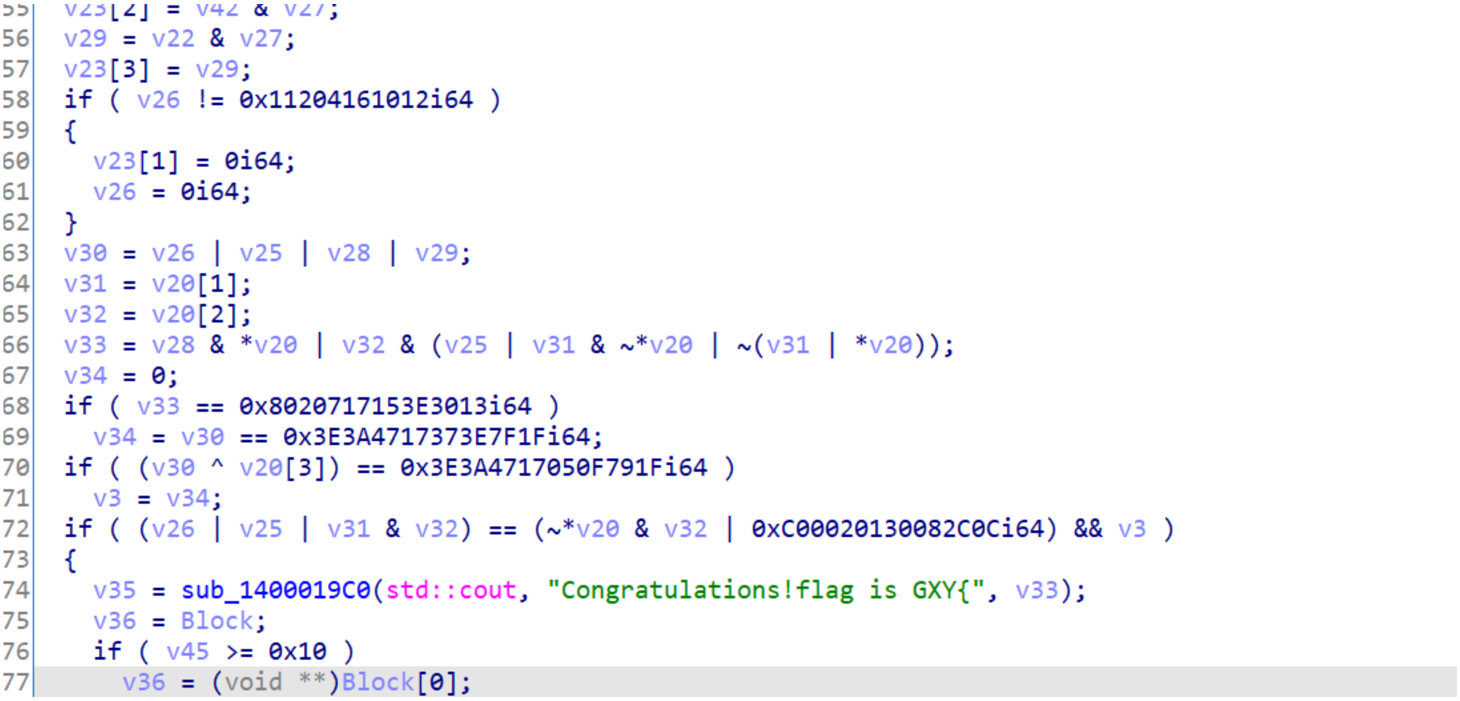
很屎的一道题

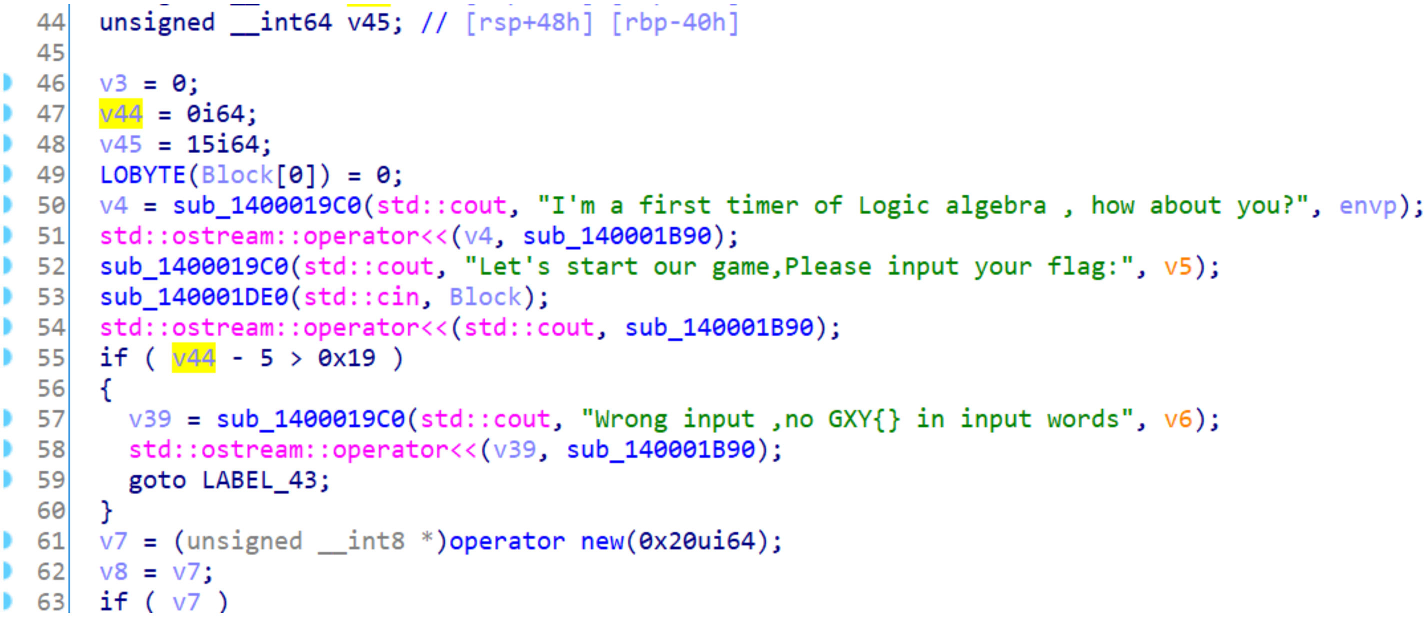
正常看看不出花样来



```
v23[2] = v42 & v27;
v29 = v22 & v27;
v23[3] = v29;
if ( v26 != 0x11204161012i64 )
{
  v23[1] = 0i64;
  v26 = 0i64;
}
v30 = v26 | v25 | v28 | v29;
v31 = v20[1];
v32 = v20[2];
v33 = v28 & *v20 | v32 & (v25 | v31 & ~*v20 | ~(v31 | *v20));
v34 = 0;
if ( v33 == 0x8020717153E3013i64 )
  v34 = v30 == 0x3E3A4717373E7F1Fi64;
if ( (v30 ^ v20[3]) == 0x3E3A4717050F791Fi64 )
  v3 = v34;
if ( (v26 | v25 | v31 & v32) == (~*v20 & v32 | 0xC00020130082C0Ci64) && v3 )
{
  v35 = sub_1400019C0(std::cout, "Congratulations!flag is GXY{", v33);
  v36 = Block;
  if ( v45 >= 0x10 )
    v36 = (void **)Block[0];
```

