BUUCTF 每日打卡 2021-4-29

原创

Σ2333! 于 2021-04-29 21:03:16 发布 93 收藏

分类专栏： crypto 文章标签： python 密码学

本文链接：https://blog.csdn.net/weixin_52446095/article/details/116274819

版权

crypto 专栏收录该内容

79 篇文章 1 订阅

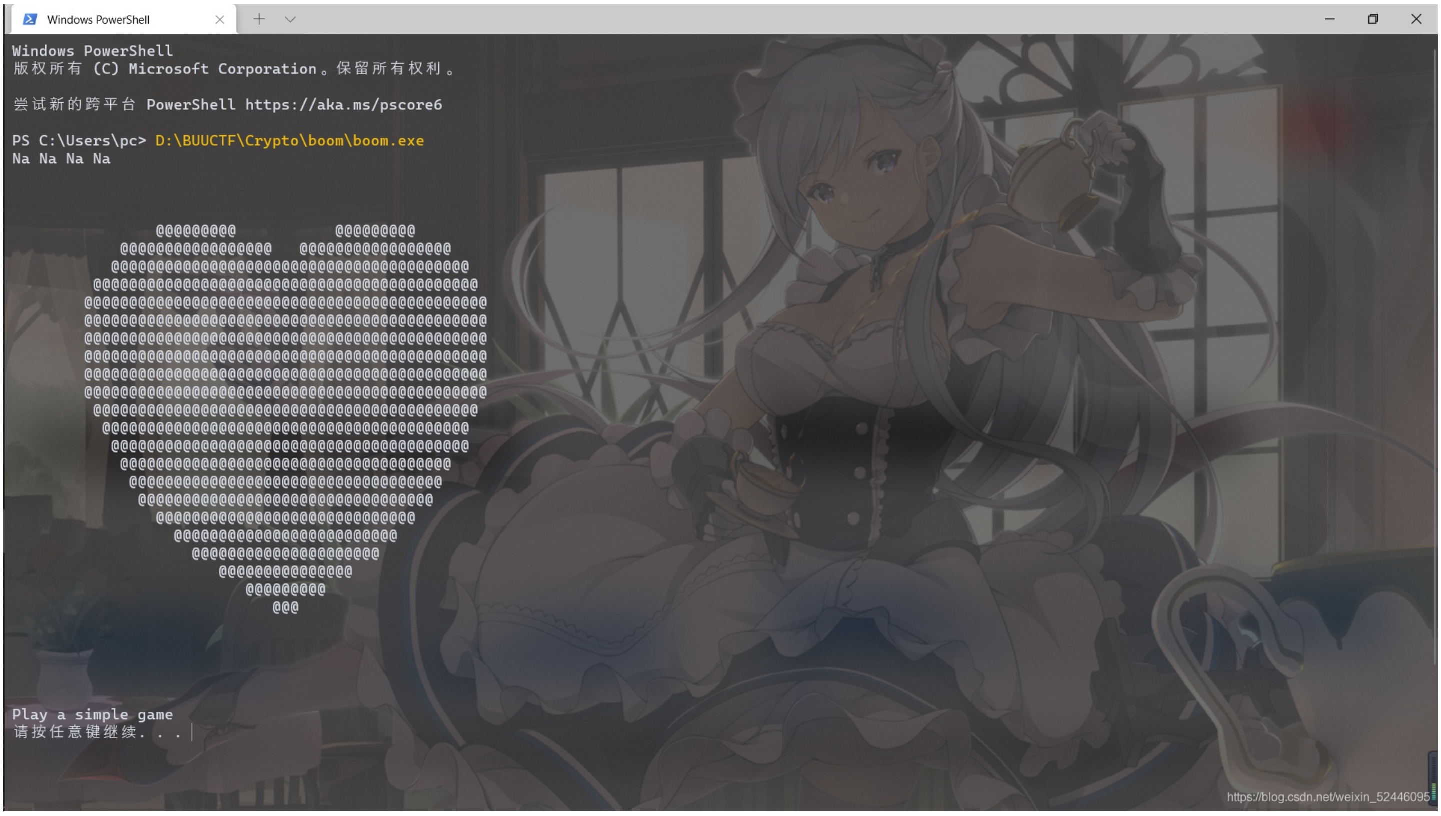
订阅专栏

引言

蓝帽杯就一道 crypto
又是斐波那契数列，又是 AES
网上查到什么斐波那契数列双混沌加密
反正就是没做出来。。。
等什么时候 wp 出来再说吧。。。

boom

附件是一个 .exe 文件
不会是个 re 题吧。。。
在命令行打开（如果不在命令行打开，最后输出会直接关闭窗口）

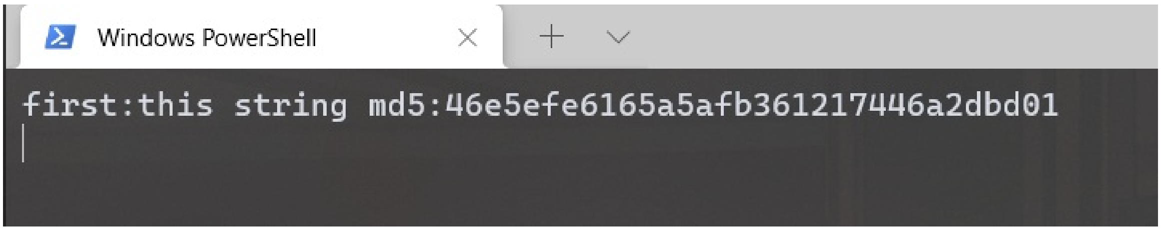


```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\pc> D:\BUUCTF\Crypto\boom\boom.exe
Na Na Na Na



                @@@@@@@@@           @@@@@@@@@
            @@@@@@@@@@@@@@@@@   @@@@@@@@@@@@@@@@@
           @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
         @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
        @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
        @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
        @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
        @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
        @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
        @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
         @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
          @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
           @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
            @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
             @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
              @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
                @@@@@@@@@@@@@@@@@@@@@@@@@@@@@
                  @@@@@@@@@@@@@@@@@@@@@@@@@
                    @@@@@@@@@@@@@@@@@@@@@
                       @@@@@@@@@@@@@@@
                          @@@@@@@@@
                             @@@




Play a simple game
请按任意键继续. . .
```

