babymaze

就是个最基本的迷宫，方向健换成了 abcd 防止不分析也能秒，用 pyinstaller 打包成 exe，按流程解包回去就行，但这题凑巧有 pycdc 不支持的字节码指令。

可以硬读字节码，可以把字节码放进 chatgpt 分析，也可以把 pycdc 的源码改一改把不支持的指令硬塞进去。

八色三月七

变种 base64，把码表换成了一二三四五六七八，就是利用 base64 的原理切割成 6bit 然后对 8 进行除，取余的操作，手算回来就行。

easymath

程序流程是取当前系统时间为随机数 seed 值，生成 16 个随机数作为 key 值对输入进行 aes128ecb 模式加密，再进行换表 base64 编码得到最终值。有个简单的 IsDebuggerPresent 反调试，这题动态调试也没啥用，忽略或 patch 即可。

因此解题的话先换表 base64 解码，然后进行时间戳的爆破，由于有 aes 加密的存在，因此爆破不是很快，而 Unix 时间戳是从 1970 年 1 月 1 日

（UTC/GMT 的午夜）开始所经过的秒数，因此不用从 0 爆，那时出题人没太生出来。

解题脚本：

```
from Cryptodome.Cipher import AES
def rand():
    global seed
    seed=(seed*214013+2531011)%2**64
    return (seed>>16)&0x7fff
encflag=[0xeb,0x6c,0xfc,0x8a,0x91,0x8e,0x0f,0xa5,
0xec,0x68,0x15,0x3b,0x09,0x5c,0x1c,0x38,
0xca,0x37,0x84,0x21,0x09,0x81,0x1e,0xf4,
0x7f,0x96,0x19,0xc4,0xf3,0xef,0x65,0x1f,
0x28,0x5f,0x70,0x1f,0x6c,0xc6,0x48,0xb2,
0xdb,0x71,0x87,0x16,0xeb,0x61,0x70,0x8f]
for seed in range(1706000000,1708000000):
    key=[0]*16
    for i in range(16):
        key[i]=rand()&0xff
    aes = AES.new(bytes(key),AES.MODE_ECB)
    flag = aes.decrypt(bytes(encflag))
    if b'SBCTF' in flag:
        print(flag)
```

rand 函数就是 c 语言 rand 函数的 python 版。