要想过关，大概率要满足拿4个大整数（猜的）

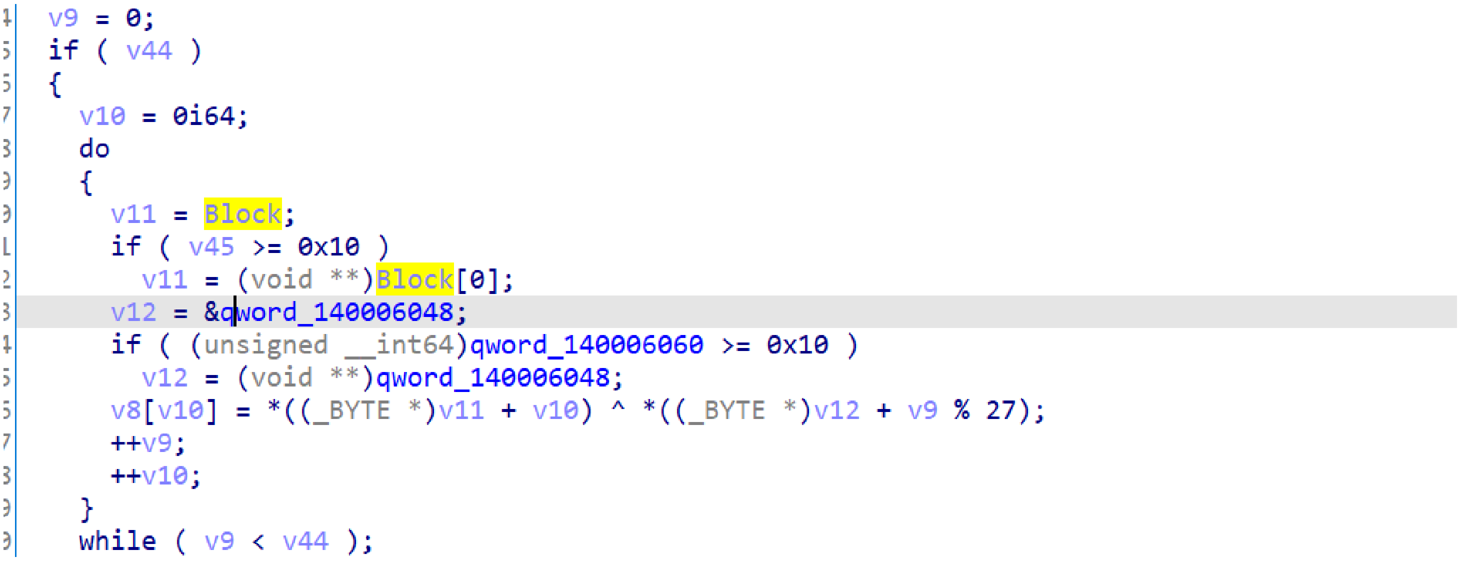
往最前面看



```
unsigned __int64 v45; // [rsp+48h] [rbp-40h]

v3 = 0;
v44 = 0i64;
v45 = 15i64;
LOBYTE(Block[0]) = 0;
v4 = sub_1400019C0(std::cout, "I'm a first timer of Logic algebra , how about you?", envp);
std::ostream::operator<<(v4, sub_140001B90);
sub_1400019C0(std::cout, "Let's start our game,Please input your flag:", v5);
sub_140001DE0(std::cin, Block);
std::ostream::operator<<(std::cout, sub_140001B90);
if ( v44 - 5 > 0x19 )
{
  v39 = sub_1400019C0(std::cout, "Wrong input ,no GXY{} in input words", v6);
  std::ostream::operator<<(v39, sub_140001B90);
  goto LABEL_43;
}
v7 = (unsigned __int8 *)operator new(0x20ui64);
v8 = v7;
if ( v7 )
```