Do you like van♂ you see?（大雾）

Do you like van♂ you see?（大雾）

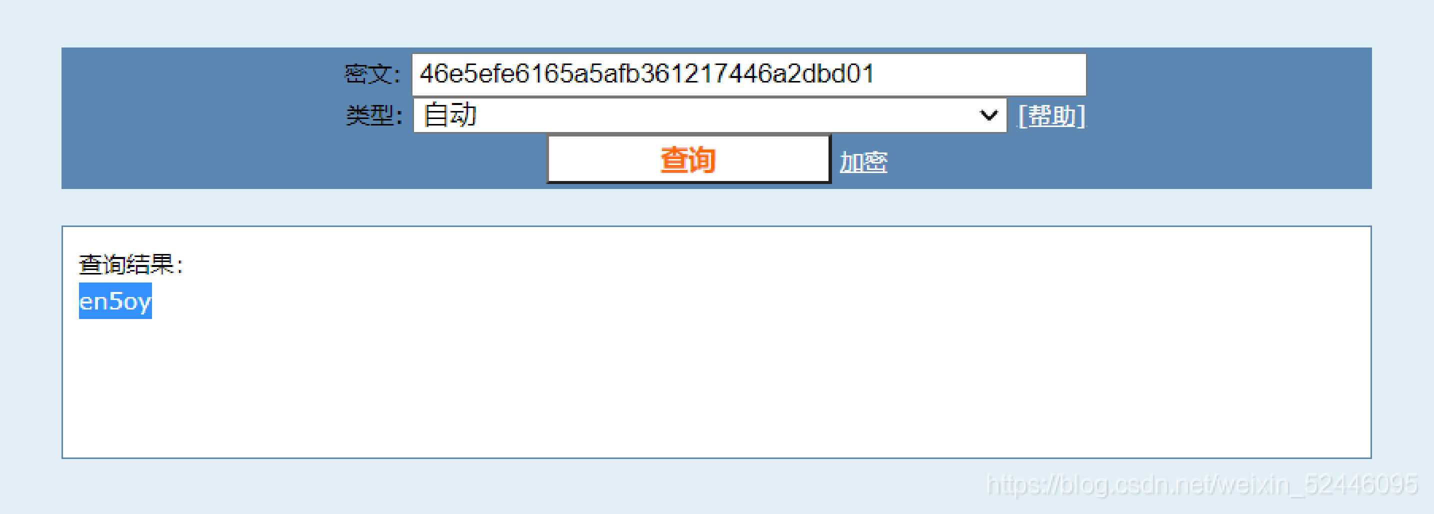
下一步



```
first:this string md5:46e5efe6165a5afb361217446a2dbd01
```

提交 md5

可以直接查询得到

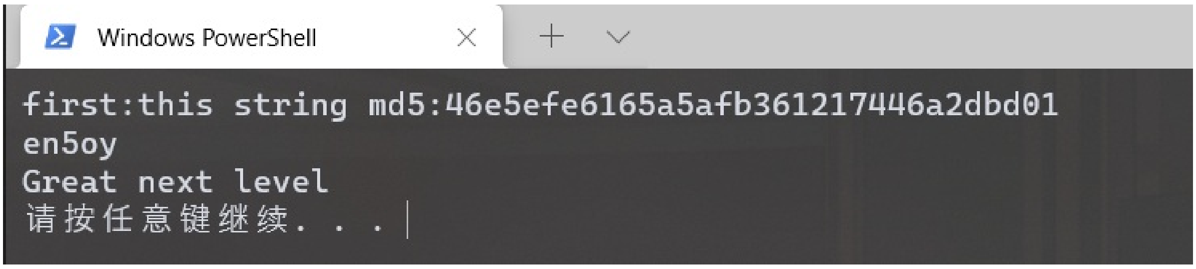


密文：46e5efe6165a5afb361217446a2dbd01

类型：自动

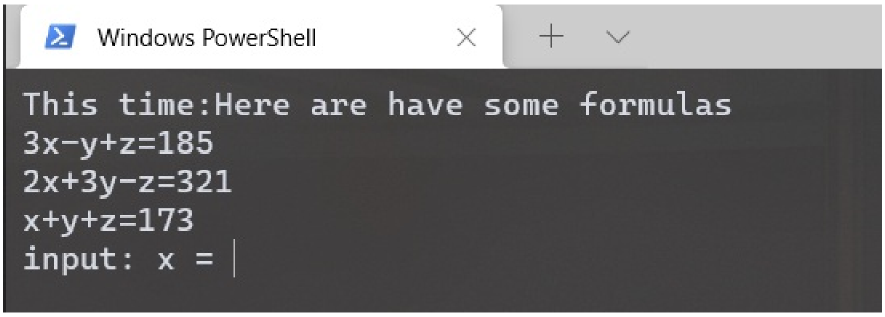
查询结果：
en5oy

输入结果



```
first:this string md5:46e5efe6165a5afb361217446a2dbd01
en5oy
Great next level
请按任意键继续. . .
```

