buuoj re逆向33-48题writeup(截图+c++源码+过程)

原创

weixin_51275728 于 2021-12-07 18:20:35 发布 2898 收藏

分类专栏： ctfre 文章标签： c++ 安全 信息安全 反编译

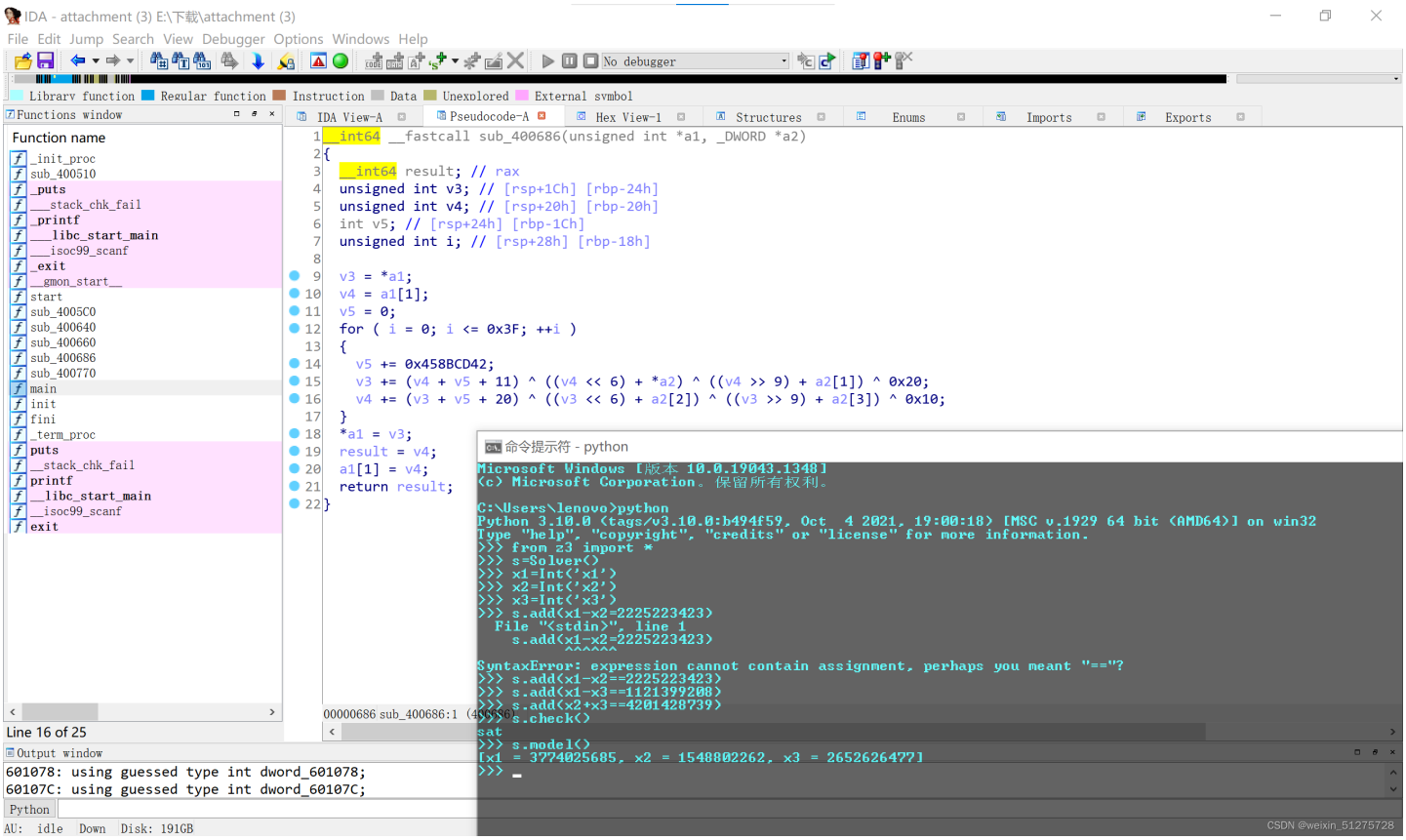
本文链接：https://blog.csdn.net/weixin_51275728/article/details/121773467

