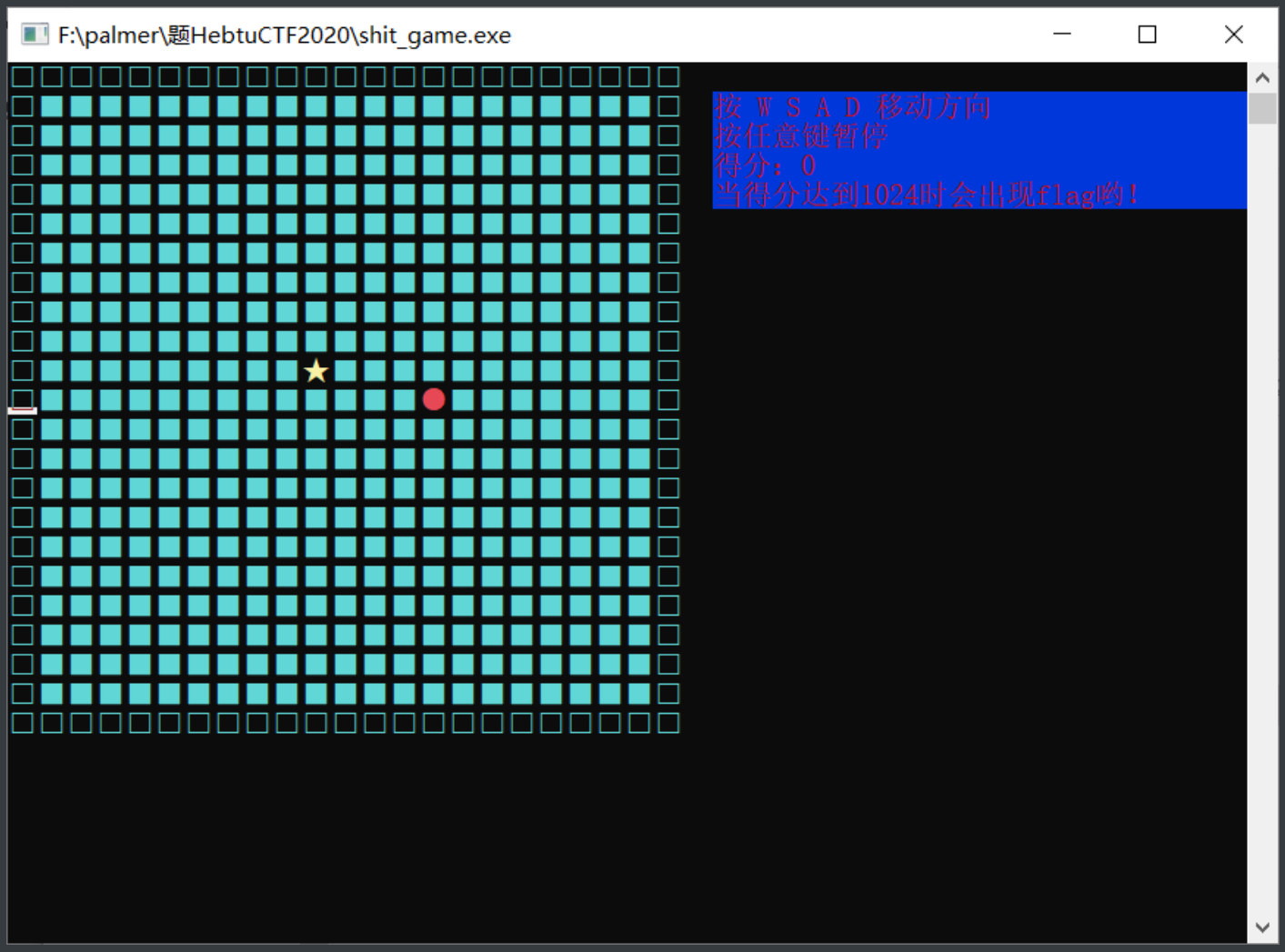
打开程序，是一个贪吃蛇游戏，写着得分达到1024时会出现flag

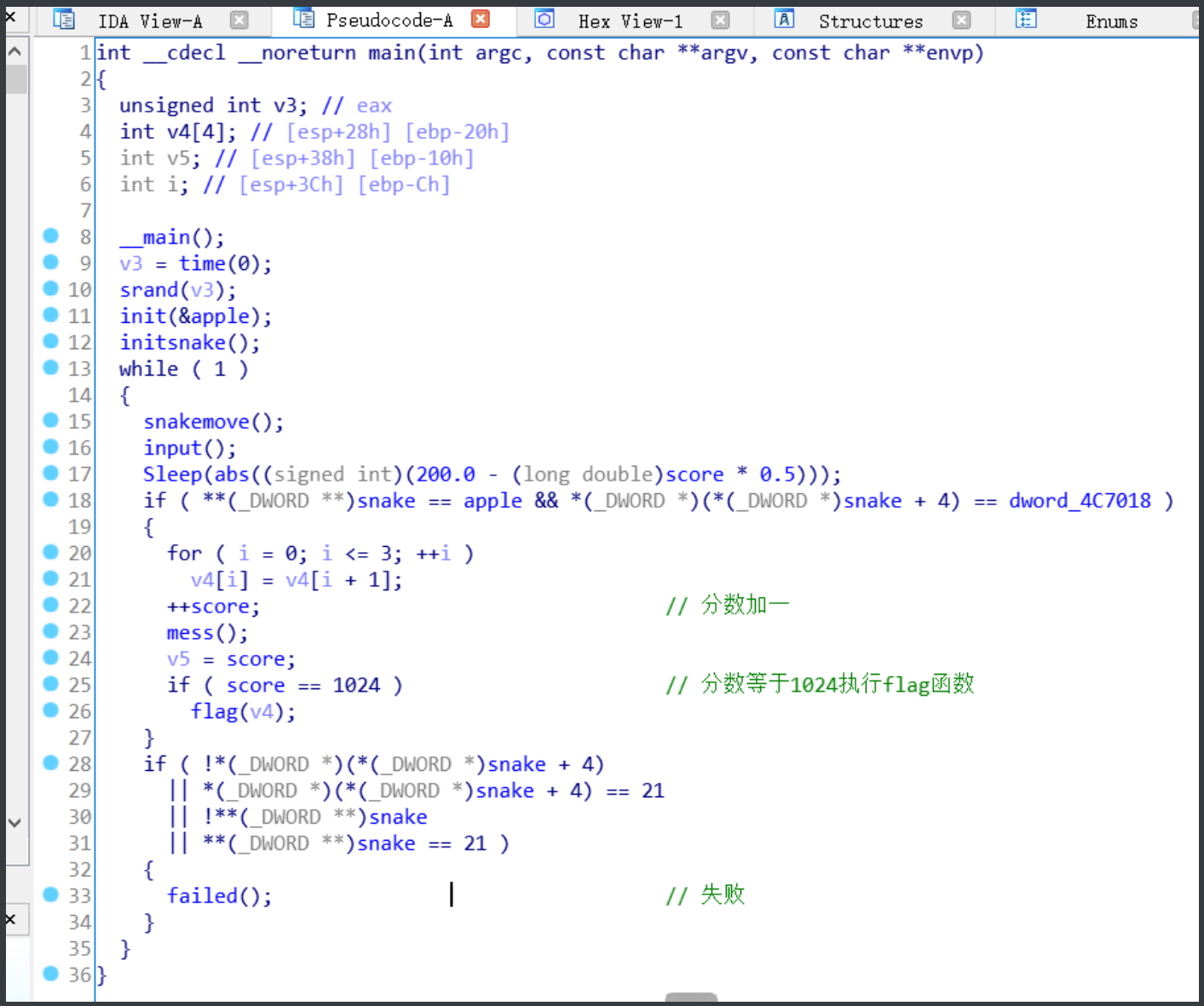


解法1：玩到1024得分flag自动出现

对于单身二十多年的手速来说，这个不难吧

ida分析

ida载入，打开main函数，F5进行反编译（也可以搜索字符串来定位




```
1  int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
2  {
3    unsigned int v3; // eax
4    int v4[4]; // [esp+28h] [ebp-20h]
5    int v5; // [esp+38h] [ebp-10h]
6    int i; // [esp+3Ch] [ebp-Ch]
7
8    __main();
9    v3 = time(0);
10   srand(v3);
11   init(&apple);
12   initsnake();
13   while ( 1 )
14   {
15     snakemove();
16     input();
17     Sleep(abs((signed int)(200.0 - (long double)score * 0.5)));
18     if ( **(_DWORD **)snake == apple && *(_DWORD *)(*(_DWORD *)snake + 4) == dword_4C7018 )
19     {
20       for ( i = 0; i <= 3; ++i )
21         v4[i] = v4[i + 1];
