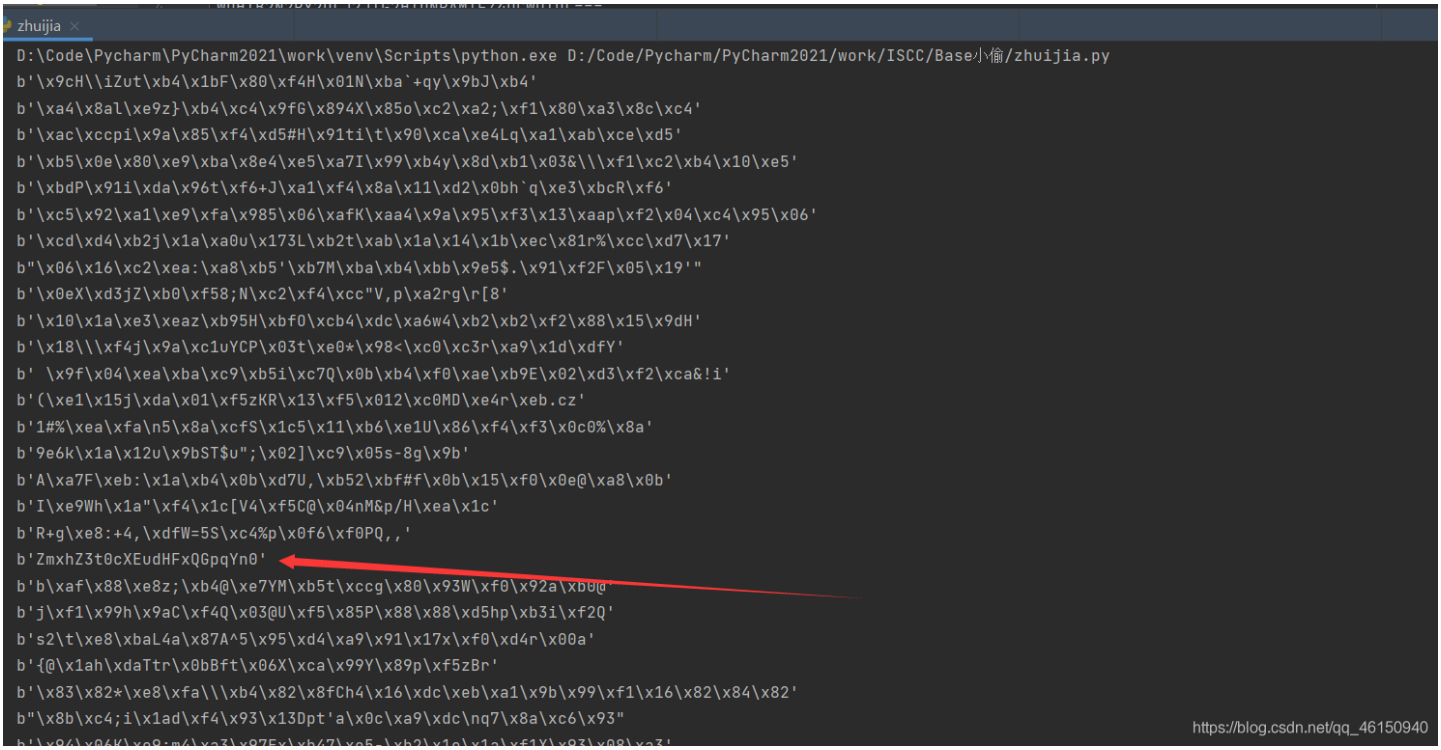
得到crack.txt

```
trefy2k2ov2lig2gqd2eqakoxjqcw4lztnfli
usfgz2l2pw2mjh2hre2frblpykrdx4mauogmj
vtgha2m2qx2nki2isf2gscmqzlsey4nbvphnk
wuhib2n2ry2olj2jtg2htdnramtfz4ocwqiol
xvijc2o2sz2pmk2kuh2iueosbnuga4pdxrjpm
ywjkd2p2ta2qnl2lvi2jvfptcovhb4qeyskqn
zxkle2q2ub2rom2mwj2kwgqudpwic4rfztlro
aylmf2r2vc2spn2nxk2lxhrveqxjd4sgaumsp
bzmng2s2wd2tqo2oyl2myiswfryke4thbvntq
canoh2t2xe2urp2pzm2nzjtxgszlf4uicwour
dbopi2u2yf2vsq2qan2oakuyhtamg4vjdxpvs
ecpqj2v2zg2wtr2rbo2pblvziubnh4wkeyqwt
fdqrk2w2ah2xus2scp2qcmwajvcoi4xlfzrxu
gersl2x2bi2yvt2tdq2rdnxbkwdpj4ymgasyv
hfstm2y2cj2zwu2uer2seoyclxeqk4znhbtzw
igtun2z2dk2axv2vfs2tfpzdmyfrl4aoicuax
jhuvo2a2el2byw2wgt2ugqaenzgsm4bpjdvby
kivwp2b2fm2czx2xhu2vhrbfoahtn4cqkewcz
ljwxq2c2gn2day2yiv2wiscgpbiuo4drlfxda
mkxyr2d2ho2ebz2zjw2xjtdhqcjvp4esmgyeb
nlyzs2e2ip2fca2akx2ykueirdkwq4ftnhzfc
omzat2f2jq2gdb2bly2zlvfjselxr4guoiagd
pnabu2g2kr2hec2cmz2amwgktfmys4hvpjbhe
qobcv2h2ls2ifd2dna2bnxhlugnzt4iwqkcif
rpcdw2i2mt2jge2eob2coyimvhoau4jxrldjg
sqdex2j2nu2khf2fpc2dpzjnwipbv4kysmekh
```