下一步



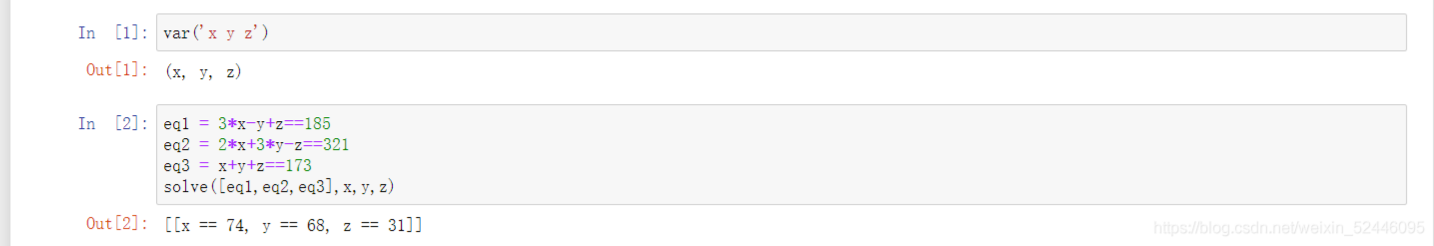
```
This time:Here are have some formulas
3x-y+z=185
2x+3y-z=321
x+y+z=173
input: x =
```

解三元一次方程组

当然你可以手算

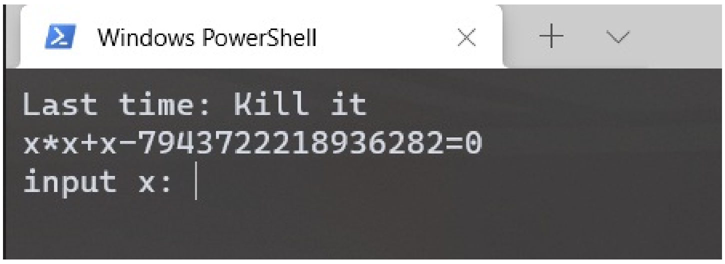
这里我们直接用 sagemath 计算（躺）

结果如下：



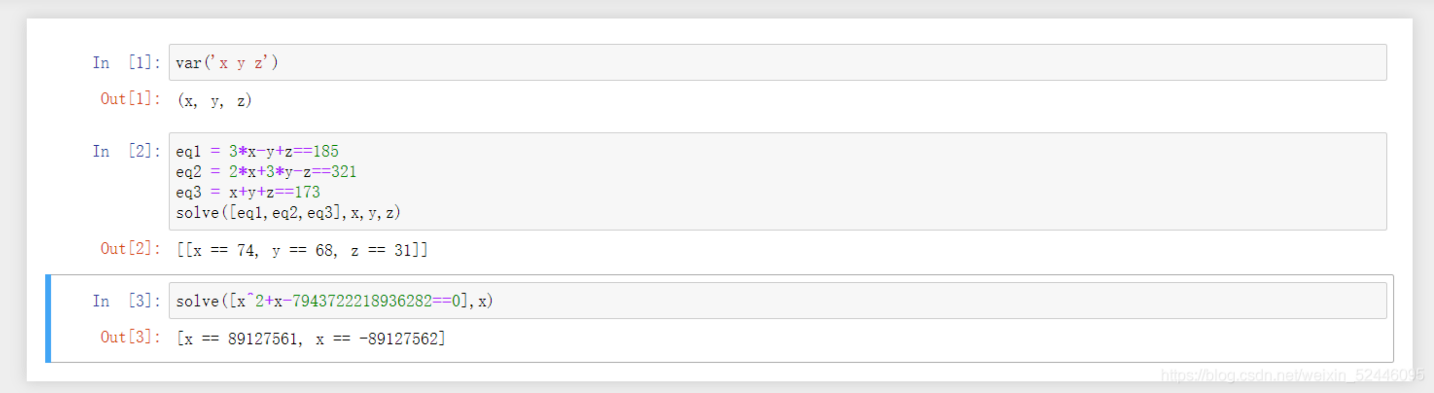
```
In [1]: var('x y z')
Out[1]: (x, y, z)

In [2]: eq1 = 3*x-y+z==185
        eq2 = 2*x+3*y-z==321
        eq3 = x+y+z==173
        solve([eq1, eq2, eq3], x, y, z)
Out[2]: [[x == 74, y == 68, z == 31]]
```

最后一步：



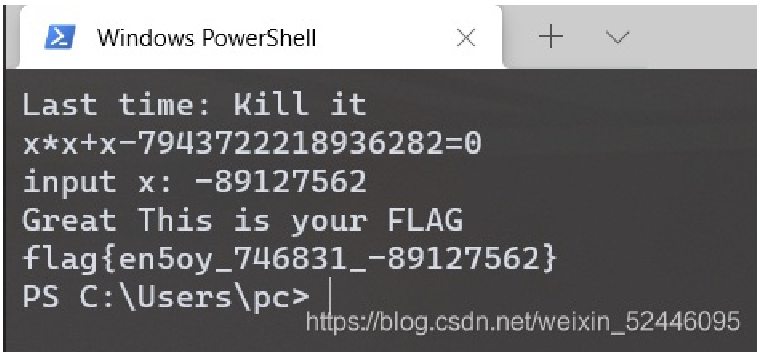
```
Last time: Kill it
x*x+x-7943722218936282=0
input x:
```

还是用 sagemath 求解



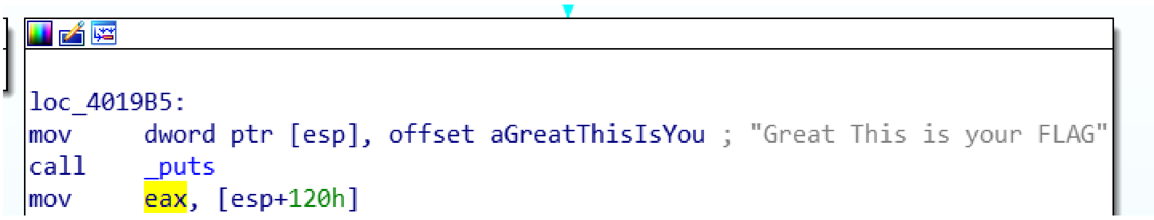
```
In [1]: var('x y z')
Out[1]: (x, y, z)

In [2]: eq1 = 3*x-y+z==185
        eq2 = 2*x+3*y-z==321
        eq3 = x+y+z==173
        solve([eq1, eq2, eq3], x, y, z)
Out[2]: [[x == 74, y == 68, z == 31]]

In [3]: solve([x^2+x-7943722218936282==0], x)
Out[3]: [x == 89127561, x == -89127562]
```

