一枚亲斤手对中大SYSUMSClub的puzzle的解题记录(writeup)(2021-10)(G2T1me)

原创

DOG-DUCK 于 2021-10-18 21:16:32 发布 93 收藏 1
文章标签： 算法 python

本文链接：https://blog.csdn.net/weixin_44684503/article/details/120832848

0x00 写在前面

> MSC Puzzle 是由中山大学 MSClub 与中山大学 W4terDr0p 战队联合举办的趣味性解谜游戏，内容除古典密码、数字谜题等经典谜题外，还包含部分需要基础计算机知识、基础CTF知识以及部分基础算法知识解决的谜题。

解谜地址https://puzzle.sysums.club/https://puzzle.sysums.club/

由于作者没有系统受过计算机教育，所学内容来源为互联网/不懂就问，如有不妥之处，还请海涵。

0x01 解题记录

01 你好！勇士

首页复制msc{}及其包裹的内容即得。

02 老古董

摩斯密码

03 躲在墙后 / 04 隐身药水

按F12/右键检查均可

05 完形填空 / 07 世间万物的答案

网上可以查到，此处从略

06 去问导航

□.□.sysums.club

复制到浏览器地址栏中，得xn--9q8h.xn--dr8h.sysums.club

右键检查发现txt提示，使用dig命令，windows系统使用nslookup -q=TXT xn--9q8h.xn--dr8h.sysums.club亦可。

```
dogduck@DESKTOP-U92OFPT:~> dig xn--9q8h.xn--dr8h.sysums.club txt +noidnin +noidnout

; <<>> DiG 9.16.6 <<>> xn--9q8h.xn--dr8h.sysums.club txt +noidnin +noidnout
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55970
;; flags: qr rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;xn--9q8h.xn--dr8h.sysums.club.  IN      TXT

;; ANSWER SECTION:
xn--9q8h.xn--dr8h.sysums.club.  0 IN     TXT     "msc{[redacted]}"

;; Query time: 170 msec
;; SERVER: 172.21.128.1#53(172.21.128.1)
;; WHEN: Mon Oct 18 19:54:46 CST 2021
;; MSG SIZE  rcvd: 110

dogduck@DESKTOP-U92OFPT:~>
```


08 我的咖啡

这题与hackergame的一道题有关：hackergame2018-writeups/official/who_am_i at master · ustclug/hackergame2018-writeups · GitHubWrite-ups for hackergame 2018. Contribute to ustclug/hackergame2018-writeups development by creating an account on GitHub.
https://github.com/ustclug/hackergame2018-writeups/tree/master/official/who_am_i

但他没有在服务器做手脚，只是阅读理解。

https://www.rfc-editor.org/rfc/rfc2324.txt https://www.rfc-editor.org/rfc/rfc2324.txt

使用curl,postman,fiddler都无果，查阅且经提示得知，这就是答案.....

09 设计师

base64 + 大小写反转

10 监听电话

https://mscpuzzle.oss-cn-guangzhou.aliyuncs.com/2021/phone.mp3

没有更新完

11 一张图片

图片隐写，这里使用matlab，参考资料：【matlab】一篇文章看懂图像隐写的简易实操_RouTine-CSDN博客_jsteg隐写matlab代码

```
>> img = imread('msc.png');
>> imshow(logical(bitget(img(:,:,1),1)))
```