要你输入字符



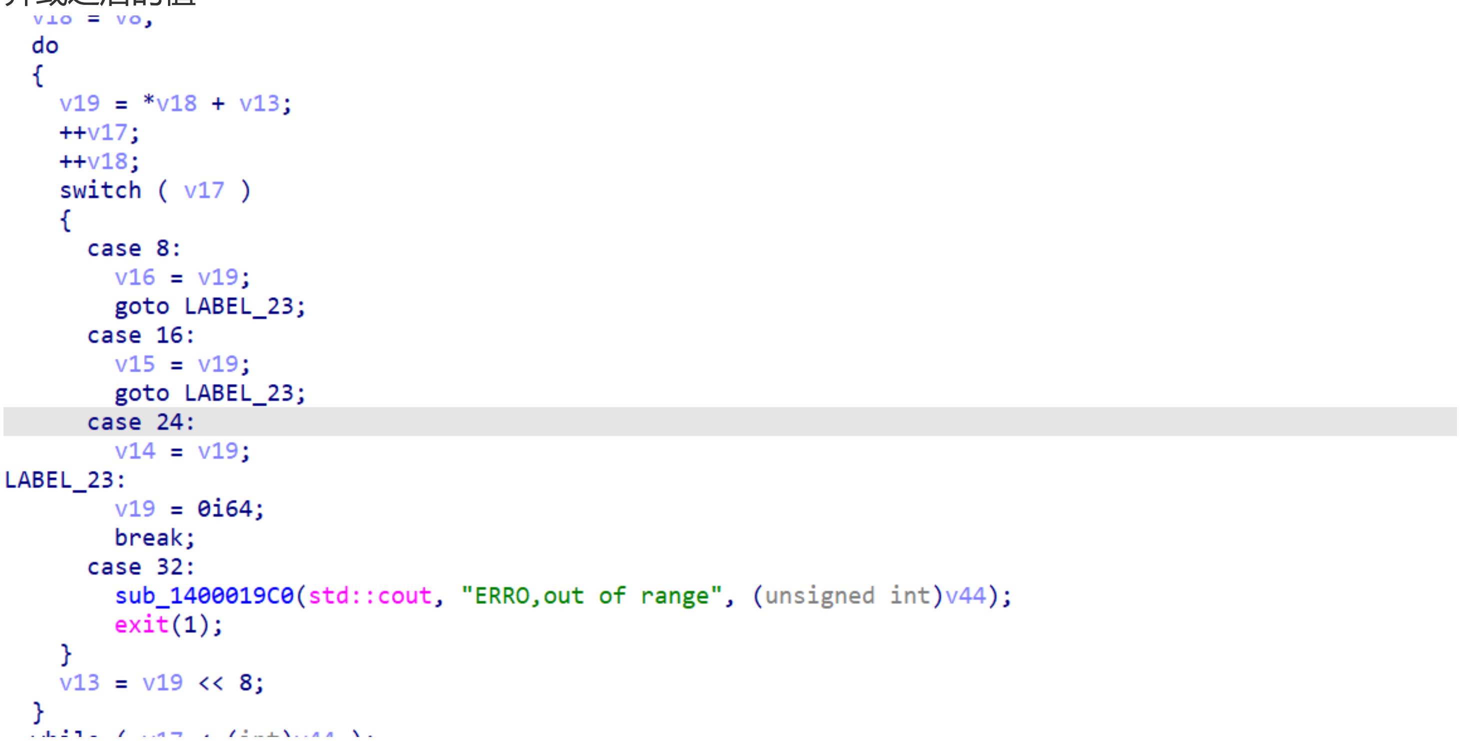
```
v9 = 0;
if ( v44 )
{
  v10 = 0i64;
  do
  {
    v11 = Block;
    if ( v45 >= 0x10 )
      v11 = (void **)Block[0];
    v12 = &qword_140006048;
    if ( (unsigned __int64)qword_140006060 >= 0x10 )
      v12 = (void **)qword_140006048;
    v8[v10] = *((_BYTE *)v11 + v10) ^ *((_BYTE *)v12 + v9 % 27);
    ++v9;
    ++v10;
  }
  while ( v9 < v44 );
```

这是对输入的字符·进行异或操作

正常看我觉得是看不出来的

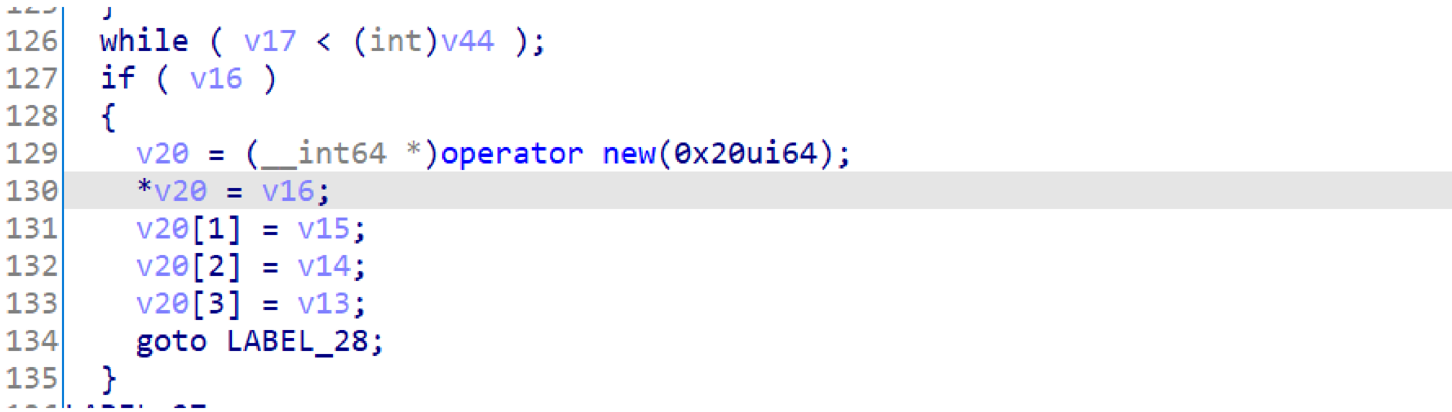
我动调了才知道是跟i_will_check_is_debug_or_not进行异或

