

Re

maze

用ida打开，f5 分析程序

发现一串奇怪的字符串

[illegible]

下面是不死的情况

```
v2 == 46
v2 == 115
if(1==v4-1) v2 == 116

a1 == 100 && y_2973<=17
a1==115 && x_2974 <=58
a1==119 && y_2973>0
a1==97 && x_2974>0
```

查了一下ASCII。嗯？不就是打游戏么，'a'后退一格，'d'前进一格，'w'向前跳跃60格，'s'向后跳跃60格\打游戏是不可能打游戏的，那就开挂吧（主要是当时觉得挺复杂的，总不能用生命打游戏吧）\于是乎，脚本如下：

[illegible]

```
str_box_new=[]
if is_end:
    break
```

这是最终版本。前几个版本由于没有意识到无意义的移动，跑了很久很久很久。。。就加了

```
and not (a2[-2:] in ['sw','ws','da','ad'])
```

分分钟End!

brainfxxker's revenge

一开始想爆破来着（还是太天真了），于是不怀好意地py出题人去了。\\知道了位数正确会输出'w rongansw er!或'congratulations!\\一开始美滋滋，后来一想就觉得不对，我问之前自己也乱输了好多字符没啥动静啊。一测，72个字符。。。我爆你个锤子啦\\后来盯着'代码'看就久了也知道要化简，怎么化简呢？（1+1-1型的已经化简了）\\

```
#brainfucker's revenge
str1='>,,>+++++[<----->-][*<<--[<+>[-]][]>[<.<--[+>-<]-<.]][<[,>.[<+>][,>.,[,>.,---<]-<[>>+][[->,+] [,>>>]>]++<][<[,>,<+,,]>]>-<][[-<.,-<--]+].>+<]<-,>,,>+++++

str2=str1.replace('+-', '')
str2=str1.replace('--+', '')
str2=str1.replace('<><', '')
str2=str1.replace('<>>', '')
print(str2)
```

我想到了在72个字符输入完成之前，没有任何字符输出，这说明'没有被执行。\\于是乎，我就去消'了。于是乎，完蛋了。\\代码确实变干净了，但就是消不干净（这哪是两个脚本能解决的事情?!）\\不过在这个过程中，我发现还有一种可以化简代码的情况:\\\u2191,能够从\u2191里出来的都是真男人都是0，当0遇上\u2191，简直完美！\\于是乎，代码如下：

```
#brainfucker's revenge
str='>,,>+++++[<----->-][*<<--[<+>[-]][]>[<.<--[+>-<]-<.]][<[,>.[<+>][,>.,[,>.,---<]-<[>>+][[->,+] [,>>>]>]++<][<[,>,<+,,]>]>-<][[-<.,-<--]+].>+<]<-,>,,>+++++[

def huajian(i,str):
    unmatched=0
    for n in range(i+1,len(str)):
        if str[n]=='[':
            unmatched+=1
        if str[n]==']':
            unmatched-=1
            box=n
        if unmatched==0:
            break
    a='0'*(box-i) #加'0'看到到底是哪些地方被删掉了
    str=str[:i+1]+a+str[box+1:]
    return[str]

for i in range(len(str)-1):
    if str[i]=='[' and str[i+1]=='[':
        str+=''.join(huajian(i,str))

str2=''
for i in range(len(str)):
    if str[i]!='0':
        str2=str2+str[i]
print(str2)

#以','为标志，分条打印
count=0;n=0
for i in range(len(str2)):
    if str2[i]==',':
        count+=1
        str3=str2[n:i]
        print(str3[1:])
        n=i-1
print(str2[n+1:])
print(count+1)
```

最后是放到bf1的代码里跑出flag

Pro的Python教室(二)

在线反编译

请选择pyc文件进行解密。支持所有Python版本

浏览...

```
#!/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 = 'io0avquaDb}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
```

话不多说，上脚本：

```
a1 = 'io0avquaD'
a2 = 'b}x2ha4[~'
a3 = 'ifqZaujQ#'
s1=[]
s2=''
for i in range(len(a1)):
    s1.append(a1[i])
    s1.append(a3[i])
    s1.append(a2[i])
s2=''.join(s1)

for i in range(len(s1)):
    if i % 2 == 0:
        s1[i]=chr(ord(s1[i])-1)
        continue
    s1[i]=chr(ord(s1[i])-2)
s2=''.join(s1)
print(s2)
```

其实有个问题\

```
>>> s = "Alex aab"
>>> ret = "aa" in s
>>> print(ret)
True
>>> ret = "ab" in s
>>> print(ret)
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    printf(ret)
NameError: name 'printf' is not defined
>>> print(ret)
True
>>> ret = "ab" in s
>>> print(ret)
True
>>> ret = "sds" in s
>>> print(ret)
False
```

```
enc2 = enc2.join(s1)
if enc2 in aaa:
    print "You 're Right!"
else:
    print "You're Wrong!"
    exit(0)
```

所以并不需要enc2==aaa吧。

CRYPTO

Vigener

→ ↺ 🏠

68.168.134.3/vigener/

📄 ☆ ⚙️ 📌

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 indechiffable. 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 gfyutyukxarjydtjlpwsxdbzwwqt

这才是真正的song-fen啊

请输入要解密的密文

加密

无密钥解密

密钥: guess

密钥长度(选填)

有密钥解密

Zbi Namyrwj k wmhzk cw s eknlgv uz ifuxstlata edhnufwlow xwpz vc mkohk s kklmwk uz mflklagnkh Gswyuv uavbjk, huwvv uh xzw ryxlwxm sx s qycogxx. Ml ay u jgjs ij hgrsedhnufwlow wmtynmlmzcsf. Lny gahnyv ak kuwq lu orvwmxsfj urv asjpwekhx, tmz cx jwycwlwj upd szniehzm xg txyec az zsj lnliw ukhxmjoyw, ozowl wsxhiv az nlw vkmgjavnmgf ry gzalzv atxiuzozjjshfi. Ests twgvfi zsby xjakh xg asjpwekhx wfilchloir kunyqwk zbel sxy ikkkhasrfc Namyrwj wmhzklw. Af kckzlkyr kadnc lzxyi, Xjoyhjaib Oskomoa ogm xzw lcvkl zi tmtcrwz s myrwjgf qwl nih gx jygahnyvafm Pmywtyvw uojlwjy. Nlw Noaifwxy gahnyv osy ivayohedde xikuxcfwv hs Kagbur Tsznmklg Viddgms af ncw gfk nlgmyurv xopi zmtxvww ghx xalnc-gfk vsgc Ru gaxxu hwd. Yck. Yaupef Tgnxakzu Fwdruwg, tan xzw ywlwek qek dgnij eomellxcfmkx xg Trumkw jy Zaykhijw oh xzw tcrwl n wifalc sfj ms suwomjwj ckk hxywwfz heew. Ifey ay ajqmenycpglmqjzndhrqwpvhtaniz