HGAME Week2 WriteUp

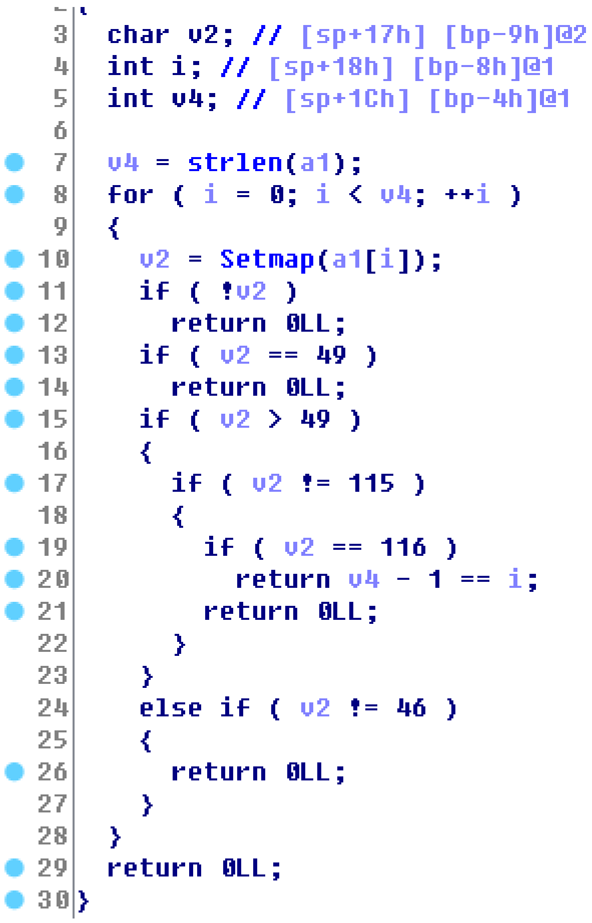
这周没做出什么题目，一来可能是第一周做题有些累，第二周做的时候没有那么有动力了，每天放空自己。然后可能心境有些烦躁，看到这些不能短时间内理解题目本意的题目就不太想做。第三可能是因为春节到了，各种事一堆？本身也没有太多完整的时间来看题，再加上总有种过节应该放松休息的想法等等。

错过了第二周的题目和知识点，很遗憾，不知道第三周能不能重新摆正心态，也不知道第三周的题目自己还能不能做出来。

到最后还是挺丧的，有点不知道该怎么办

RE

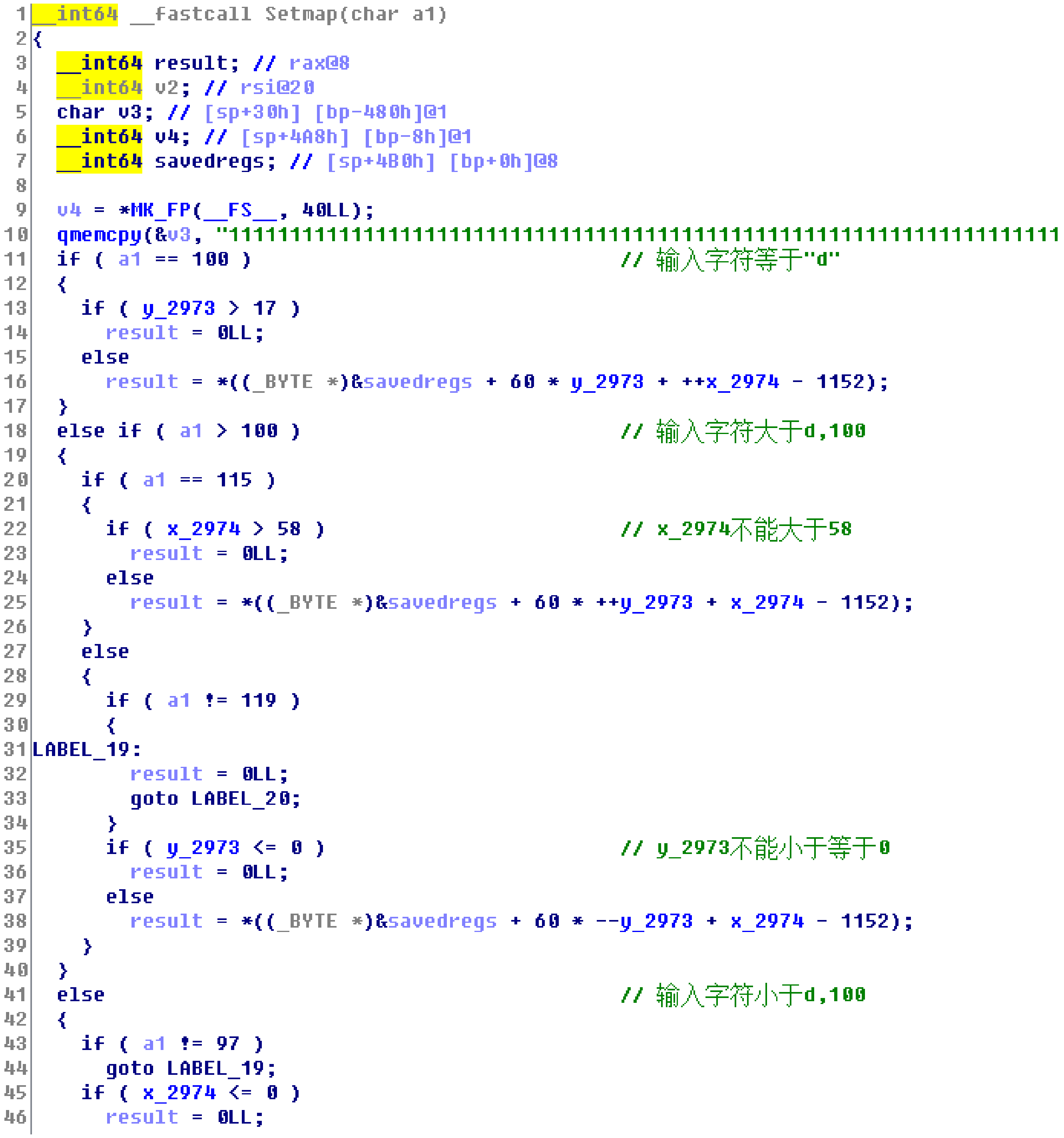
maze



```
{
  char v2; // [sp+17h] [bp-9h]@2
  int i; // [sp+18h] [bp-8h]@1
  int v4; // [sp+1Ch] [bp-4h]@1

  v4 = strlen(a1);
  for ( i = 0; i < v4; ++i )
  {
    v2 = Setmap(a1[i]);
    if ( !v2 )
      return 0LL;
    if ( v2 == 49 )
      return 0LL;
    if ( v2 > 49 )
    {
      if ( v2 != 115 )
      {
        if ( v2 == 116 )
          return v4 - 1 == i;
        return 0LL;
      }
    }
    else if ( v2 != 46 )
    {
      return 0LL;
    }
  }
  return 0LL;
}
```