猜测是Base32编码，得到密文每行37字符，而Base32按5比特切分的二进制数据必须是40比特的倍数，需要在每行末尾追三个===，并且小写转换为大写

```python
# coding: UTF-8
import base64

ff = open('output.txt','w')
with open('crack.txt','r') as f:
    lines = f.readlines()
    for line in lines:
        line_n =line.replace('\n','')
        line_n=line_n+r'==='+'\n'      #行末尾加上"===",同时加上"\n"换行符
        line_new=line_n.upper()         #小写转换为大写
        ff.write(line_new)            #写入一个新文件中
ff.close()

with open('output.txt','r') as f:

    list = f.read().splitlines()   #存入列表中
    i = 0
    ls2 = [str(i) for i in list]  #转换为字符串类型
    while i < len(list):
        print(base64.b32decode(ls2[i]))
        i += 1
```

运行脚本，发现一段base64编码



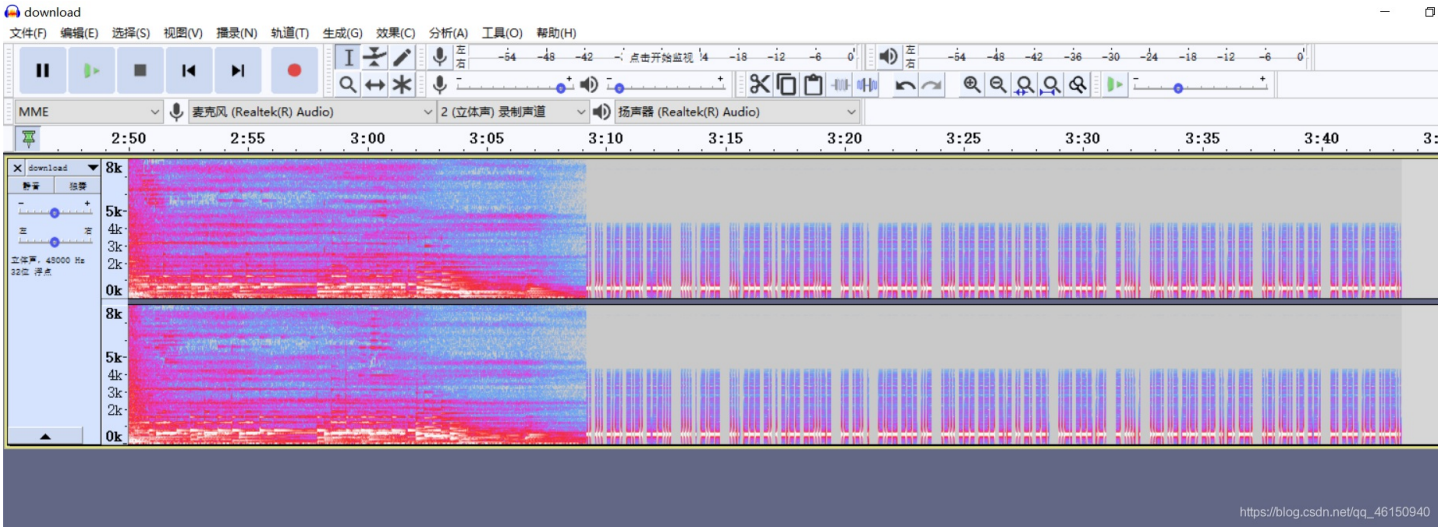
base64解码得到flag

```
flag{tqq.tqq@jjb}
```