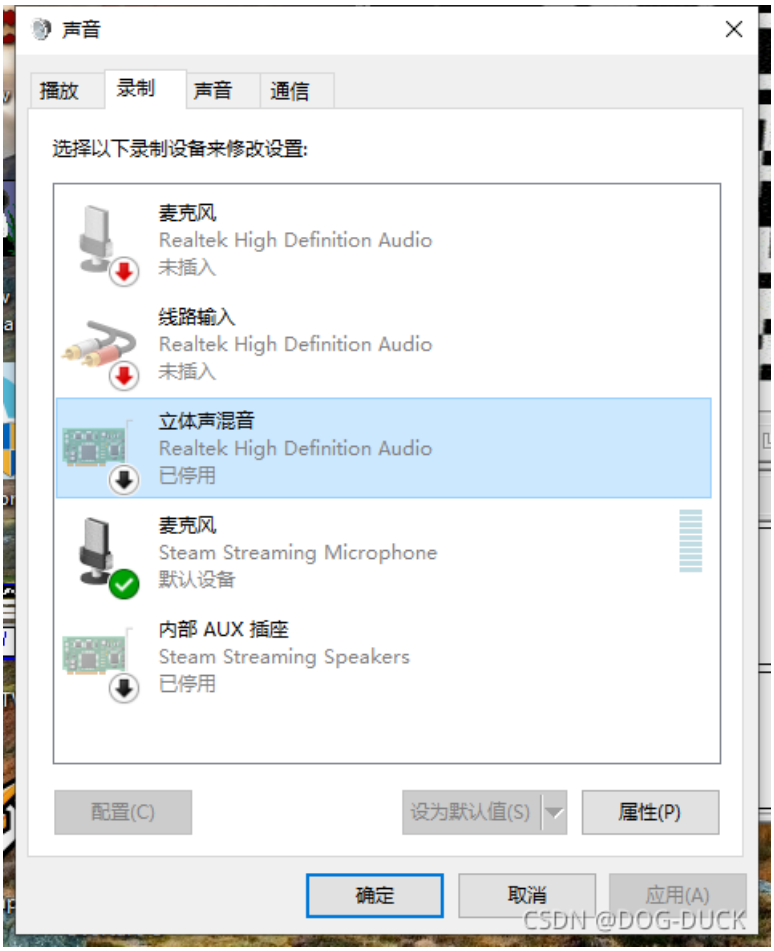
12 数字序列

F12/右键检查发现提示OEIS，到oeis中查即可

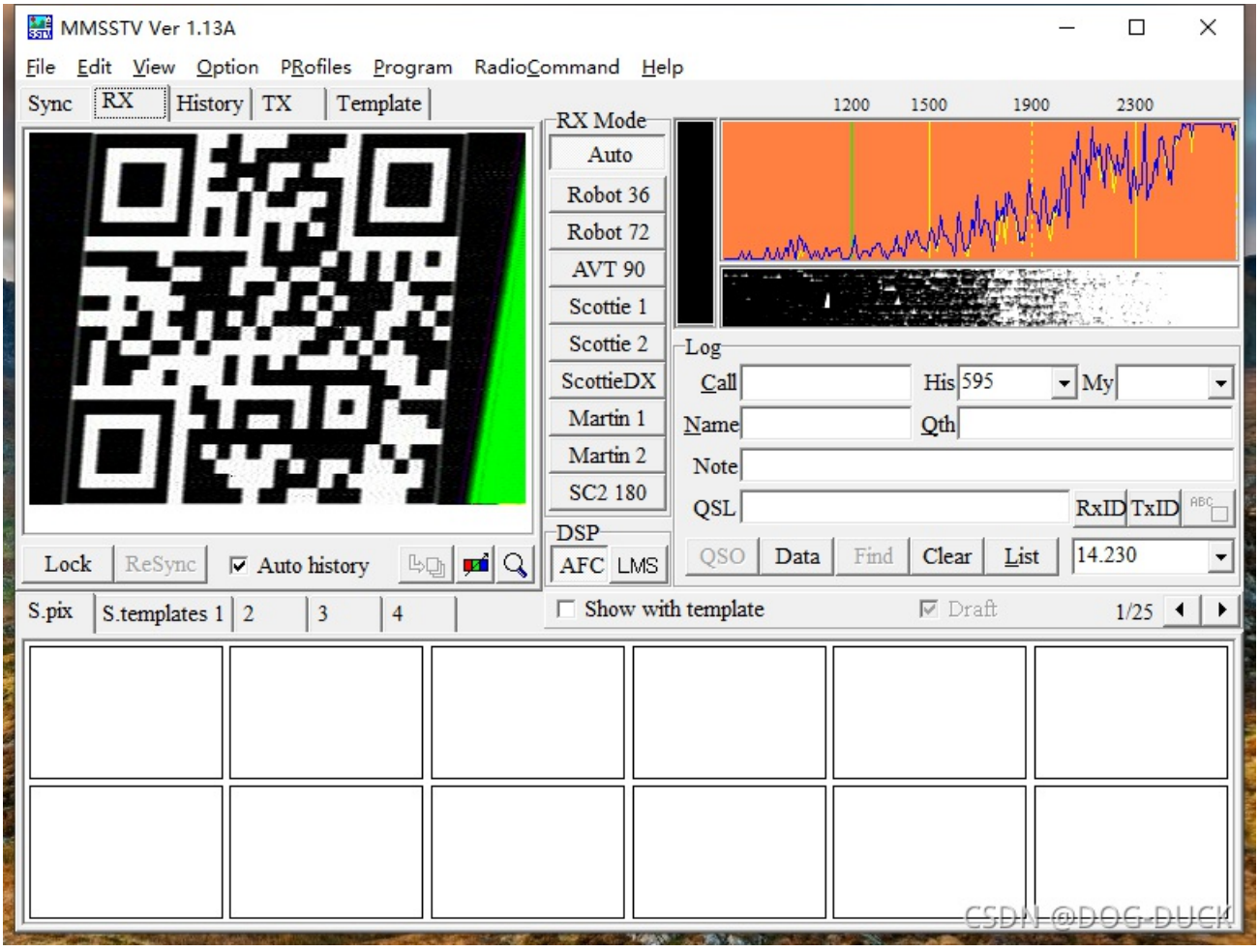
13 未知信号

https://mscpuzzle.oss-cn-guangzhou.aliyuncs.com/2021/signal.mp3

我用的是MMSSTV，先启用立体声混音，使用内录。



一边播放音频就可以得出二维码



14 又一张图片

```
dogduck@DESKTOP-U92OFPT:~> python -m binwalk /mnt/c/Users/Inscription/Downloads/msc2.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 1024 x 1024, 8-bit/color RGB, non-interlaced
41            0x29            Zlib compressed data, default compression
49722         0xC23A          Zip archive data, encrypted at least v2.0 to extract, compressed size: 6785, uncompressed
size: 17419, name: image.svg
56659         0xDD53          End of Zip archive, footer length: 22

dogduck@DESKTOP-U92OFPT:~>
```


binwalk一下，发现压缩包，改文件扩展名为zip，用图片隐写得到的密码解压得image.svg

用记事本编辑，把<rect之后字符全删掉，</svg>不要动，得到答案

15 盲人数学家