外面这个函数的大意就是前面几个Setmap返回值得是"."，最后一个Setmap的返回值则是"t"

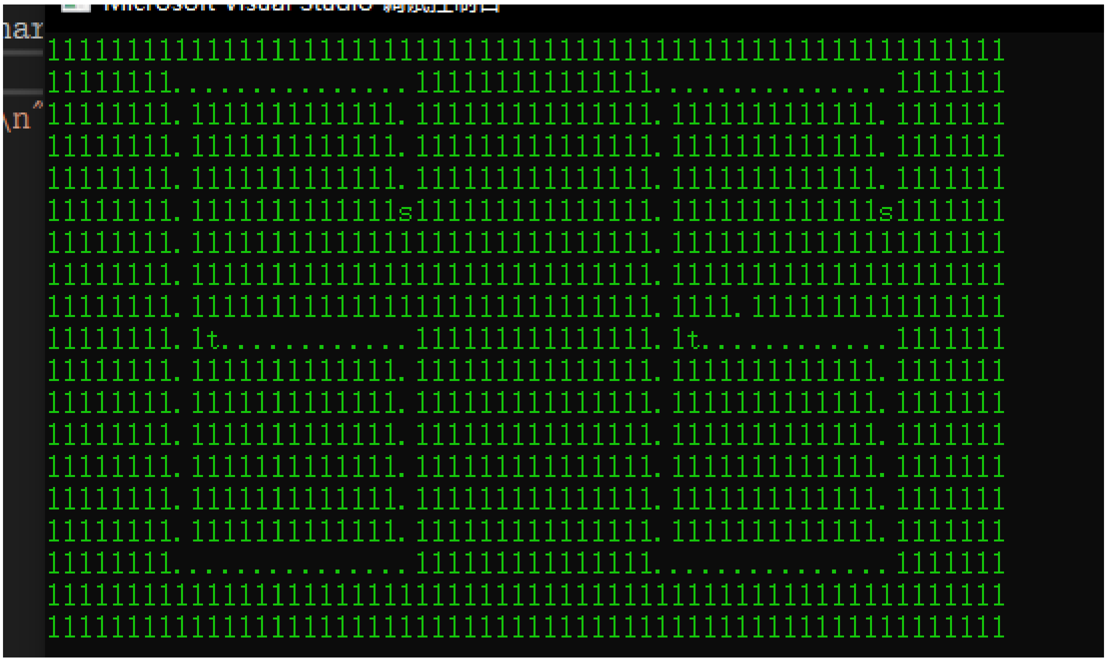
再进入Setmap函数，



```
__int64 __fastcall Setmap(char a1)
{
  __int64 result; // rax@8
  __int64 v2; // rsi@20
  char v3; // [sp+30h] [bp-480h]@1
  __int64 v4; // [sp+4A8h] [bp-8h]@1
  __int64 savedregs; // [sp+4B0h] [bp+0h]@8

  v4 = *MK_FP(__FS__, 40LL);
  qmemcpy(&v3, "11111111111111111111111111111111111111111111111111111111111111111111
  if ( a1 == 100 )                              // 输入字符等于"d"
  {
    if ( y_2973 > 17 )
      result = 0LL;
    else
      result = *((_BYTE *)&savedregs + 60 * y_2973 + ++x_2974 - 1152);
  }
  else if ( a1 > 100 )                          // 输入字符大于d,100
  {
    if ( a1 == 115 )
    {
      if ( x_2974 > 58 )                        // x_2974不能大于58
        result = 0LL;
      else
        result = *((_BYTE *)&savedregs + 60 * ++y_2973 + x_2974 - 1152);
    }
    else
    {
      if ( a1 != 119 )
      {
LABEL_19:
        result = 0LL;
        goto LABEL_20;
      }
      if ( y_2973 <= 0 )                        // y_2973不能小于等于0
        result = 0LL;
      else
        result = *((_BYTE *)&savedregs + 60 * --y_2973 + x_2974 - 1152);
    }
  }
  else                                          // 输入字符小于d,100
  {
    if ( a1 != 97 )
      goto LABEL_19;
    if ( x_2974 <= 0 )
      result = 0LL;
```