异或之后的值



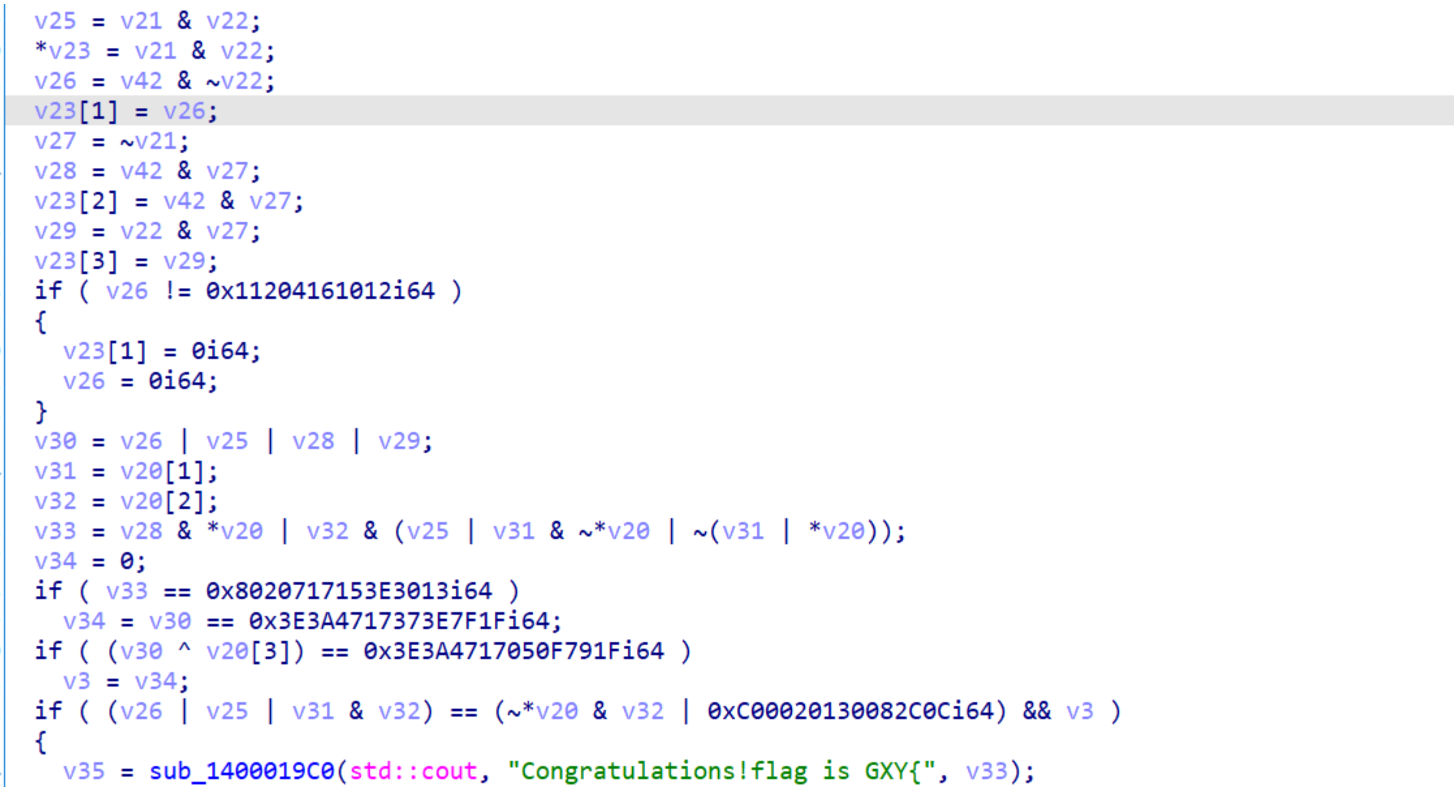
```
  v18 = v8;
  do
  {
    v19 = *v18 + v13;
    ++v17;
    ++v18;
    switch ( v17 )
    {
      case 8:
        v16 = v19;
        goto LABEL_23;
      case 16:
        v15 = v19;
        goto LABEL_23;
      case 24:
        v14 = v19;
LABEL_23:
        v19 = 0i64;
        break;
      case 32:
        sub_1400019C0(std::cout, "ERRO,out of range", (unsigned int)v44);
        exit(1);
    }
    v13 = v19 << 8;
  }
  while ( v17 < (int)v44 );
```

每8字节存放在一个寄存器中



```
}
while ( v17 < (int)v44 );
if ( v16 )
{
  v20 = (__int64 *)operator new(0x20ui64);
  *v20 = v16;
  v20[1] = v15;
  v20[2] = v14;
  v20[3] = v13;
  goto LABEL_28;
}
```