1. □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
2. □□

参考资料：視覺障礙輔助科技筆記本

16 最小的语言

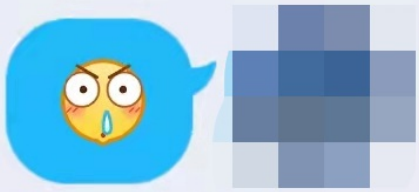
brainf***(BF)解码得=WYxy^WXArn(qs|p!-k<~<X4i;A[!n97zu4Yru{@=={qkw+78tx~
[94:*By66&@U<5.&;66&:047V@D[,r:60}55^{yX<3@u~.sv/m4(]i)1pV!op[(Z:54f/ssgt(!XZ|xq|ouuv+@|!=u

得到提示：

oh这样

你可以写个程序穷举一下位移（

下午6:58



然后把结果限制在 ASCII range [33, 126]

```
str = '=WYxy^WXArn(qs|p!-k<~<X4i;A[!n97zu4Yru{@=={qkw+78tx~[94:*By66&@U<5.&;66&:047V@D[,r:60}55^{yX<3@u~.sv

for i in range(94):
    print('#'*70, i, '次偏移')
    tmpstr = ''
    for j in range(len(str)):
        if ord(str[j:j+1])-i >= 33:
            tmpstr = tmpstr + chr(ord(str[j:j+1])-i)
        else:
            tmpstr = tmpstr + chr(ord(str[j:j+1])-i+94)
    try:
        print(tmpstr.index('msc'))
        print(tmpstr)
        print('偏移成功！！！')
        break
    except:
        pass
```