输入其中一个解即可

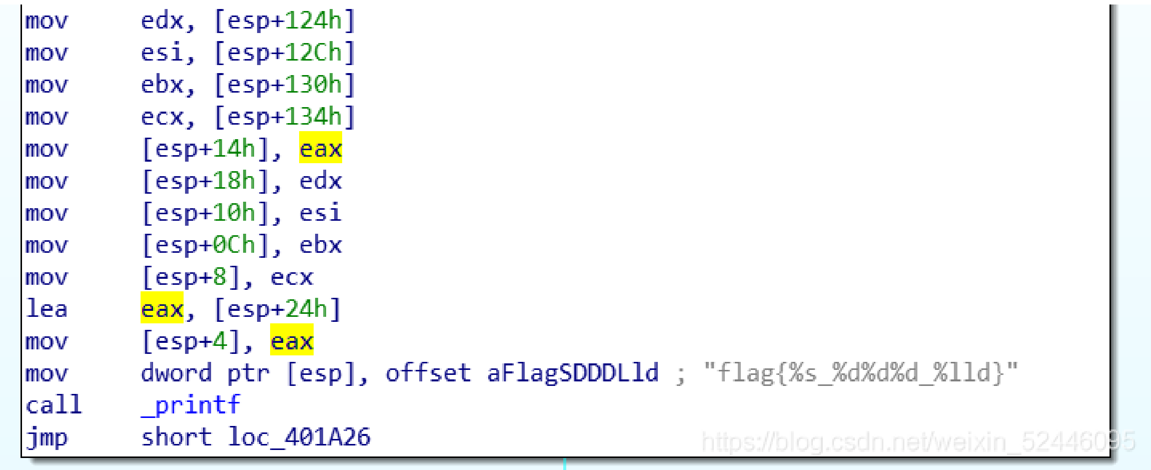


```
Last time: Kill it
x*x+x-7943722218936282=0
input x: -89127562
Great This is your FLAG
flag{en5oy_746831_-89127562}
PS C:\Users\pc>
```

当然你可以 re [doge]
把它丢尽 IDA 里面，可以得到程序的框架
这里用的的软件是 IDA Freeware 7.0
先看输出 flag 的部分：



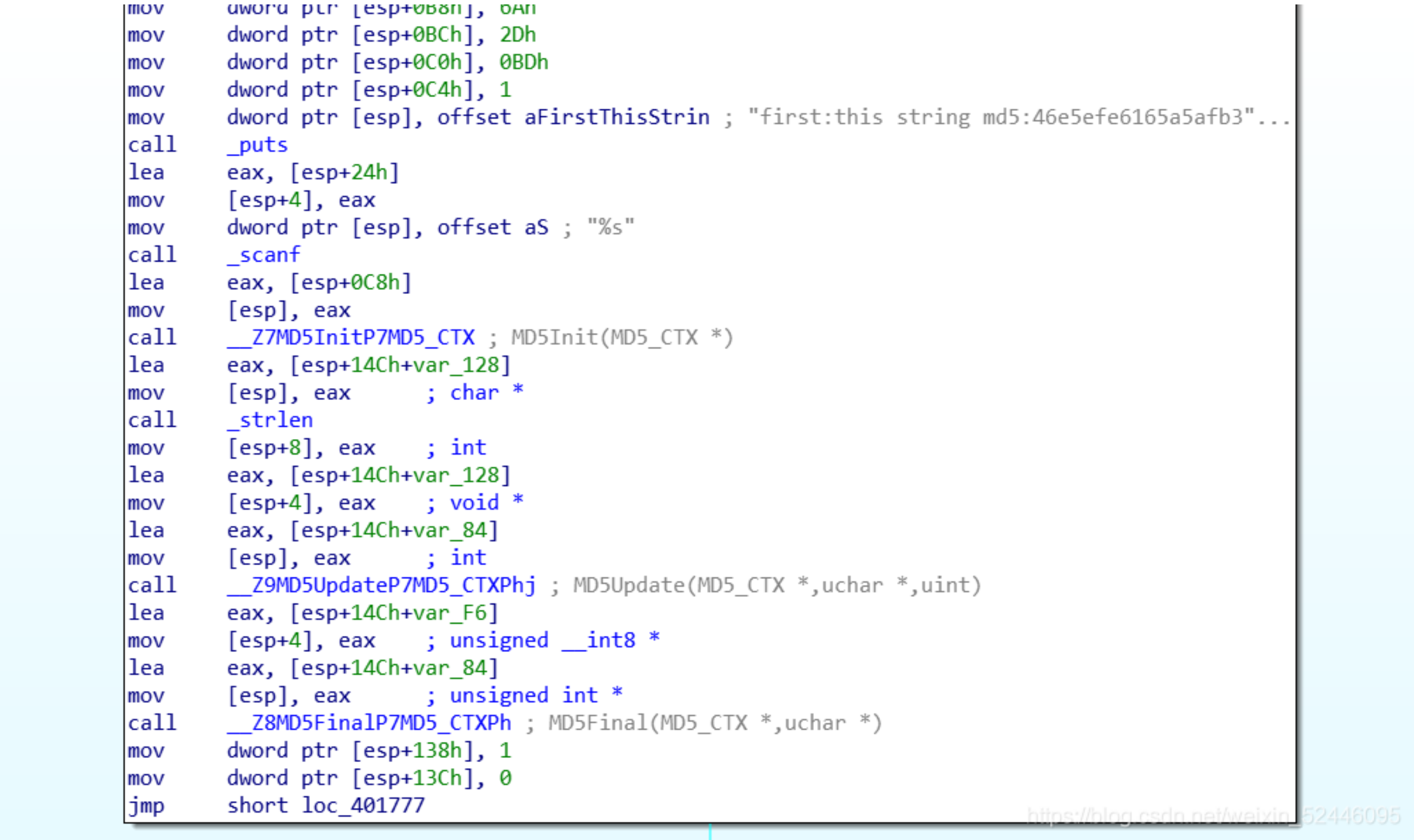
```
loc_4019B5:
mov     dword ptr [esp], offset aGreatThisIsYou ; "Great This is your FLAG"
call    _puts
mov     eax, [esp+120h]
```



```
mov     edx, [esp+124h]
mov     esi, [esp+12Ch]
mov     ebx, [esp+130h]
mov     ecx, [esp+134h]
mov     [esp+14h], eax
mov     [esp+18h], edx
mov     [esp+10h], esi
mov     [esp+0Ch], ebx
mov     [esp+8], ecx
lea     eax, [esp+24h]
mov     [esp+4], eax
mov     dword ptr [esp], offset aFlagSDDDLld ; "flag{%s_%d%d%d_%lld}"
call    _printf
jmp     short loc_401A26
```