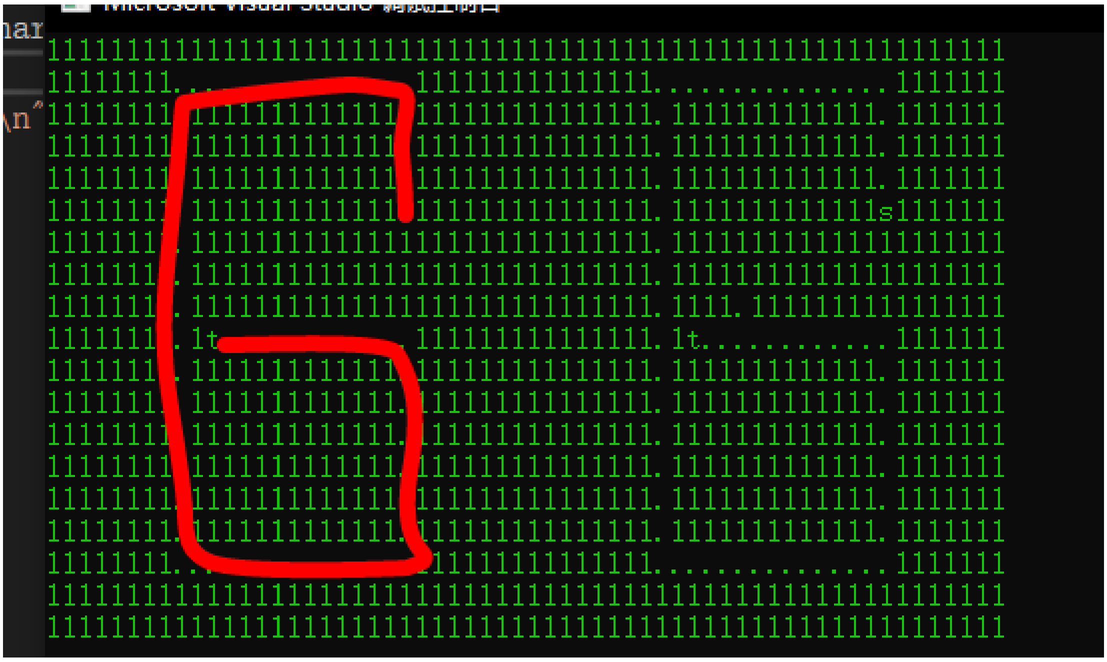
发现是走迷宫，这个迷宫是60个一行的，所以 y_2973 代表 y轴，x_2974 代表 x 轴

所以先生成迷宫



```
111111111111111111111111111111111111111111111111111111111111
11111111...............111111111111111...............1111111
11111111.1111111111111.111111111111111.1111111111111.1111111
11111111.1111111111111.111111111111111.1111111111111.1111111
11111111.1111111111111.111111111111111.1111111111111.1111111
11111111.1111111111111s111111111111111.1111111111111s1111111
11111111.11111111111111111111111111111.111111111111111111111
11111111.11111111111111111111111111111.111111111111111111111
11111111.11111111111111111111111111111.1111.1111111111111111
11111111.1t............111111111111111.1t............1111111
11111111.1111111111111.111111111111111.1111111111111.1111111
11111111.1111111111111.111111111111111.1111111111111.1111111
11111111.1111111111111.111111111111111.1111111111111.1111111
11111111.1111111111111.111111111111111.1111111111111.1111111
11111111.1111111111111.111111111111111.1111111111111.1111111
11111111.1111111111111.111111111111111.1111111111111.1111111
11111111...............111111111111111...............1111111
111111111111111111111111111111111111111111111111111111111111
111111111111111111111111111111111111111111111111111111111111
```