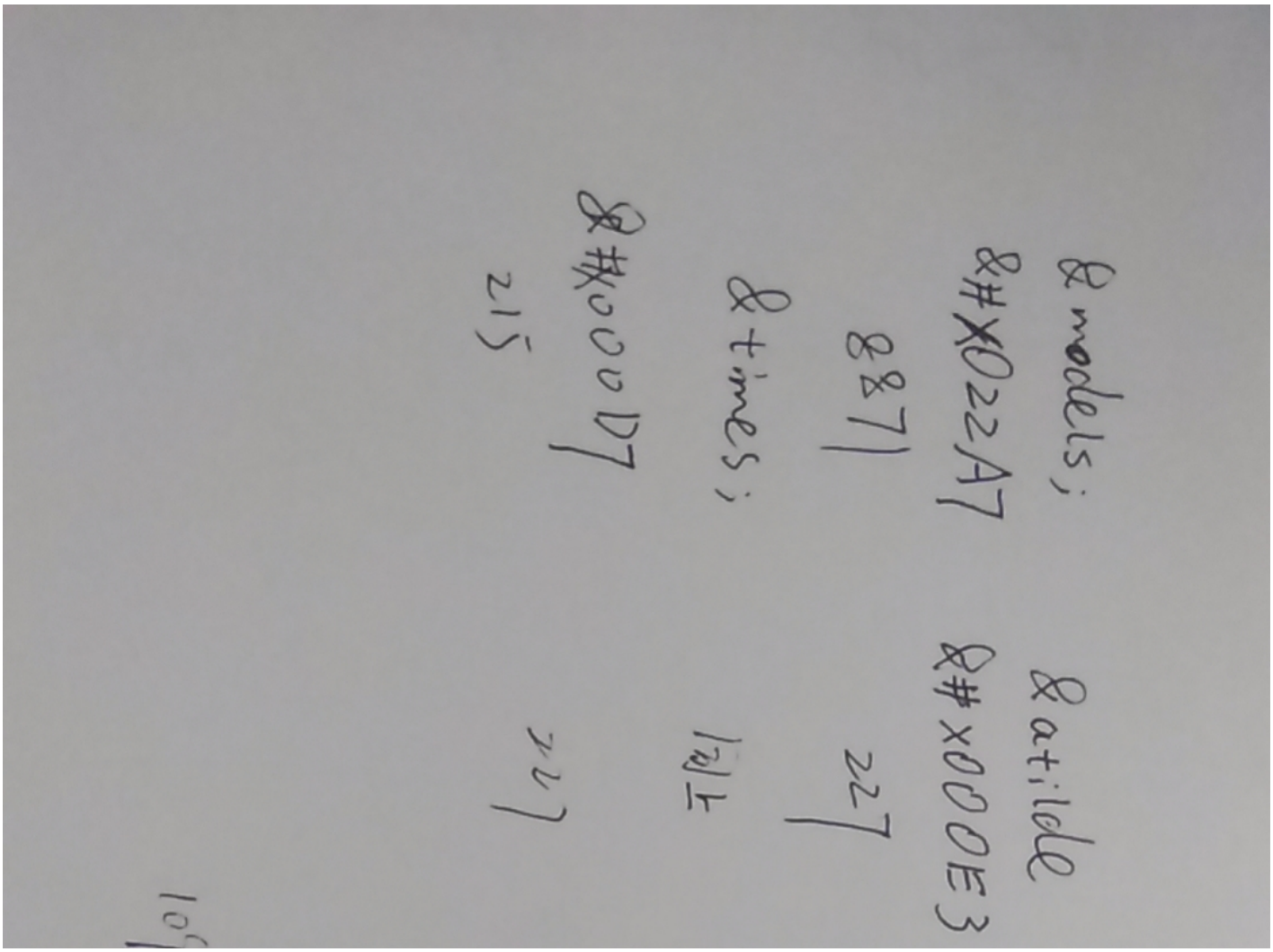
17 一个传说

参考资料：Project NANO: Official Writeup – 翰林的小站

18 时间商人

从F12中得到提示：charref

参考资料：Character Entity Reference Chart



找到首字母，答案为msc{ma*****cK}

19 追踪目标

Google + 三词地址

20 紧急唤醒

先找出私钥，再根据私钥解密

```
p = 0xbb1a21ab46e16672a6dfe537c5d03121252685a1a72fab827ed14d61caa80f68b9bda9fb0d9651719ab099d05fd0da03443a5
q = 0x85eed89c104292b715a45ec8a1c3328506b429c2b7477c9ab094313fcf0020ba7352b102608cab79bde92978aedb052a546fb

n = p * q
phi = (p-1) * (q-1)
e = 0x10001

for d in range(n):
    if (d * phi + 1) % e == 0:
        res = (d * phi + 1) // e
        print(res)
        print('%#x'%res)
        break
```

```
from Crypto.Util.number import*

p = 0xbb1a21ab46e16672a6dfe537c5d03121252685a1a72fab827ed14d61caa80f68b9bda9fb0d9651719ab099d05fd0da03443a5
q = 0x85eed89c104292b715a45ec8a1c3328506b429c2b7477c9ab094313fcf0020ba7352b102608cab79bde92978aedb052a546fb

n = p * q
phi = (p-1) * (q-1)
e = 0x10001
d = 0x321fd4c16859acf0e8a7b3ddfda8e0acc875c8ab8ec6aa588bcdb3879e4c5093de4c95332cfb8739713b841f148830be5de3c

answer = b'something you dont know'
m = bytes_to_long(answer)
print(m)
print('+'*120)
c = pow(m,e,n)
print(c)
print('+'*120)
ori = pow(c,d,n)
print(ori)
print(long_to_bytes(ori).decode('utf-8','ignore'))
c = 0x17c7af1ec9c020eb9d8f26049f002b58f93591a817ebff4c00e9e46254261db54a2c2d086dd0f532994329faf2133b1c70029
ori = pow(c,d,n)
print(ori)
print(long_to_bytes(ori).decode('utf-8','ignore'))
```