真作假时假亦真

> 浅韵姐姐第一次参加ISCC，但没有很多CTF工具，为此她很苦恼。
> 你能切身体会一下吗？
> 附件下载链接：https://pan.baidu.com/s/1jroyNjtnCwPXxt6A8_0pAw
> 提取码：a0oh

下载附件，得到download.wav，拖入Audacity查看频谱图



弄下来

```
.. ... -.-. -.-. ------ .... . -.-. .... --- -. . -.-- --..-- .-.-.. -.-- --- ..- .-.. --- --- -.- .. -. --. ..-
. --- .-. - .... . ..-. ..- -.-. -.- .. -. --. ...- .-.. .- --. -. --- .-- ..---
```

摩斯密码解密

```
ISCCHECHONEYYOULOOKINGFORTHEFUCKINGFLAGNOW?
```

不是flag，foremost分离文件得到一张png图片，exiftool查看图片信息，得到QQ号

```
┌──(kali㉿kali)-[~/桌面/output/png]
└─$ exiftool 00083771.png
ExifTool Version Number         : 12.16
File Name                       : 00083771.png
Directory                       : .
File Size                       : 288 KiB
File Modification Date/Time     : 2021:05:08 08:46:57-04:00
File Access Date/Time           : 2021:05:08 08:47:00-04:00
File Inode Change Date/Time     : 2021:05:08 08:46:57-04:00
File Permissions                : rw-r--r--
File Type                       : PNG
File Type Extension             : png
MIME Type                       : image/png
Image Width                     : 1242
Image Height                    : 1704
Bit Depth                       : 8
Color Type                      : RGB with Alpha
Compression                     : Deflate/Inflate
Filter                          : Adaptive
Interlace                       : Noninterlaced
Pixels Per Unit X               : 2874
Pixels Per Unit Y               : 2874
Pixel Units                     : meters
XMP Toolkit                     : Adobe XMP Core 6.0-c002 79.164488, 2020/07/10-22:06:53
Creator Tool                    : Adobe Photoshop 22.0 (Windows)
Create Date                     : 2021:05:03 00:55:27+08:00
Modify Date                     : 2021:05:03 00:59:09+08:00
Metadata Date                   : 2021:05:03 00:59:09+08:00
Format                          : image/png
Color Mode                      : RGB
ICC Profile Name                : sRGB IEC61966-2.1
Instance ID                     : xmp.iid:08dbb1cf-06a6-6147-a19e-3c30c111887d
Document ID                     : adobe:docid:photoshop:0cf82e22-bc60-394b-9808-1ec5719e29bc
Original Document ID            : xmp.did:94a58657-d199-9a45-bb13-6a6d55604521
Text Layer Name                 : ボ④wang猪榮燿排位卡  Q (-) 5.2肆 5.5.23) 2ず
Text Layer Text                 : ボ④wang猪榮燿排位卡  Q (-) 5.2肆 5.5.23) 2ず
History Action                  : created, saved
History Instance ID             : xmp.iid:94a58657-d199-9a45-bb13-6a6d55604521, xmp.iid:08dbb1cf-06a6-6147-a19e-3c30c111887d
History When                    : 2021:05:03 00:55:27+08:00, 2021:05:03 00:59:09+08:00
```