22       ++score;                                  // 分数加一
23       mess();
24       v5 = score;
25       if ( score == 1024 )                      // 分数等于1024执行flag函数
26         flag(v4);
27     }
28     if ( !*(_DWORD *)(*(_DWORD *)snake + 4)
29       || *(_DWORD *)(*(_DWORD *)snake + 4) == 21
30       || !**(_DWORD **)snake
31       || **(_DWORD **)snake == 21 )
32     {
33       failed();                                 // 失败
34     }
35   }
36 }
```

找到判断条件及进行的操作

第一个if循环应该是判断分数是否加1的（蛇是否吃到苹果

然后如果满足判断条件就将分数加1，赋值给局部变量v5，并且判断是否等于1024，是否执行flag函数，并且flag函数使用了一个int数组v4

先进flag函数内部查看

IDA View-A | Pseudocode-A | Hex View-1 | Structures

```
1  int __cdecl flag(int *a1)
2  {
3    int result; // eax
4    signed int v2; // [esp+14h] [ebp-14h]
5    signed int i; // [esp+18h] [ebp-10h]
6    int v4; // [esp+1Ch] [ebp-Ch]
7
8    v2 = strlen(_data_start__);
9    for ( i = 0; i < v2; ++i )
10   {
11     if ( _data_start__[i] <= 96 || _data_start__[i] > 122 )
12     {
13       if ( _data_start__[i] <= 64 || _data_start__[i] > 90 )
14         LOBYTE(v4) = _data_start__[i];
15       else
16         v4 = (_data_start__[i] - 65 + a1[i % 5]) % 26 + 97;
17     }
18     else
19     {
20       v4 = (_data_start__[i] - 97 + a1[i % 5]) % 26 + 65;
21     }
22     _data_start__[i] = v4;
23   }
24   gotoxy(24, 5);
25   color(20);
26   if ( _data_start__[0] != 'H'
27     || _data_start__[1] != 'E'
28     || _data_start__[2] != 'C'
29     || _data_start__[3] != 'T'
30     || _data_start__[4] != 'F' )
31   {
32     result = printf("No! you're cheating!");
33   }
34   else
35   {
36     result = printf("You win! The flag is %s ", _data_start__);
37   }
38   return result;
```