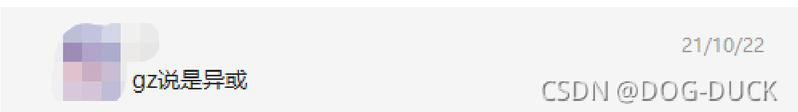
21 Emojis

```
□□□□□□□□□□□□□□□□□□□□□☃□□□□□□□□□□□□□□□□□□□□□✖□□□□□□□□✉□□□□□□□□□
```

密钥就是EMOJI，参考：emoji-aes

结果得到五色环电阻的五种颜色，求出阻值即可。

22 简易加密

果断放弃 

23 图像处理大师

文件下载：https://mscpuzzle.oss-cn-guangzhou.aliyuncs.com/2021/cv.zip

> 小纸条：0.8560967955058971 17.682512473330895

```
import cv2
import numpy as np

def ifft(m, p):
    absfft = np.exp(m)
    fft = absfft * ( np.cos(p) + np.sin(p)*1j )
    fft = np.fft.ifftshift(fft)
    img = np.fft.ifft2(fft)
    return img.astype(np.uint8)

def demapping(new_data, data_min, data_max):
    new_data = new_data.astype(np.float64)
    interval = data_max - data_min
    data = interval * new_data / 255 + data_min
    return data

if __name__ == '__main__':
    img1 = cv2.imread(f'cv1.png', cv2.IMREAD_GRAYSCALE)
    img2 = cv2.imread(f'cv2.png', cv2.IMREAD_GRAYSCALE)

    m = demapping(img1, 0.8560967955058971, 17.682512473330895)
    p = demapping(img2, -np.pi, np.pi)

    # print(m, p)
    img = ifft(m, p)

    cv2.imwrite(f'answer.png', img)
```

由于有复数存在，$fft = |fft|\,(\cos p + i \sin p)$

24 混乱的二维码

参考资料：祥云杯2021 Writeup | GZTime

```
str = [
  "10111010111000011000001011101", "00111010110010101000000000010",
  "11100001101100100001011010110", "10000010000111101111111111101",
  "10111010000010001000001011101", "01110011010111001001011100001",
  "10101101110101001110000110010", "10111010111010111110001011101",
  "10010100110101101111100111110", "10111110011101011000011010111",
  "11001010001011011000000001000", "10111010100101100111111110100",
  "10000010110010110100100011001", "11111110101010101010101111111",
  "10111010001100010100110110111", "11111110101101010101110110000",
  "00000000100100100101100011100", "00000000111001101011011111001",
  "11111100100110010010001011110", "00010001110001001101110001010",
  "10000010111110001101101000001", "11001110110110010101100100001",
  "11011101111010111101111110100", "11111110100010100101101011100",
  "00000000011111011001100000000", "10111010111001000110010100010",
  "11111110101110011001001111111", "10000010001100011010101000001",
  "10011111111101000101110010111"
]

with open('res.csv','w') as f:
    for i in str:
        for j in range(len(i)):
            if i[j:j+1] == '0':
                f.write(' ,')
            else:
                f.write('X,')
        f.write('\n')
```

把数据导到excel，分析

X		X	X	X		X		X	X	X					X	X						X		X	X	X		X		3、5	1
		X	X	X		X		X	X			X		X		X											X			11、13、15、17、19	2
X	X	X					X	X		X	X			X					X			X		X		X	X			10、12、14、16、18、20	3
X						X					X	X	X	X		X	X	X	X	X	X	X	X		X			28	4
X		X	X	X		X						X				X						X		X	X	X		X		4	5
	X	X	X			X	X		X		X	X	X			X			X			X		X	X	X				11、13、15、17、19	6
X		X		X	X		X	X	X		X		X			X	X	X							X	X				10、12、14、16、18、20	7
X		X	X	X		X		X	X	X		X		X	X	X	X	X			X		X	X	X		X			3、5	8
X			X		X			X	X		X		X	X	X	X	X	X	X			X	X	X	X	X				10、12、14、16、18、20	9