查看QQ空间，在留言板发现假flag

来社我啊 第1楼

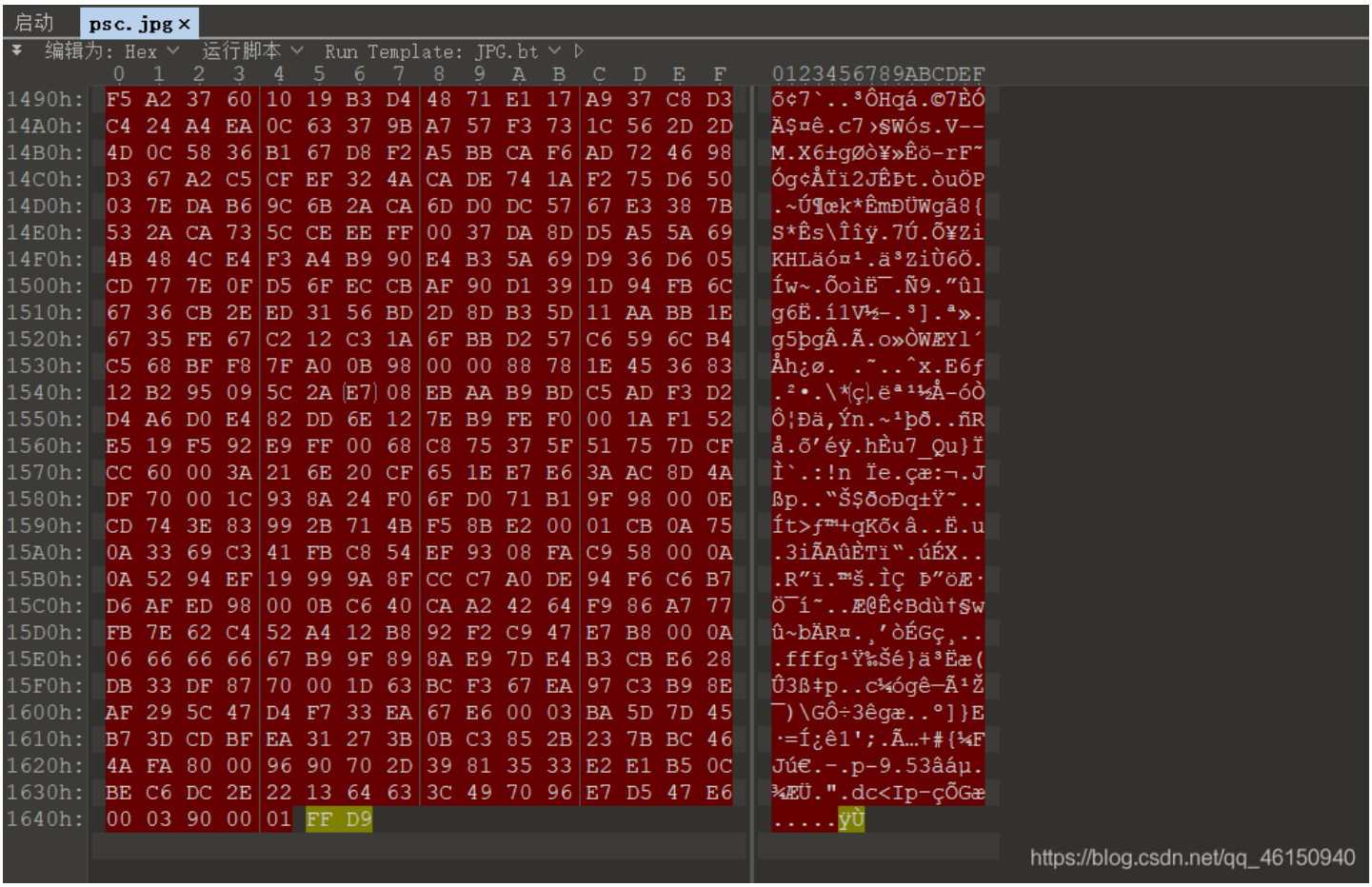
ISCC{Qianyun_is_Lovely}