从"s"开始，走到"t"就行了



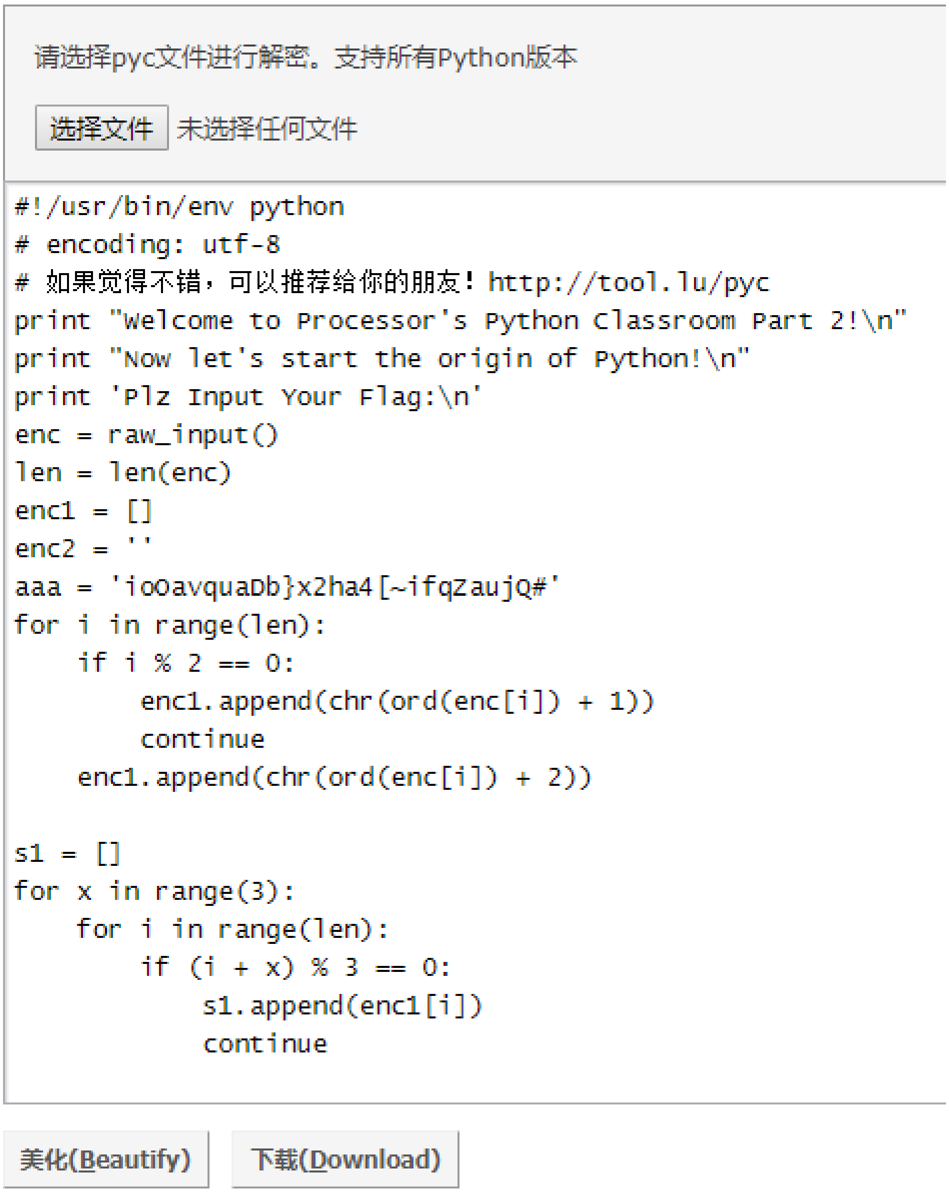
```
111111111111111111111111111111111111111111111111111111111111
11111111...............111111111111111...............1111111
11111111.1111111111111.111111111111111.1111111111111.1111111
11111111.1111111111111.111111111111111.1111111111111.1111111
11111111.1111111111111.111111111111111.1111111111111.1111111
11111111.1111111111111s111111111111111.1111111111111s1111111
11111111.11111111111111111111111111111.111111111111111111111
11111111.11111111111111111111111111111.111111111111111111111
11111111.11111111111111111111111111111.1111.1111111111111111
11111111.1t............111111111111111.1t............1111111
11111111.1111111111111.111111111111111.1111111111111.1111111
11111111.1111111111111.111111111111111.1111111111111.1111111
11111111.1111111111111.111111111111111.1111111111111.1111111
11111111.1111111111111.111111111111111.1111111111111.1111111
11111111.1111111111111.111111111111111.1111111111111.1111111
11111111.1111111111111.111111111111111.1111111111111.1111111
11111111...............111111111111111...............1111111
111111111111111111111111111111111111111111111111111111111111
111111111111111111111111111111111111111111111111111111111111
```

得到flag

hgame{wwwwaaaaaaaaaaaaaasssssssssssssssddddddddddddddwwwwwwwaaaaaaaaaaaa}

Pro的Python教室(二)