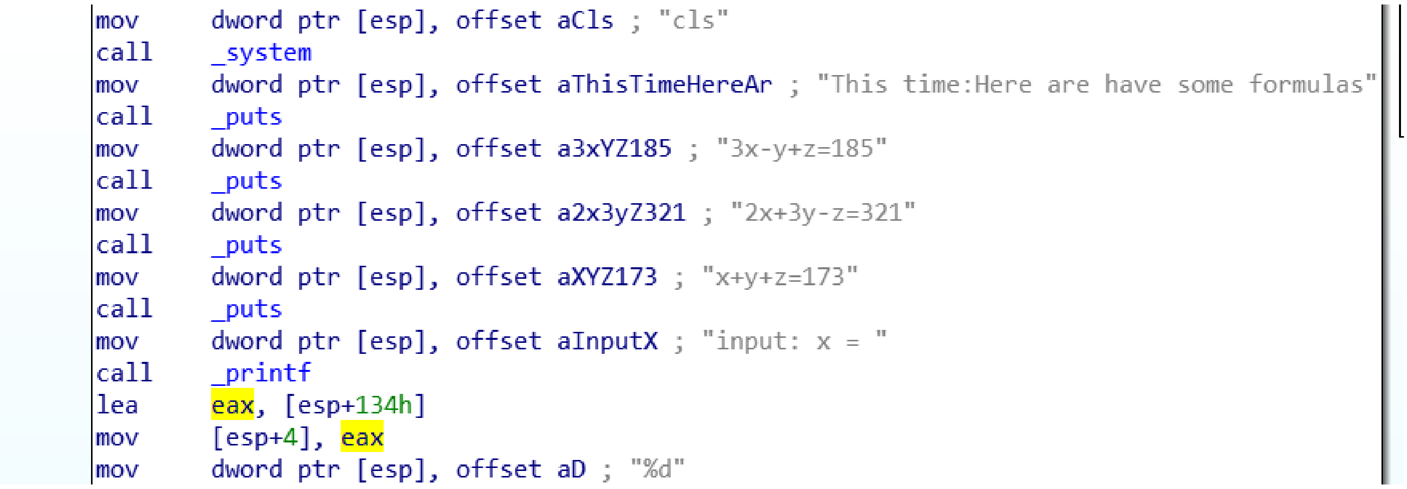
发现 flag 是由三部分组成

第一部分：

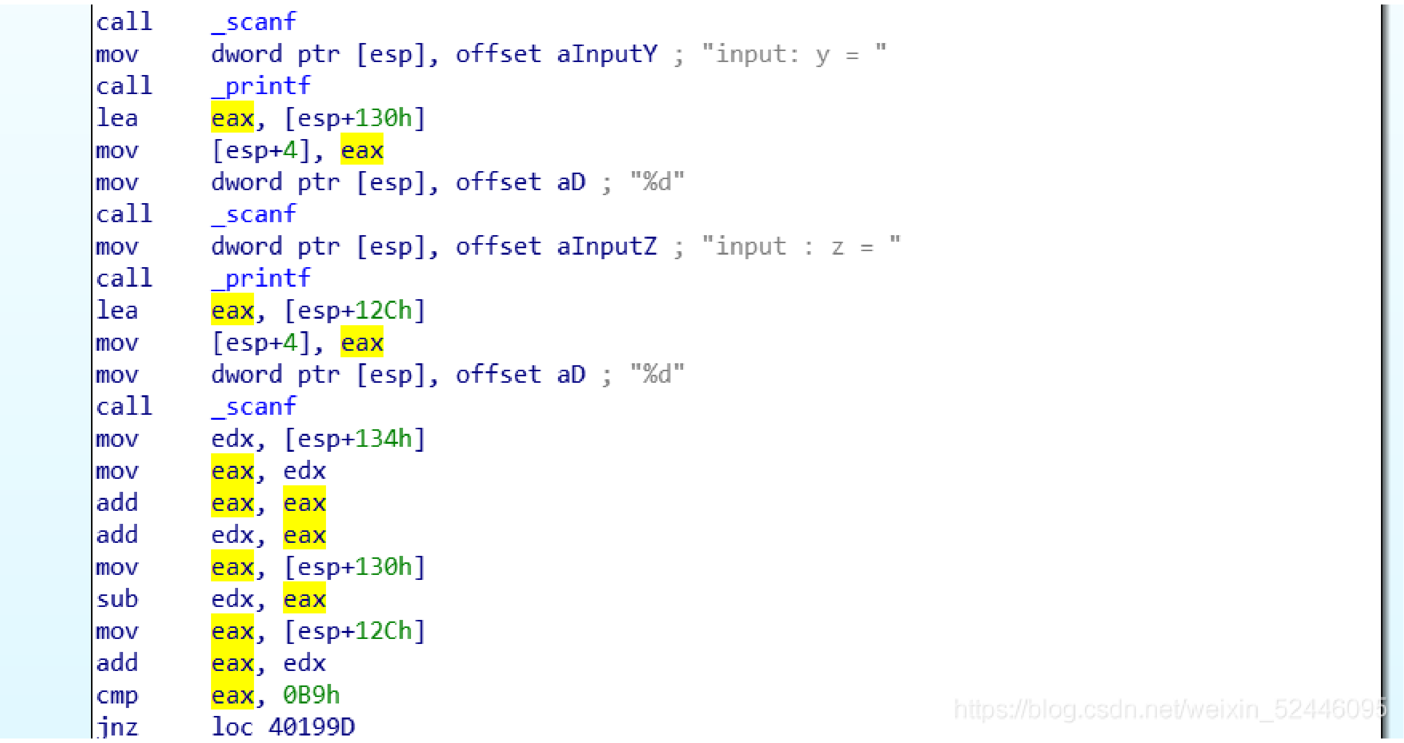


```
mov     dword ptr [esp+0B8h], 0Ah
mov     dword ptr [esp+0BCh], 2Dh
mov     dword ptr [esp+0C0h], 0BDh
mov     dword ptr [esp+0C4h], 1
mov     dword ptr [esp], offset aFirstThisStrin ; "first:this string md5:46e5efe6165a5afb3"...
call    _puts
lea     eax, [esp+24h]
mov     [esp+4], eax
mov     dword ptr [esp], offset aS ; "%s"
call    _scanf
lea     eax, [esp+0C8h]
mov     [esp], eax
call    __Z7MD5InitP7MD5_CTX ; MD5Init(MD5_CTX *)
lea     eax, [esp+14Ch+var_128]
mov     [esp], eax      ; char *
call    _strlen
mov     [esp+8], eax    ; int
lea     eax, [esp+14Ch+var_128]
mov     [esp+4], eax    ; void *
lea     eax, [esp+14Ch+var_84]
mov     [esp], eax      ; int
call    __Z9MD5UpdateP7MD5_CTXPhj ; MD5Update(MD5_CTX *,uchar *,uint)
lea     eax, [esp+14Ch+var_F6]
mov     [esp+4], eax    ; unsigned __int8 *
lea     eax, [esp+14Ch+var_84]
mov     [esp], eax      ; unsigned int *
call    __Z8MD5FinalP7MD5_CTXPh ; MD5Final(MD5_CTX *,uchar *)
mov     dword ptr [esp+138h], 1
mov     dword ptr [esp+13Ch], 0
jmp     short loc_401777
```