2021-05-03 03:53

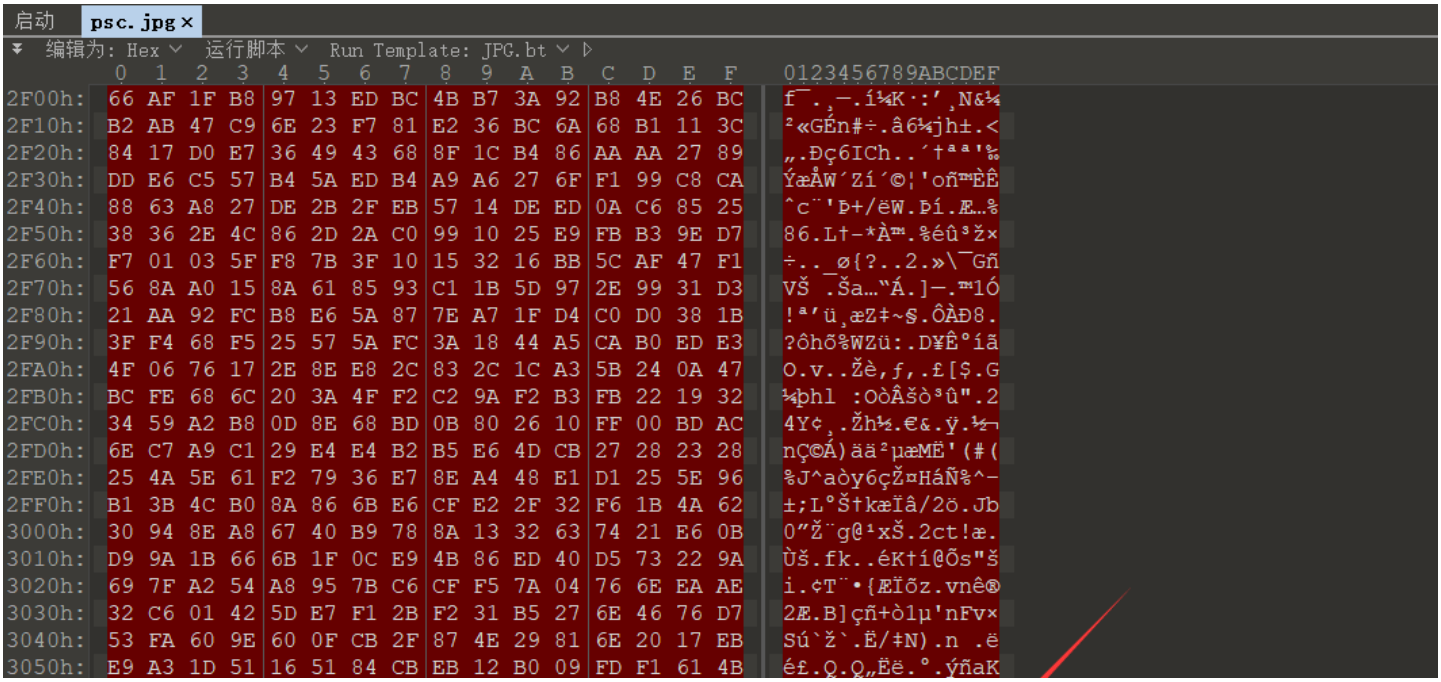
加好友问题`ISCC{e@$Y_s0cia1_engineering}`也是假flag