发现下载到的是一个.pyc，查了一下发现有在线反编译工具

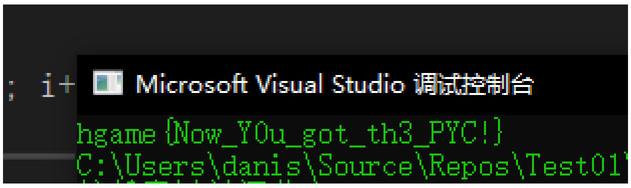


请选择pyc文件进行解密。支持所有Python版本

```python
#!/usr/bin/env python
# encoding: utf-8
# 如果觉得不错，可以推荐给你的朋友！http://tool.lu/pyc
print "Welcome to Processor's Python Classroom Part 2!\n"
print "Now let's start the origin of Python!\n"
print 'Plz Input Your Flag:\n'
enc = raw_input()
len = len(enc)
enc1 = []
enc2 = ''
aaa = 'ioOavquaDb}x2ha4[~ifqZaujQ#'
for i in range(len):
    if i % 2 == 0:
        enc1.append(chr(ord(enc[i]) + 1))
        continue
    enc1.append(chr(ord(enc[i]) + 2))

s1 = []
for x in range(3):
    for i in range(len):
        if (i + x) % 3 == 0:
            s1.append(enc1[i])
            continue
```

这题还是比较易懂的，写了一个逆向代码

```c
int main()
{
    char a[] = "ioOavquaDb}x2ha4[~ifqZaujQ#";
    int b[] = { 0,3,6,9,12,15,18,21,24,2,5,8,11,14,17,20,23,26,1,4,7,10,13,16,19,22,25};
    char c[27];
    for (int i = 0; i < 27; i++)
    {
        c[b[i]] = a[i];
    }
    for (int i = 0; i < 27; i++)
    {
        if (i % 2 == 0)
        {
            c[i] -= 1;
        }
        else
        {
            c[i] -= 2;
        }
        putchar(c[i]);
    }
}
```



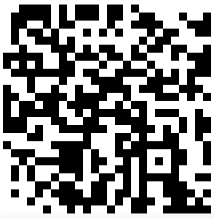
```
hgame{Now_Y0u_got_th3_PYC!}
C:\Users\danis\Source\Repos\Test01\
```