第二部分：

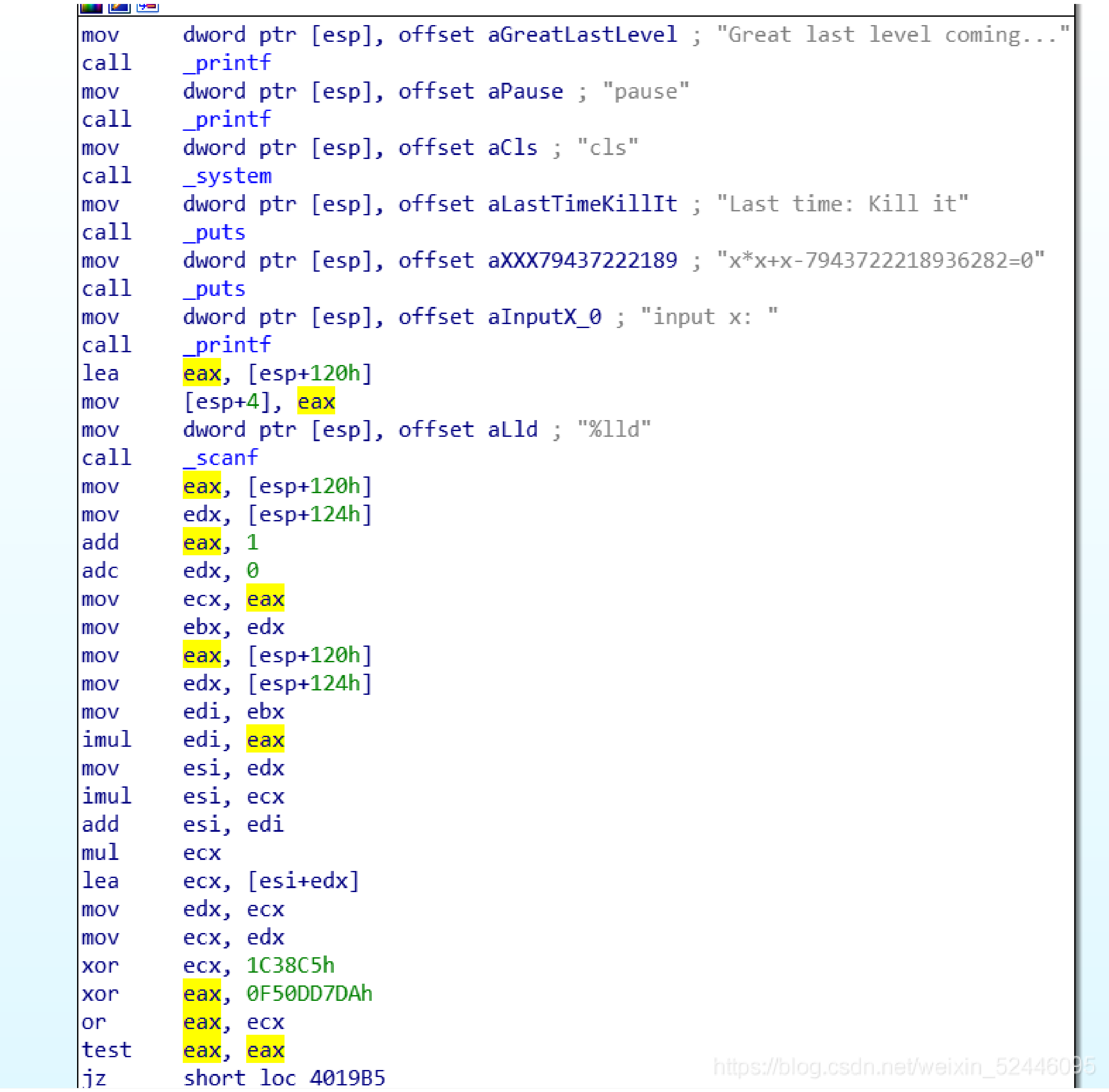


```
mov     dword ptr [esp], offset aCls ; "cls"
call    _system
mov     dword ptr [esp], offset aThisTimeHereAr ; "This time:Here are have some formulas"
call    _puts
mov     dword ptr [esp], offset a3xYZ185 ; "3x-y+z=185"
call    _puts
mov     dword ptr [esp], offset a2x3yZ321 ; "2x+3y-z=321"
call    _puts
mov     dword ptr [esp], offset aXYZ173 ; "x+y+z=173"
call    _puts
mov     dword ptr [esp], offset aInputX ; "input: x = "
call    _printf
lea     eax, [esp+134h]
mov     [esp+4], eax
mov     dword ptr [esp], offset aD ; "%d"
```



```
call    _scanf
mov     dword ptr [esp], offset aInputY ; "input: y = "
call    _printf
lea     eax, [esp+130h]
mov     [esp+4], eax
mov     dword ptr [esp], offset aD ; "%d"
call    _scanf
mov     dword ptr [esp], offset aInputZ ; "input : z = "
call    _printf
lea     eax, [esp+12Ch]
mov     [esp+4], eax
mov     dword ptr [esp], offset aD ; "%d"
call    _scanf
mov     edx, [esp+134h]
mov     eax, edx
add     eax, eax
add     edx, eax
mov     eax, [esp+130h]
sub     edx, eax
mov     eax, [esp+12Ch]
add     eax, edx
cmp     eax, 0B9h
jnz     loc_40199D
```