发现他这里进行了一堆变换，然后判断前5位是不是等于 HECTF

如果是的话就输出flag

上面的一堆变换涉及到了参数a1，用的是a1[i%5]，不难猜出a1有5位

返回上层函数查看a1参数也就是v4

IDA View-A | Pseudocode-A | Hex View-1 | Structures

```
1  int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
2  {
3    unsigned int v3; // eax
4    int v4[4]; // [esp+28h] [ebp-20h]
5    int v5; // [esp+38h] [ebp-10h]
6    int i; // [esp+3Ch] [ebp-Ch]
7
8    __main();
9    v3 = time(0);
10   srand(v3);
11   init(&apple);
12   initsnake();
13   while ( 1 )
14   {
15     snakemove();
16     input();
17     Sleep(abs((signed int)(200.0 - (long double)score * 0.5)));
18     if ( **(_DWORD **)snake == apple && *(_DWORD *)(*(_DWORD *)snake + 4) ==
19     {
20       for ( i = 0; i <= 3; ++i )
21         v4[i] = v4[i + 1];
22       ++score;                                  // 分数加一
23       mess();
24       v5 = score;
25       if ( score == 1024 )                      // 分数等于1024执行flag函数
26         flag(v4);
27     }
```

而这里的v4 IDA只显示了有4位，说明IDA是不准的

for循环里，将v4的后一位赋值给前一位，

双击v4，去看看v4和v5的关系

```
-00000021                db ? ; undefined
-00000020 var_20         dd 4 dup(?)
-00000010 var_10         dd ?
```

v4和v5紧挨着，说明v5所在的位置正是v4[4]

模拟一下，刚开始score=0

- 第一次操控蛇吃到了苹果
  for循环将v4的后一位赋值给他的前一位，由于v4处于栈上，所以一开始是不知道他的值的
  score++变成了1
  v5也就是v4[4]=score =1

第一次，参数v4的[0] [1] [2] [3]位不知道，[4]=1 分数score=1

- 第二次操控蛇吃到了苹果
  score的 [0][1][2][3] 是不知道的，在for循环中，将[1] [2] [3]位赋值给[0] [1] [2]位
  将[4]赋值给[3]，也就是 [3]=1 [4]=1
  score++变成了2
  [4]变成了2
  第二次，参数v4的[0] [1] [2] 位不知道，[3]=1 [4]=2 分数score=2
- 第三次，参数v4的[0] [1]位不知道，[2]=1 [3]=2 [4]=3 分数score=3
- 第四次，参数v4的[0]位不知道，[1]=1 [2]=2 [3]=3 [4]=4 分数score=4
- 第四次，参数v4的[0]=1 [1]=2 [2]=3 [3]=4 [4]=5 分数score=5
- 第n(n>=4)次，v4[0]=score-4，v4[1]=score-3，v4[2]=score-2，v4[3]=score-1，v4[4]=score

可以知道flag是历史分数

当score等于1024的时候，传进去的v4就变成了v4[0]=1020，v4[1]=1021，v4[2]=1022，v4[3]=1023，v4[4]=1024

解法2 写C脚本

知道原始数据知道加密算法可以写脚本

```
#include <iostream>
#include <string>
using namespace std;
int arr[] = { 1020,1021,1022,1023,1024 };
char enstr[] = "bxukv{pW1SiFW_J0_jV}";
int len = 20;
void encode(char sstr[])
{
    int temp;
    for (int i = 0; i < len; i++)
    {
        if (sstr[i] >= 'a' && sstr[i]<='z')
        {
            temp = (sstr[i] - 'a' + arr[i % 5]) % 26 + 'A';
        }
        else
        {
            if (sstr[i] >= 'A' && sstr[i] <= 'Z')
            {
                temp = (sstr[i] - 'A' + arr[i % 5]) % 26 + 'a';
            }
            else
                temp = sstr[i];
        }
```

```
        sstr[i] = temp;
    }
}
int main()
{
    encode(enstr);
    cout << enstr;
    return 0;
}
```

解法3 修改分数以及v4数组

使用CE修改score<1020，比如1019（不然1020无法赋值到数组数据），然后玩几下得分达到1024flag自动出现

使用OD修改分数及v4数组的内存，让分数等于1024，v4分别为1020,1021,1022,1023,1024

或使用IDA修改

最后

直接修改分数等于1024是没用的，因为会验证V4数组

flag为

```
HECTF{We1cOme_t0_Re}
```