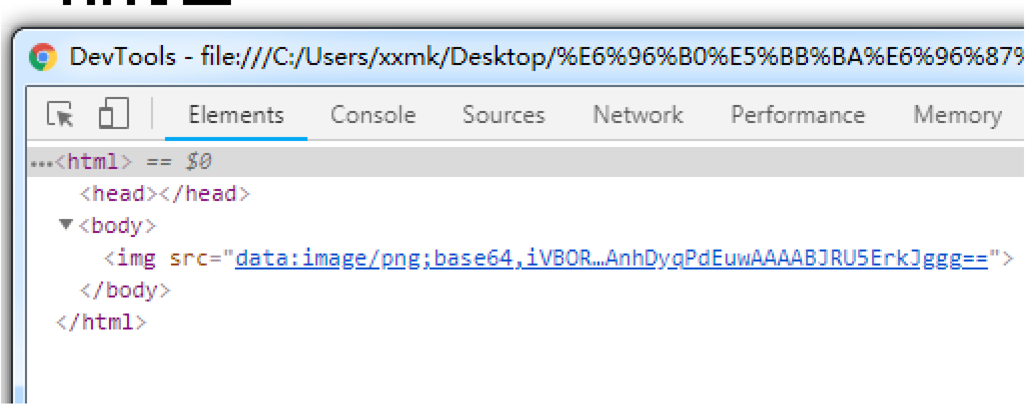
得到flag

MISC

初识二维码

得到的内容是base64编码的图片，所以在html里加上就行了

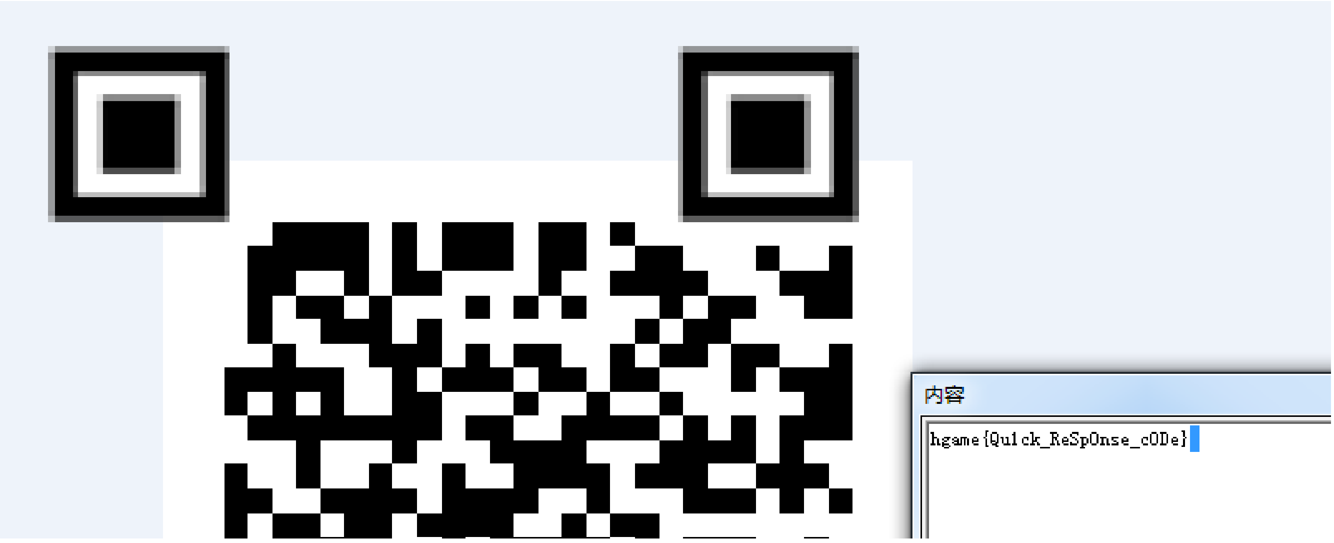




DevTools - file:///C:/Users/xxmk/Desktop/%E6%96%B0%E5%BB%BA%E6%96%879
Elements
Console
Sources
Network
Performance
Memory

```
…<html> == $0
  <head></head>
 ▼<body>
    <img src="data:image/png;base64,iVBOR…AnhDyqPdEuwAAAABJRU5ErkJggg==">
  </body>
</html>
```

得到残缺的二维码，扫了一下发现扫不出来，然后发现少了一些东西，看了看大多数的二维码发现缺了用来定位的三个正方形，所以PS里加上



```
hgame{Qu1ck_ReSp0nse_c0De}
```

扫描得到flag

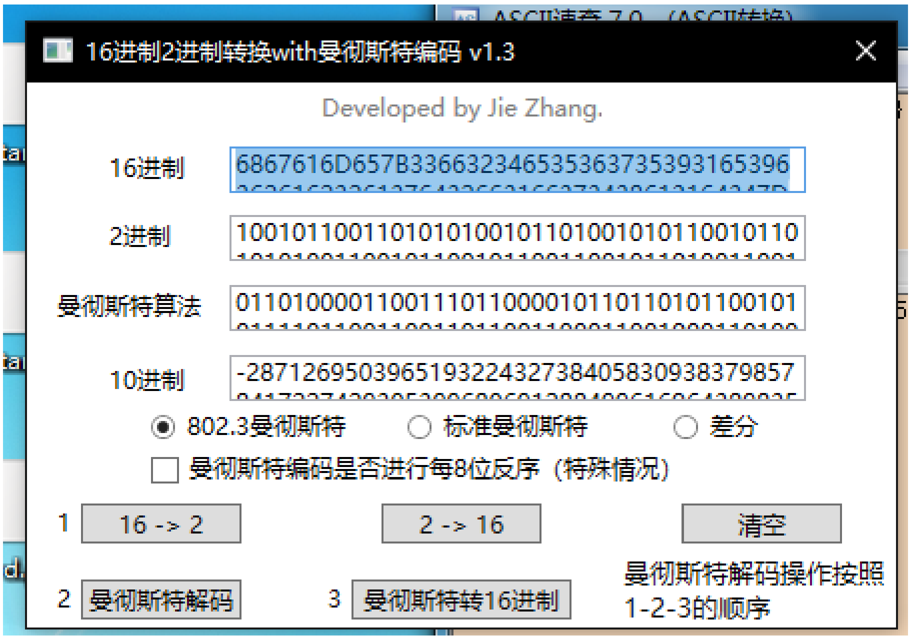
CRYPTO

浪漫的足球圣地

查了一下题目，发现浪漫的足球圣地是指曼切斯特

然后查了一下相关的内容，发现了一个曼切斯特编码

给的内容是十六进制数，所以要转换成二进制先


16进制2进制转换with曼彻斯特编码 v1.3

Developed by Jie Zhang.

16进制 6867616D657B336632346535363735393165396

2进制 1001011001101010100101101001010110010110

曼彻斯特算法 0110100001100111011000010110110101100101

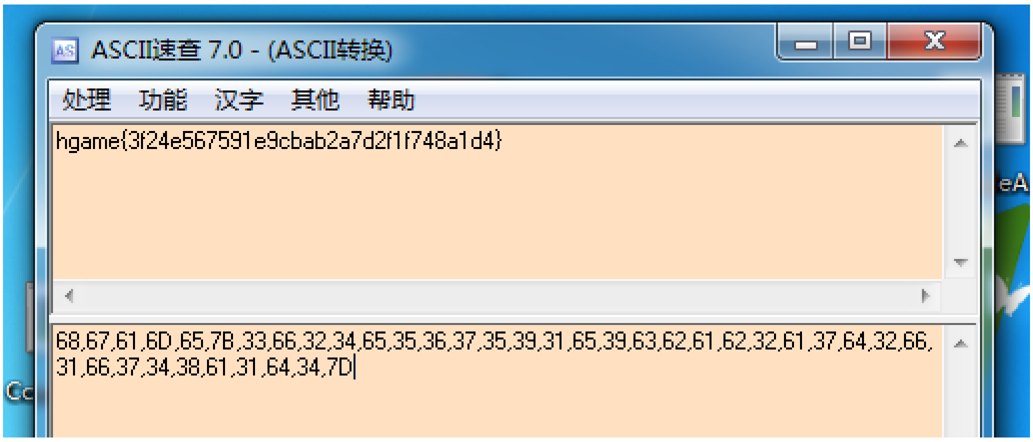
10进制 -287126950396519322432738405830938379857