第三部分：



```
mov     dword ptr [esp], offset aGreatLastLevel ; "Great last level coming..."
call    _printf
mov     dword ptr [esp], offset aPause ; "pause"
call    _printf
mov     dword ptr [esp], offset aCls ; "cls"
call    _system
mov     dword ptr [esp], offset aLastTimeKillIt ; "Last time: Kill it"
call    _puts
mov     dword ptr [esp], offset aXXX79437222189 ; "x*x+x-7943722218936282=0"
call    _puts
mov     dword ptr [esp], offset aInputX_0 ; "input x: "
call    _printf
lea     eax, [esp+120h]
mov     [esp+4], eax
mov     dword ptr [esp], offset aLld ; "%lld"
call    _scanf
mov     eax, [esp+120h]
mov     edx, [esp+124h]
add     eax, 1
adc     edx, 0
mov     ecx, eax
mov     ebx, edx
mov     eax, [esp+120h]
mov     edx, [esp+124h]
mov     edi, ebx
imul    edi, eax
mov     esi, edx
imul    esi, ecx
add     esi, edi
mul     ecx
lea     ecx, [esi+edx]
mov     edx, ecx
mov     ecx, edx
xor     ecx, 1C38C5h
xor     eax, 0F50DD7DAh
or      eax, ecx
test    eax, eax
jz      short loc_4019B5
```

容易知道 flag 即为上面三部分答案拼接而成

B@se

附件内容如下：

```
密文：MyLkTaP3FaA7KOWjTmKkVjWjVzKjdeNvTnAjoH9iZOIvTeHbvD==
JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rs****kxyz012789+/

oh holy shit, something is missing...
```