X		X	X	X	X	X			X	X	X		X		X	X					X	X		X		X	X	X		11、13、15、17、19	10
X	X			X		X				X		X	X		X	X									X					11、13、15、17、19	11
X		X	X	X		X		X			X		X	X			X	X	X	X	X	X		X						25	12
X						X		X	X			X		X	X		X					X	X			X				24	13
X	X	X	X	X	X	X		X		X		X		X		X			X		X	X	X	X	X	X	X	X		7	14
X		X	X	X		X				X	X				X		X			X	X		X	X		X	X	X		27	15
X	X	X	X	X	X	X		X		X	X		X		X		X			X	X									29	16
								X			X			X			X					X	X							22	17
								X	X	X			X	X		X		X	X		X	X	X			X				10、12、14、16、18、20	18
X	X	X	X	X	X			X			X	X			X			X			X		X	X	X	X				10、12、14、16、18、20	19
			X				X	X	X				X			X		X	X					X		X				10、12、14、16、18、20	20
X						X		X	X	X	X	X				X	X		X								X			2	21
X	X			X	X	X		X	X		X	X			X		X		X		X							X		11、13、15、17、19	22
X	X		X	X	X	X		X	X	X	X		X	X		X		X	X	X	X	X		X						21	23
X	X	X	X	X	X	X		X				X		X			X		X		X		X	X	X					23	24
									X	X	X	X	X		X	X			X	X										8	25
X		X	X	X		X		X	X	X			X				X	X			X		X				X			26	26
X	X	X	X	X	X	X		X		X	X	X			X	X			X			X	X	X	X	X	X	X		1	27
X						X				X	X				X	X		X		X								X		6	28
X			X	X	X	X	X	X	X	X	X		X				X		X	X	X			X		X	X	X		9	29

```
import numpy as np
from pyzbar import pyzbar
from tqdm import tqdm
import cv2

str = [
  "10111010111000011000001011101", "00111010110010101000000000010",
  "11100001101100100000101101O110", "10000010000111101111111111101",
  "10111010000010001000001011101", "01110011010111001001011100001",
  "10101101110101001110000110010", "10111010111010111110001011101",
```

```
  "10010100110101101111100111110", "10111110011101011000011010111",
  "11001010001011011000000001000", "10111010100101100111111110100",
  "10000010110010110100100011001", "11111110101010101010101111111",
  "10111010001100010100110110111", "11111110101101010101110110000",
  "00000000100100100101100011100", "00000000111001101011011111001",
  "11111100100110010010001011110", "00010001110001001101110001010",
  "10000010111110001101101000001", "11001110110110010101100100001",
  "11011110111101111101111110100", "11111110100010100101101011100",
  "00000000011111011001100000000", "10111010111001000110010100010",
  "11111110101110011001001111111", "10000010001100011010101000001",
  "10011111111101000101110010111"
]

def perms(elements):
    if len(elements) <=1:
        yield elements
    else:
        for perm in perms(elements[1:]):
            for i in range(len(elements)):
                yield perm[:i] + elements[0:1] + perm[i:]

evens = list(perms([3,7,9,18,19,20]))
odds = list(perms([2,6,10,11,22]))
heads = [27,21,8,5,1,28,14,25,29]
tails = [23,17,24,13,12,26,15,4,16]

with tqdm(total=720 * 120) as pbar:
    for odd in odds:
        for even in evens:
            tmporder = []
            tmporder.extend(heads)
            for i in range(5):
                tmporder.append(even[i])
                tmporder.append(odd[i])
            tmporder.append(even[5])
            tmporder.extend(tails)
            # print(tmporder)
            tmpgraph = [[]]
            for i in range(31):
                tmpgraph[0].append([255,255,255])
            for i in range(29):
                tmpgraph.append([])
                tmpgraph[i+1].append([255,255,255])
                for j in range(29):
                    if str[tmporder[i]-1][j:j+1] == '0':
                        tmpgraph[i+1].append([255,255,255])
                    else:
                        tmpgraph[i+1].append([0,0,0])
                tmpgraph[i+1].append([255,255,255])
            tmpgraph.append([])
            for i in range(31):
                tmpgraph[30].append([255,255,255])
            tmpgraph = np.array(tmpgraph, dtype=np.uint8)
            # print(tmpgraph)
            # cv2.imshow('image',tmpgraph)
            # cv2.waitKey()
            data = pyzbar.decode(tmpgraph)
            if data != []:
                text = data[0].data.decode('utf-8')
                print(text)
```

```
                print(text)
                cv2.imshow('image',tmpgraph)
                cv2.waitKey()
            pbar.update(1)
```