● 802.3曼彻斯特 ○ 标准曼彻斯特 ○ 差分

□ 曼彻斯特编码是否进行每8位反序（特殊情况）

曼彻斯特解码操作按照1-2-3的顺序

最后得到一串十六进制数，其实是ASCII码


ASCII速查 7.0 - (ASCII转换)

```
hgame{3f24e567591e9cbab2a7d2f1f748a1d4}
```

```
68,67,61,6D,65,7B,33,66,32,34,65,35,36,37,35,39,31,65,39,63,62,61,62,32,61,37,64,32,66,
31,66,37,34,38,61,31,64,34,7D
```

得到flag

hill

这道题想了蛮长时间的，但是python实在写不好，写了半天也不知道为啥就是跑不出来，改了很多次，最终也没有得到flag，先把有问题的脚本放在这里了，希望学长能指一下哪里出错了

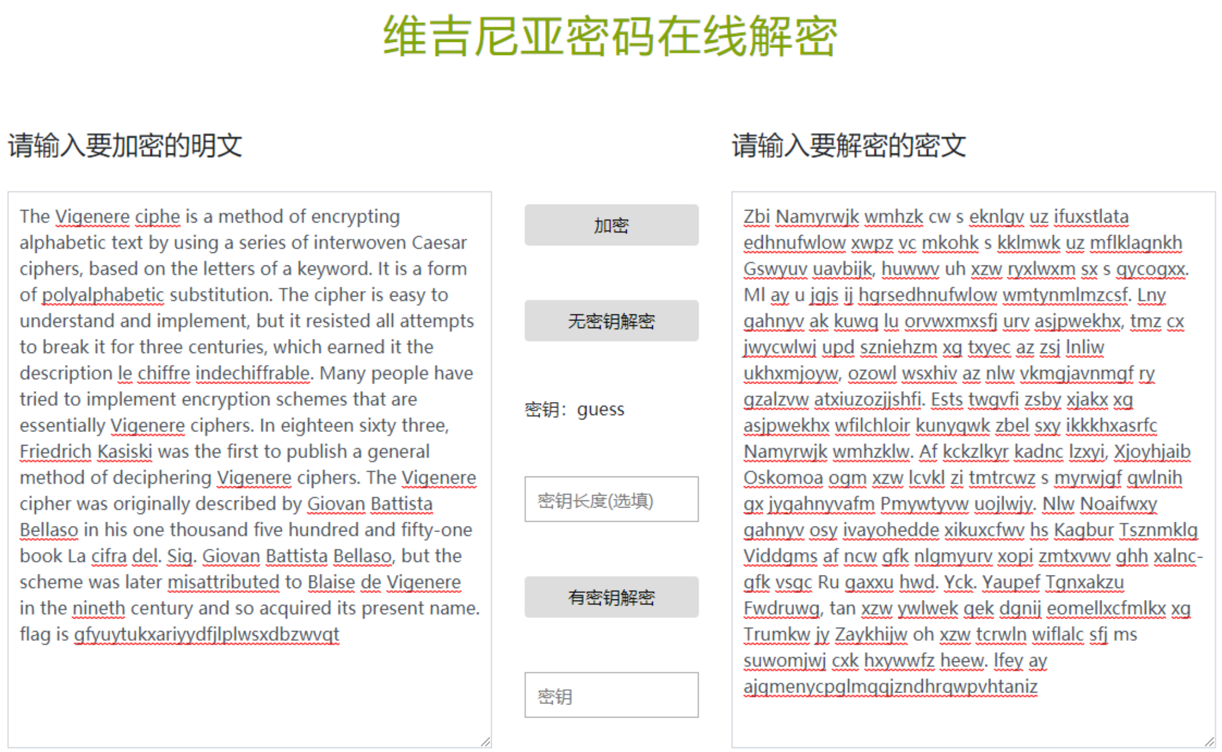
```
from numpy import *

word = [[0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0]]
new = [[0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0]]
word2 = [[0,0,0],[0,0,0],[0,0,0]]
word3 = [[0,0,0],[0,0,0],[0,0,0]]
string=''
```

```
a1 = 'TCS'
a2 = 'HXZ'
a3 = 'TCX'
a4 = 'APB'
a5 = 'DKJ'
a6 = 'VJD'
a7 = 'OHJ'
a8 = 'EAE'
e = ''
answer = ''
for i in range(3):
    word[i][0] = ord(a1[i]) - ord('A') + 1
    word[i][1] = ord(a2[i]) - ord('A') + 1
    word[i][2] = ord(a3[i]) - ord('A') + 1
    word[i][3] = ord(a4[i]) - ord('A') + 1
    word[i][4] = ord(a5[i]) - ord('A') + 1
    word[i][5] = ord(a6[i]) - ord('A') + 1
    word[i][6] = ord(a7[i]) - ord('A') + 1
    word[i][7] = ord(a8[i]) - ord('A') + 1

A = mat(word)
#print 'A = '
#print A
#BABYSHILL
d = 'BABYSHILL'
for i in range(3):
    word2[0][i] = ord(d[i*3]) - ord('A') + 1
    word2[1][i] = ord(d[i*3+1]) - ord('A') + 1
    word2[2][i] = ord(d[i*3+2]) - ord('A') + 1
B = mat(word2)
#print 'B = '
#print B
#print 'B_1 = '
#print B_1
s = 'TCSHXZTCXAPBDKJVJDOHJEAE'
for k in range(15):
    e = s[k:k+10]

    print e

    for j_1 in range(3):
        for j_2 in range(3):
            for j_3 in range(3):
                for j_4 in range(3):
                    for j_5 in range(3):
                        for j_6 in range(3):
                            for j_7 in range(3):
                                for j_8 in range(3):
                                    for j_9 in range(3):
                                        word3[0][0] = ord(e[0]) - ord('A') + 1 + j_1 *
26
                                        word3[1][0] = ord(e[1]) - ord('A') + 1 + j_2 *
26
```

```
                                        word3[2][0] = ord(e[2]) - ord('A') + 1 + j_3 *
26
                                        word3[0][1] = ord(e[3]) - ord('A') + 1 + j_4 *
26
                                        word3[1][1] = ord(e[4]) - ord('A') + 1 + j_5 *
26
                                        word3[2][1] = ord(e[5]) - ord('A') + 1 + j_6 *
26
                                        word3[0][2] = ord(e[6]) - ord('A') + 1 + j_7 *
26
                                        word3[1][2] = ord(e[7]) - ord('A') + 1 + j_8 *
26
                                        word3[2][2] = ord(e[8]) - ord('A') + 1 + j_9 *
26
                                        C = mat(word3)
#print C
                                        try:
                                            C_1 = C.I
#print C_1

                                            K = B *C_1
#print K
                                            K_1 = K.I

                                            AN = K * A
                                            #print AN

                                            for k in range(8):
                                                for l in range(3):
                                                    if AN[l][k] > 0:
                                                        answer += chr(int(AN[l]
[k])%26+ord('A') - 1)
                                                    else:
                                                        #print e + ' error ' + str(j_1)
+str(j_2)+str(j_3)+str(j_4)+str(j_5)+str(j_6)+str(j_7)+str(j_8)+str(j_9)
                                                        break
                                            print answer
                                            answer = ''

                                        except:
                                            #print e + ' error0'
                                            pass
```