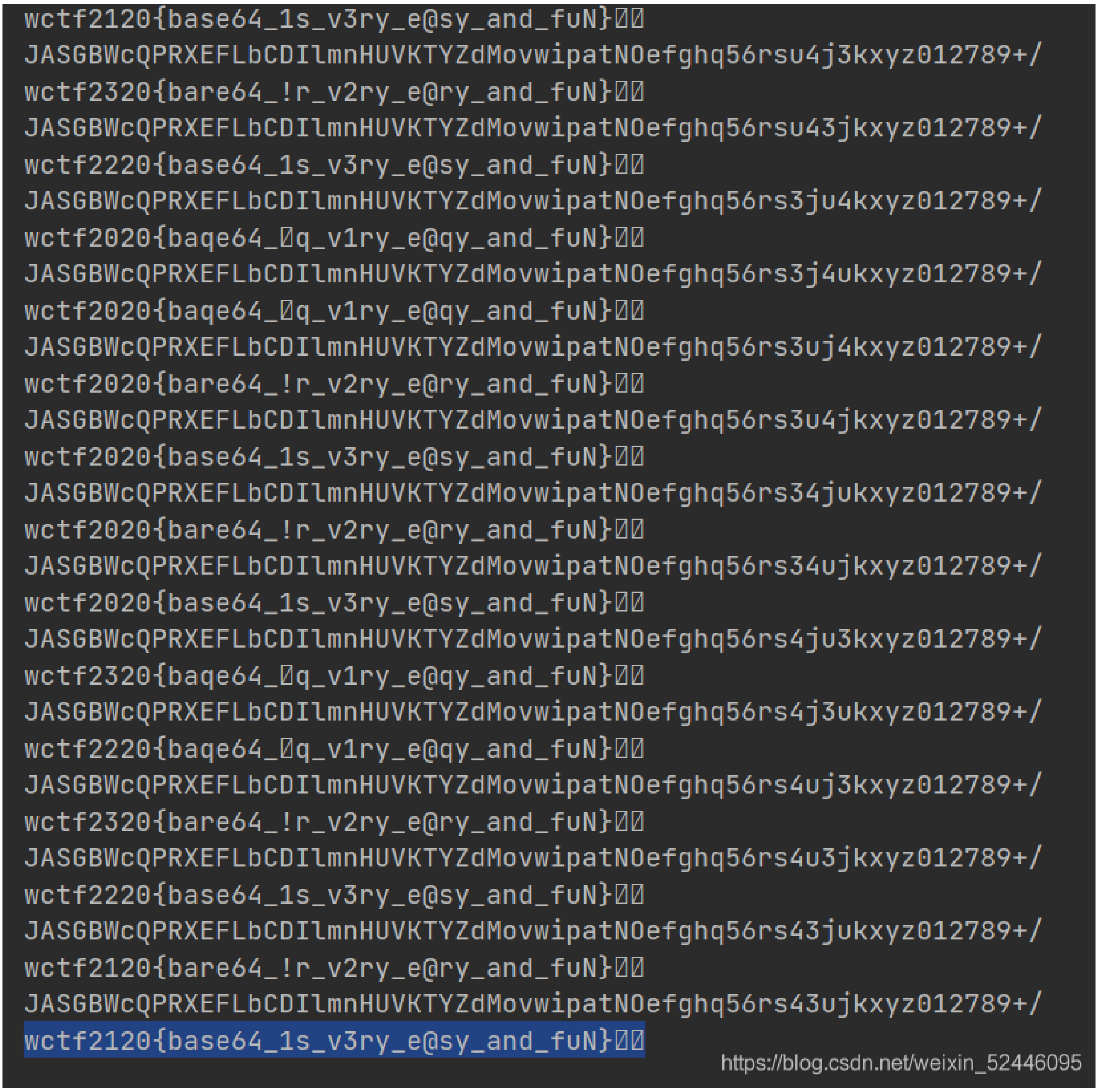
第一行是密文
第二行容易猜到是重新排列的 Base64 编码对照表
只是其中几个字符缺失了
编写代码：

```
def judge(key, start, end):
    s = ''
    for i in range(start, end+1):
        if not chr(i) in key:
            s += chr(i)
    return s
unknown = judge(key, ord('A'), ord('Z')) + judge(key, ord('a'), ord('z')) + judge(key, ord('0'), ord('9'))
unknown_list = list(unknown)
print(unknown_list)
```

可以得出缺失的字符为 [‘j’, ‘u’, ‘3’, ‘4’]
对其进行排列组合，替换对照表中缺失的字符
然后按照 Base64 的编码规则编写程序
代码如下：

```
import itertools

c = 'MyLkTaP3FaA7KOWjTmKkVjWjVzKjdeNvTnAjoH9iZOIvTeHbvD=='
key = 'JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rs****kxyz012789+/'

def decrypt(c, key):
    b = ''
    s = ''
    for i in range(len(c)):
        if c[i] == '=':
            b += '0'*6
        else:
            b += bin(list(key).index(c[i]))[2:].zfill(6)
    for i in range(0, len(b), 8):
        s += chr(int(b[i:i+8], 2))
    print(s)

def judge(key, start, end):
    s = ''
    for i in range(start, end+1):
        if not chr(i) in key:
            s += chr(i)
    return s
unknown = judge(key, ord('A'), ord('Z')) + judge(key, ord('a'), ord('z')) + judge(key, ord('0'), ord('9'))
unknown_list = list(unknown)
print(unknown_list)
combination = list(itertools.permutations(unknown_list,4))
for i in range(len(combination)):
    key_new = key.replace('****', ''.join(list(combination[i])))
    print(key_new)
    decrypt(c, key_new)
```

输出结果为：



```
wctf2120{base64_1s_v3ry_e@sy_and_fuN}⍰⍰
JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rsu4j3kxyz012789+/
wctf2320{bare64_!r_v2ry_e@ry_and_fuN}⍰⍰
JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rsu43jkxyz012789+/
wctf2220{base64_1s_v3ry_e@sy_and_fuN}⍰⍰
JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rs3ju4kxyz012789+/
wctf2020{baqe64_⍰q_v1ry_e@qy_and_fuN}⍰⍰
JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rs3j4ukxyz012789+/
wctf2020{baqe64_⍰q_v1ry_e@qy_and_fuN}⍰⍰
JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rs3uj4kxyz012789+/
wctf2020{bare64_!r_v2ry_e@ry_and_fuN}⍰⍰
JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rs3u4jkxyz012789+/
wctf2020{base64_1s_v3ry_e@sy_and_fuN}⍰⍰
JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rs34jukxyz012789+/
wctf2020{bare64_!r_v2ry_e@ry_and_fuN}⍰⍰
JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rs34ujkxyz012789+/
wctf2020{base64_1s_v3ry_e@sy_and_fuN}⍰⍰
JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rs4ju3kxyz012789+/
wctf2320{baqe64_⍰q_v1ry_e@qy_and_fuN}⍰⍰
JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rs4j3ukxyz012789+/
wctf2220{baqe64_⍰q_v1ry_e@qy_and_fuN}⍰⍰
JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rs4uj3kxyz012789+/
wctf2320{bare64_!r_v2ry_e@ry_and_fuN}⍰⍰
JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rs4u3jkxyz012789+/
wctf2220{base64_1s_v3ry_e@sy_and_fuN}⍰⍰
JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rs43jukxyz012789+/
wctf2120{bare64_!r_v2ry_e@ry_and_fuN}⍰⍰
JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rs43ujkxyz012789+/
wctf2120{base64_1s_v3ry_e@sy_and_fuN}⍰⍰
```

可以看到会有很多重复的内容
应该是由于有些字符在编码表中没有对应的字符，或者对应的字符在不同排列组合的编码表中的位置相同
结果为：wctf2120{base64_1s_v3ry_e@sy_and_fuN}

结语

希望继续坚持