```
from z3 import *
final=[0x0001C633, 0x0001DF94, 0x00020EBF,
0x0002BA40, 0x0001E884, 0x000260D1, 0x0001F9B1,
0x0001EA1A, 0x0001EEAA, 0x0001DFB2, 0x0001C1D0,
0x0001EEF2, 0x000216E1, 0x0002BE00, 0x0001FB5E,
0x00025D74, 0x0001F000, 0x000202D6, 0x00020002,
0x0001DDFE, 0x0001C017, 0x0001F08C, 0x000227F6,
0x0002C7BA, 0x000201AE, 0x00027FBF, 0x00020E21,
0x0001FF5C, 0x0001FD62, 0x0001E948, 0x0001BE6E,
0x0001F4D7, 0x00022C8D, 0x0002C353, 0x0001F8DB,
0x00026E1D, 0x0001FF61, 0x0001EA0F, 0x0001F0D6,
0x0001EDA8, 0x0001AD7D, 0x00018218, 0x0001CCD4,
0x000239B6, 0x0001AC4C, 0x00020D7C, 0x0001D967,
0x0001A4F4, 0x0001CAD8, 0x000196AE, 0x0001831B,
0x00017E45, 0x0001D0CF, 0x00023EDF, 0x000181AE,
0x00021760, 0x0001D3B4, 0x000175D6, 0x00017D3A,
0x0001994F, 0x0001189D, 0x00014CCF, 0x0001568E,
0x00017EEB, 0x0001327E, 0x00016A45, 0x00012921,
0x00011FF0, 0x00013643, 0x00011729, 0x00015191,
0x00017D17, 0x00017262, 0x0001A863, 0x00017010,
0x00017B10, 0x00014F9C, 0x000143E8, 0x00015E9B,
0x0001242C, 0x0000F68C, 0x0001192A, 0x000150AD,
0x0001B1A0, 0x00014C60, 0x000182AB, 0x00013F4B,
0x000141A6, 0x00015AA3, 0x000135C9, 0x0001D86F,
0x0001E8FA, 0x0002158D, 0x0002BDAC, 0x00020E4F,
0x00027EE6, 0x000213B9, 0x00020E86, 0x000211FF,
0x0001E1EF]
second=[0x000000FE, 0x0000000B, 0x0000001D,
0x000000F6, 0x00000083, 0x000000FF, 0x000000E0,
0x000000B8, 0x000000DD, 0x000000B0, 0x000000C5,
0x000000DE, 0x000000F6, 0x00000014, 0x0000009F,
0x000000DD, 0x000000D9, 0x00000007, 0x0000002D,
0x0000006B, 0x00000019, 0x000000CA, 0x00000073,
0x000000FD, 0x00000087, 0x00000072, 0x00000024,
0x00000004, 0x00000049, 0x0000007E, 0x000000A9,
0x000000CE, 0x00000091, 0x000000BE, 0x00000041,
0x00000018, 0x00000060, 0x0000003F, 0x0000002B,
0x00000063, 0x0000001C, 0x000000D2, 0x00000090,
0x000000E9, 0x0000008E, 0x000000BA, 0x0000001E,
0x000000F3, 0x00000041, 0x000000AD, 0x0000002C,
0x00000003, 0x00000069, 0x000000DA, 0x00000010,
```

```
0x000000FD, 0x000000FD, 0x000000E7, 0x00000006,
0x00000036, 0x000000D6, 0x00000002, 0x00000059,
0x00000018, 0x000000CC, 0x00000050, 0x00000087,
0x000000AF, 0x000000FB, 0x00000018, 0x00000044,
0x0000007F, 0x000000AD, 0x000000F8, 0x0000002C,
0x00000067, 0x0000001D, 0x00000022, 0x00000084,
0x000000AC, 0x0000000E, 0x00000023, 0x000000DC,
0x000000E6, 0x000000BB, 0x000000D2, 0x000000B8,
0x0000004A, 0x000000BC, 0x000000DE, 0x00000050,
0x0000009C, 0x0000001C, 0x0000001E, 0x00000086,
0x0000003A, 0x0000002D, 0x000000DD, 0x000000C3,
0x00000003]
input=[0x00000074, 0x00000068, 0x00000065,
0x00000072, 0x00000065, 0x0000005F, 0x00000061,
0x00000072, 0x00000065, 0x0000005F, 0x00000061,
0x0000005F, 0x0000006C, 0x0000006F, 0x00000074,
0x0000005F, 0x00000075, 0x00000073, 0x00000065,
0x0000006C, 0x00000065, 0x00000073, 0x00000073,