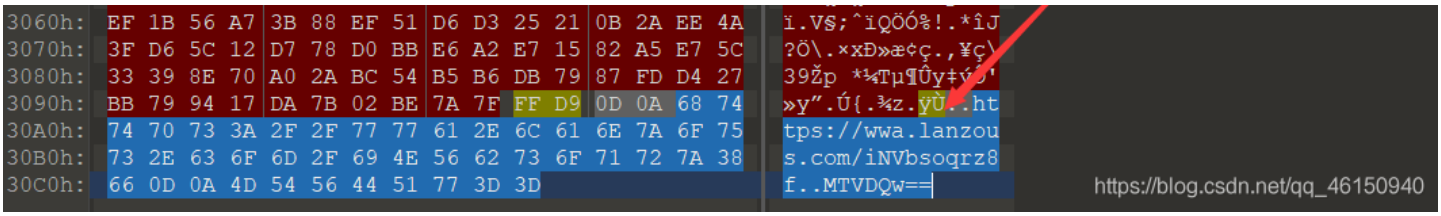
空间里唯一的一张图片

我当时做的时候掉坑里去了，直接下载图片，010 editor什么也看不到，思路就断了。



用原图地址链接下载图片，能看到一个蓝奏云链接

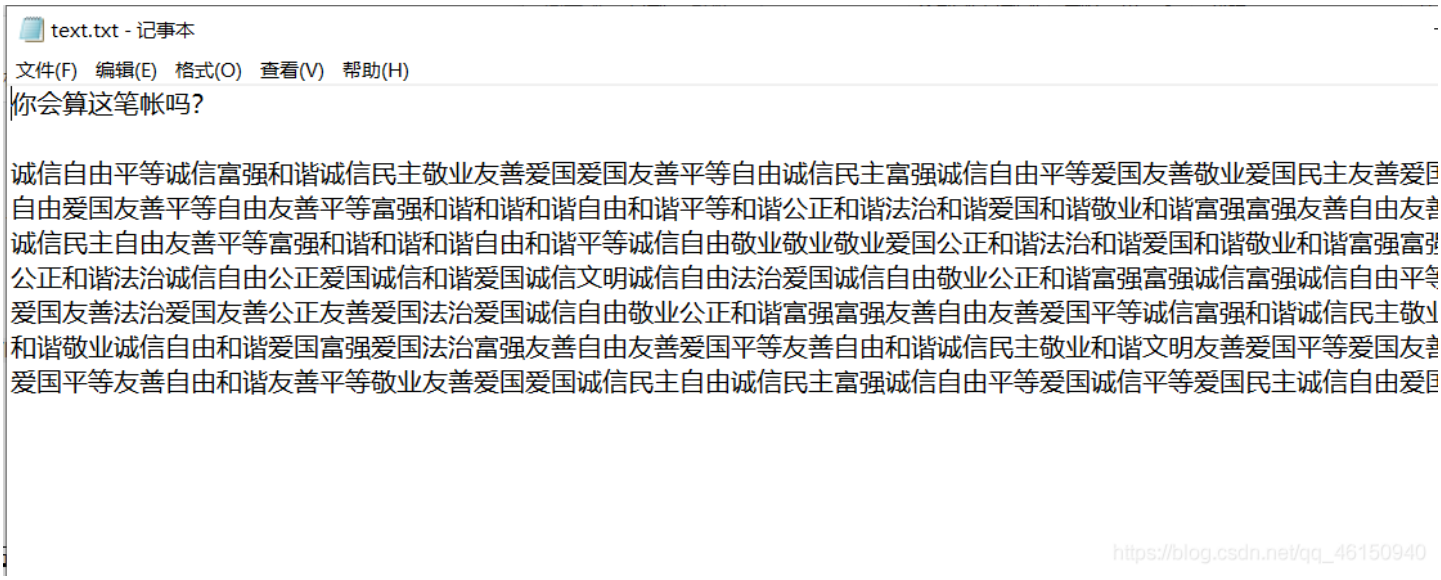




提取出来

```
https://wwa.lanzous.com/iNVbsoqrz8f
MTVDQw==
```

提取码是`MTVDQw==`， Base64解码得到`15CC`,下载附件



使用在线网站进行社会主义核心价值观解密得到

```
壹贰叁肆伍陆柒8玖〇壹2叁肆56柒89〇壹23肆伍
陆78901贰3肆56柒89〇1贰叁4伍陆7捌玖0
壹2叁肆伍6柒89〇壹贰叁肆5陆柒捌9〇123肆5
陆7捌玖〇1贰345陆柒8玖〇壹贰34567890
壹2叁肆伍6柒8玖01贰叁4伍6柒捌玖〇壹234伍
陆78901贰3肆5陆78玖0壹贰345陆7捌玖0
壹贰叁肆伍陆柒890壹贰叁肆56柒8玖0壹234伍
67890123肆伍6柒捌玖〇1贰345陆7890
壹贰3肆56柒捌90123肆56柒捌玖〇壹贰3肆5
67捌9〇1234伍陆78玖〇壹贰3肆5陆7捌9〇
12叁肆伍6柒8901贰叁肆伍陆7捌9〇壹2叁肆伍
陆柒89〇123肆伍67890壹234567捌玖0
壹2叁4伍陆柒捌901贰3肆56柒捌9〇壹2345
陆78玖0壹23肆伍67捌90壹2叁4伍陆7捌9〇
壹2叁45陆柒89〇1贰叁45陆7890壹23肆伍
陆柒8901234伍陆柒8玖〇1贰34伍6柒捌玖0
壹贰34伍陆柒89〇壹2叁肆5陆柒89〇1贰叁肆伍
678901234伍67捌9〇1234567890
壹贰叁肆伍陆柒8玖0壹2叁4伍6柒8玖〇壹贰叁肆伍
陆78901贰3肆伍陆7捌901贰3肆56789〇
壹2叁肆伍6柒89012345陆柒8玖0壹贰叁4伍
陆7捌玖〇1贰3肆56789〇1贰3肆5陆柒捌9〇
壹2叁肆伍6柒8玖〇1贰345陆柒8玖0壹贰叁4伍
陆78901贰3肆5678玖01贰3肆56789〇
壹贰叁肆伍陆柒8玖01贰3肆伍陆78玖〇壹贰叁肆伍
```

汉字转换成1，数字换成0 。

```
11111110111011001001100111
10000010100100101101101110
10111010011111011101000010
10111010001101111000000000
10111010100110101111100001
10000010101001011000101100
11111110001111001010100001
00000000110111101000100000
11010011000001001111111010
00101000011001111010101011
00111010000111111010110111
11001000110000010000000110
10101111000101001101100000
10010100110010010101101011
