

# 2020 巅峰极客 Re部分WriteUp

原创

Hk\_Mayfly 于 2020-09-29 11:15:00 发布 373 收藏 1

文章标签: [字符串](#) [python](#) [算法](#) [php](#) [列表](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/qq\\_39542714/article/details/108879759](https://blog.csdn.net/qq_39542714/article/details/108879759)

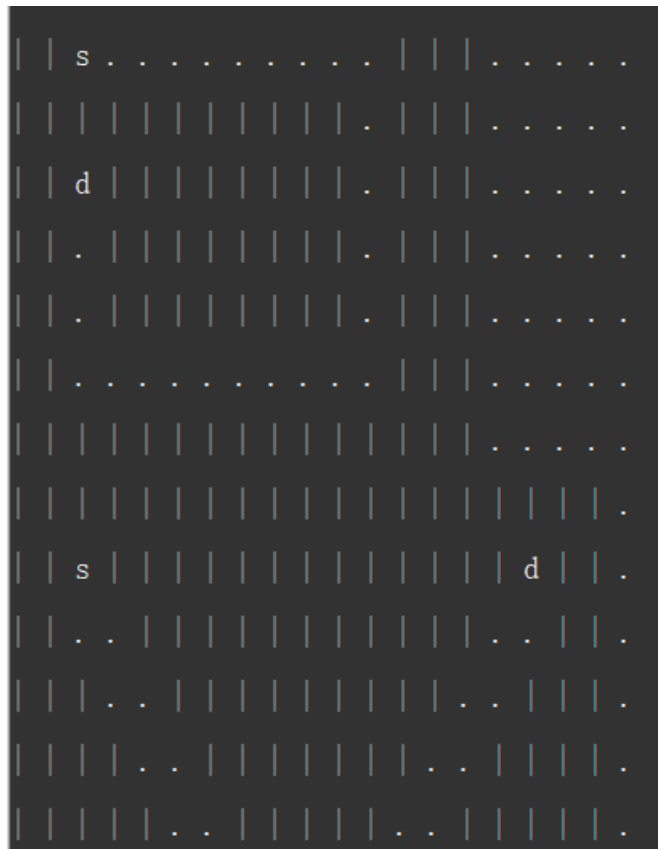
版权

## virus

附件下载: <https://wwa.lanzous.com/iZh5tgy9gbg>

代码分为三部分

1. 以'-'为间隔, 将flag的第一部分转换为整型数字, 并且满足后项-前项分别为[0x13,0x19,0x1a,0x1c]。最后一项为len(flag)-lastpos('-')
2. 以'-'为间隔, 将中间的字符串分别存储到road中
3. checkflag, 即走四个20x...的迷宫, 从s->d。以第1点存储的顺序, 决定迷宫的顺序。通过确定后面的字符串顺序, 前面的数字也会被确定。



1~4为: -ddddddddddssssssaaaaaaaawww-sdsdsdsdsdsdsddwdwdwdwdwdwdw-  
aaaaaaaaassssssssdddddddddd-wwwwwddddddddddsssss

```

import itertools
a = ['-dddddddddsssssaaaaaaaaawww', '-sdsdsdsdsdsdsddwdwdwdwdwdw', '-aaaaaaaaassssssddddddddd', '-wwwwwddd
b = list(itertools.permutations(a, len(a)))
s = []
m = [0x13,0x19,0x1a,0x1c]
for val in b:
s.append(''.join(val))
for elem in s:
index = -1
pos = []
while elem.find('-',index+1) != -1:
index = elem.find('-', index+1)
pos.append(index)
pos.append(len(elem))
print (pos)
flag = True
right = 0
for k in range(len(pos)-1):
right = k
if(pos[k+1] - pos[k] - 1 != m[k]):
flag = False
break
if (flag == True):
print (elem)

```

flag{4312-wwwwwdddddddddsssss-aaaaaaaaassssssssddddddddd-dddddddddsssssaaaaaaaaawww-sdsdsdsdsdsdsddwdwdwdwdwdwdw}

## fu!k\_py

附件下载: <https://wwa.lanzous.com/iy2Hegy9g9e>

将pyc文件转换为py

```

# uncompyle6 version 3.4.0
# Python bytecode 2.7 (62211)
# Decompiled from: Python 3.7.1 (default, Dec 10 2018, 22:54:23) [MSC v.1915 64 bit (AMD64)]
# Embedded file name: test233_ol.py
# Compiled at: 2020-03-20 13:22:50
(lambda __g, __print: [ (lambda __after: [ (lambda __after: (__print('Error len!'), (exit(), __after)),
for __g['input'] in [raw_input('Input your flag:')] ][0] if __name__ == '__main__' else __after)))(
for __g['check1'], check1.__name__ in [(lambda arg: (lambda __l: [ (lambda __after: 0 if len(list(arg)) == 9 else __after))]
for __g['check2'], check2.__name__ in [(lambda arg: (lambda __l: [ sum(map(int, __l['arg'])) for __l in arg])]
# okay decompiling fu!k.pyc

```

d,l,h分别表示3x3块，列，行

check1检测长度是否为9

check2求和

实际就是一个9x9的数独游戏，根据已知数据，求解未知就行。

1	4	5	3	2	7	6	9	8
8	3	9	6	5	4	1	2	7
6	7	2	9	1	8	5	4	3
4	9	6	1	8	5	3	7	2
2	1	8	4	7	3	9	5	6
7	5	3	2	9	6	4	8	1
3	6	7	5	4	2	8	1	9
9	8	4	7	6	1	2	3	5
5	2	1	8	3	9	7	6	4

**1 2 3 4 5 6 7 8 9**

flag{145327698839654127672918543496185372218473956753296481367542819984761235521839764}