Vigener~

查了一下题目发现是维吉尼亚密码，然后用了一个在线解密



维吉尼亚密码在线解密

请输入要加密的明文

The Vigenere ciphe is a method of encrypting alphabetic text by using a series of interwoven Caesar ciphers, based on the letters of a keyword. It is a form of polyalphabetic substitution. The cipher is easy to understand and implement, but it resisted all attempts to break it for three centuries, which earned it the description le chiffre indechiffrable. Many people have tried to implement encryption schemes that are essentially Vigenere ciphers. In eighteen sixty three, Friedrich Kasiski was the first to publish a general method of deciphering Vigenere ciphers. The Vigenere cipher was originally described by Giovan Battista Bellaso in his one thousand five hundred and fifty-one book La cifra del. Sig. Giovan Battista Bellaso, but the scheme was later misattributed to Blaise de Vigenere in the nineth century and so acquired its present name. flag is gfyuytukxariyydfjlplwsxdbzwvqt

密钥：guess

请输入要解密的密文

Zbi Namyrwjk wmhzk cw s eknlgv uz ifuxstlata edhnufwlow xwpz vc mkohk s kklmwk uz mflklagnkh Gswyuv uavbijk, huwwv uh xzw ryxlwxm sx s qycogxx. Ml ay u jgjs ij hgrsedhnufwlow wmtynmlmzcsf. Lny gahnyv ak kuwq lu orvwxmxsfj urv asjpwekhx, tmz cx jwycwlwj upd szniehzm xg txyec az zsj lnliw ukhxmjoyw, ozowl wsxhiv az nlw vkmgjavnmgf ry gzalzvw atxiuzozjjshfi. Ests twgvfi zsby xjakx xg asjpwekhx wfilchloir kunyqwk zbel sxy ikkkhxasrfc Namyrwjk wmhzklw. Af kckzlkyr kadnc lzxyi, Xjoyhjaib Oskomoa ogm xzw lcvkl zi tmtrcwz s myrwjgf qwlnih gx jygahnyvafm Pmywtyvw uojlwjy. Nlw Noaifwxy gahnyv osy ivayohedde xikuxcfwv hs Kagbur Tsznmklg Viddgms af ncw gfk nlgmyurv xopi zmtxvwv ghh xalnc-gfk vsgc Ru gaxxu hwd. Yck. Yaupef Tgnxakzu Fwdruwg, tan xzw ywlwek qek dgnij eomellxcfmlkx xg Trumkw jy Zaykhijw oh xzw tcrwln wiflalc sfj ms suwomjwj cxk hxywwfz heew. Ifey ay ajqmenycpglmqqjzndhrqwpvhtaniz