0x0000005F, 0x00000069, 0x0000006E, 0x00000066,
0x0000006F, 0x00000072, 0x0000006D, 0x00000061,
0x00000074, 0x00000069, 0x0000006F, 0x0000006E,
0x0000005F, 0x00000062, 0x00000075, 0x00000074,
0x0000005F, 0x0000006F, 0x00000068, 0x0000002E,
0x0000006F, 0x00000030, 0x0000004F, 0x0000005F,
0x00000031, 0x00000032, 0x00000033, 0x00000034,
0x00000035, 0x00000036, 0x00000037, 0x00000038,
0x00000039, 0x00000031, 0x00000032, 0x00000033,
0x00000034, 0x00000035, 0x00000036, 0x00000037,
0x00000038, 0x00000039, 0x00000031, 0x00000032,
0x00000033, 0x00000034, 0x00000035, 0x00000036,
0x00000037, 0x00000038, 0x00000039, 0x00000031,
0x00000032, 0x00000033, 0x00000034, 0x00000035,
0x00000036, 0x00000037, 0x00000038, 0x00000039,
0x00000031, 0x00000032, 0x00000033, 0x00000034,
0x00000035, 0x00000036, 0x0000005F, 0x00000079,
0x0000006F, 0x00000075, 0x0000005F, 0x00000067,
0x00000065, 0x00000074, 0x0000005F, 0x00000069,
0x00000074]
Str=[BitVec('flag%d'%tmp,8) for tmp in range(42)]
s=Solver()
# 先处理前三个的问题
```

```
for count in range(10):

s.add(input[40]*second[count+0]+input[41]*second[count
+10]+input[42]*second[count+20]+input[43]*second[count
+30]+input[44]*second[count+40]+input[45]*second[count
+50]+input[46]*second[count+60]+Str[0]*second[count+7
0]+Str[1]*second[count+80]+Str[2]*second[count+90]==fi
nal[4*10+count])
# 再处理中间的三十的玩意
for count1 in range(10):

s.add(Str[3]*second[count1+0]+Str[4]*second[count1+10]
+Str[5]*second[count1+20]+Str[6]*second[count1+30]+Str
[7]*second[count1+40]+Str[8]*second[count1+50]+Str[9]*
second[count1+60]+Str[10]*second[count1+70]+Str[11]*s
econd[count1+80]+Str[12]*second[count1+90]==final[5*1
0+count1])
for count2 in range(10):

s.add(Str[13]*second[count2+0]+Str[14]*second[count2+1
0]+Str[15]*second[count2+20]+Str[16]*second[count2+30
]
+Str[17]*second[count2+40]+Str[18]*second[count2+50]+
Str[19]*second[count2+60]+Str[20]*second[count2+70]+S
tr[21]*second[count2+80]+Str[22]*second[count2+90]==fi
nal[6*10+count2])
for count3 in range(10):

s.add(Str[23]*second[count3+0]+Str[24]*second[count3+1
0]+Str[25]*second[count3+20]+Str[26]*second[count3+30
]
+Str[27]*second[count3+40]+Str[28]*second[count3+50]+
Str[29]*second[count3+60]+Str[30]*second[count3+70]+S
tr[31]*second[count3+80]+Str[32]*second[count3+90]==fi
nal[7*10+count3])
# 继续处理后9个
for count4 in range(10):

s.add(Str[33]*second[count4+0]+Str[34]*second[count4+1
0]+Str[35]*second[count4+20]+Str[36]*second[count4+30
]
```

```
+Str[37]*second[count4+40]+Str[38]*second[count4+50]+
Str[39]*second[count4+60]+Str[40]*second[count4+70]+S
tr[41]*second[count4+80]+input[89]*second[count4+90]==
final[8*10+count4])

if s.check():
    a=s.model()
    print(a)
    flag=""
    for i in range(42):
        flag+=chr(a[Str[i]].as_long())
    print(flag)
```