再把寄存器中的值丢到v20的数组中



```
v25 = v21 & v22;
*v23 = v21 & v22;
v26 = v42 & ~v22;
v23[1] = v26;
v27 = ~v21;
v28 = v42 & v27;
v23[2] = v42 & v27;
v29 = v22 & v27;
v23[3] = v29;
if ( v26 != 0x11204161012i64 )
{
  v23[1] = 0i64;
  v26 = 0i64;
}
v30 = v26 | v25 | v28 | v29;
v31 = v20[1];
v32 = v20[2];
v33 = v28 & *v20 | v32 & (v25 | v31 & ~*v20 | ~(v31 | *v20));
v34 = 0;
if ( v33 == 0x8020717153E3013i64 )
  v34 = v30 == 0x3E3A4717373E7F1Fi64;
if ( (v30 ^ v20[3]) == 0x3E3A4717050F791Fi64 )
  v3 = v34;
if ( (v26 | v25 | v31 & v32) == (~*v20 & v32 | 0xC00020130082C0Ci64) && v3 )
{
  v35 = sub_1400019C0(std::cout, "Congratulations!flag is GXY{", v33);
```

然后就是所谓的代数化简了
这题目给的信息，我还没认真看，看了wp才知道是代数化简，5个等式可以求4个值，刚刚好和输入处理后的4个寄存器对应。

知道算法后可以解密
解密脚本如下

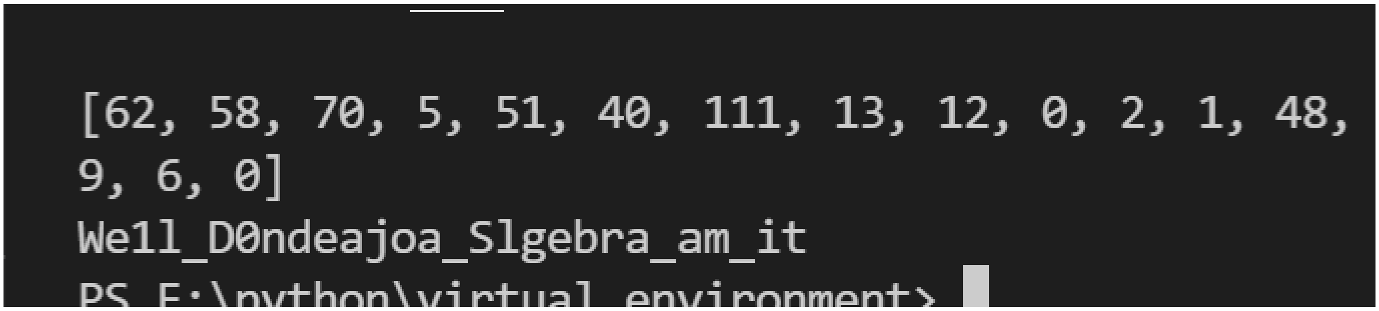
```
from z3 import *

s = Solver()
x,y,z,w = BitVecs('x y z w',64)
s.add((z&(~x))==0x11204161012)
s.add(((z&(y))&x|z&(x&y|y&(x)|~(y|x)))==0x8020717153E3013)
s.add(((z&(x))|(x&y)|(z&y)|(x&~y))==0x3E3A4717373E7F1F)
```

```
s.add((((z&(x))|(x&y)|(z&y)|(x&~y))^w)==0x3E3A4717050F791F)
s.add(((z&(x)|(x&y)|y&z))==(x&z|0xC00020130082C0C))

s.check()
model = s.model()
print(model)
check = ''
check += hex(model[x].as_long())[2:].rjust(16,'0')
check += hex(model[y].as_long())[2:].rjust(16,'0')
check += hex(model[z].as_long())[2:].rjust(16,'0')
check += hex(model[w].as_long())[2:].rjust(8,'0')

enc = []
for i in range(28):
a = check[2i:2i+2]
enc.append(int(a,16))
print(enc)
key = 'i_will_check_is_debug_or_not'
flag = []
for i in range(28):
print(chr(ord(key[i])^enc[i]),end='')
```



```
[62, 58, 70, 5, 51, 40, 111, 13, 12, 0, 2, 1, 48,
9, 6, 0]
We1l_D0ndeajoa_Slgebra_am_it
PS E:\python\virtual_environment>
```

然后看了wp，发现有一部分题目有给

最终flag

We1l_D0ne!P0or_algebra_am_i