10100110010110010000100111
11000000011101101001011110
11001110011011011001011111
00000000001001010000000000
11111110101010101011111111
10000010111010001010000001
10111010000000001101011101
10111010100000101010111011
10111010110100011010111011
10000010100001001010000001
11111110100101110011111111
```

25x25,之前做过一样的，直接使用脚本转换成二维码

```
import PIL.Image
MAX = 25
pic = PIL.Image.new("RGB",(MAX,MAX))
str = '1111111011101100100110011100000101001001011011011010111010011111011101000101011101000110111100000000101110101001101011111000110000010101001011000101101111111000111100101010001000000001101111010001000011010011000001001111110100010100001100111101010101001110100001111101011011111100100011000001000000110101011110001010011011000010010100110010010101101011010011001011001000010011110000000111011010010111011001110011011011001011110000000000100101000000000111111101010101010111111111000001011101000101000001101110100000000110101110110111010100000101010111011010000011010011101100001010000100101000001111111101001011100111111'
i=0
for y in range (0,MAX):
        for x in range (0,MAX):
                if(str[i] == '1'):
                        pic.putpixel([x,y],(0,0,0))
                else:
                        pic.putpixel([x,y],(255,255,255))
                i = i+1
pic.show()
pic.save("flag.png")
```

扫描二维码得到flag