Re

33-48

33 xxor

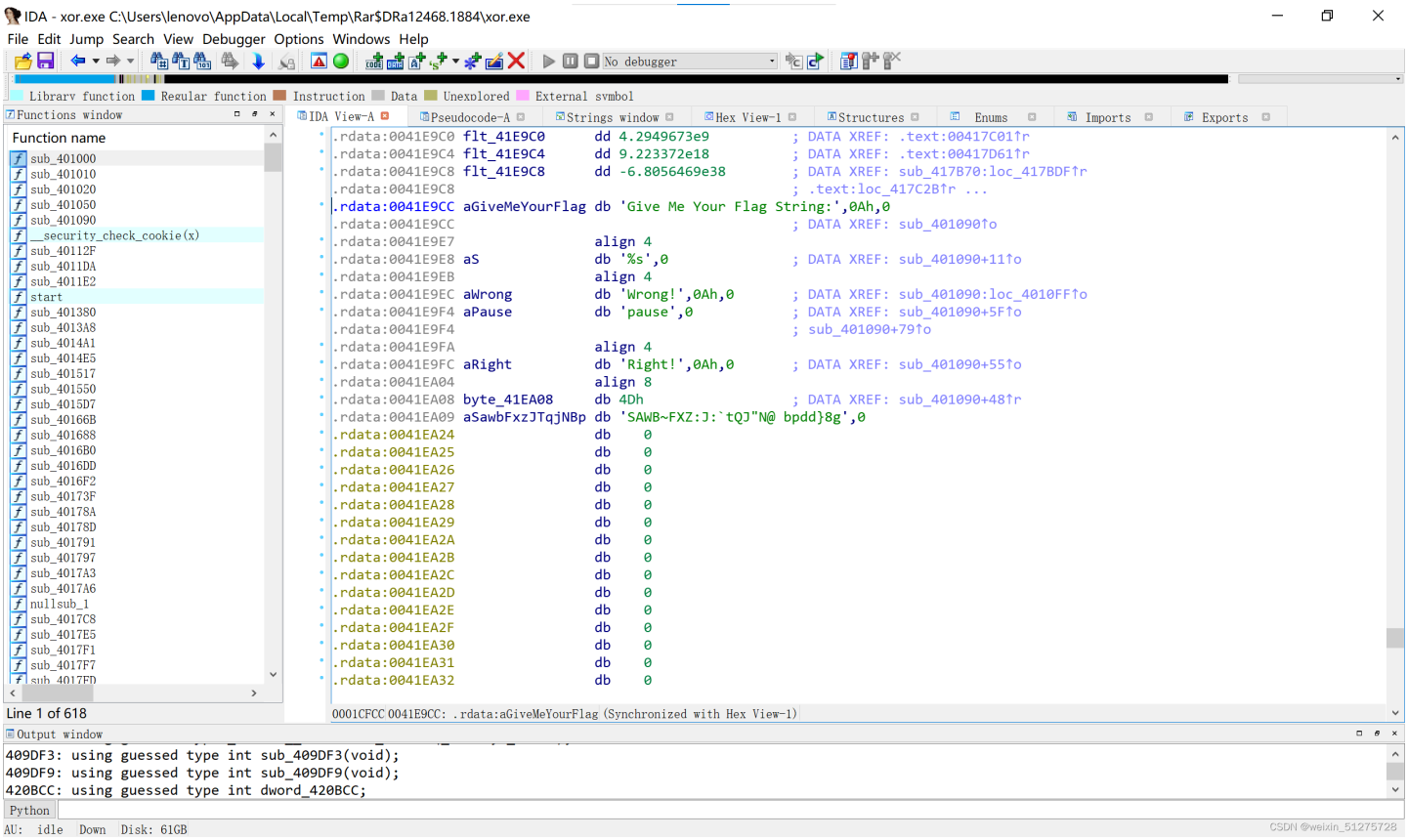
33 xxor 简单的二重循环，然后比较。记得把数据换成unsigned不然就是不行




```
#include < iostream>
using namespace std;
int main(){
unsigned int a2[4]={2,2,3,4};
unsigned int  a1[6]={0xDF48EF7E,0x20CAACF4,3774025685,1548802262,2652626477,0x84F30420};
  for(int j=0;j<=4;j+=2){
 unsigned int v3 = a1[j];
 unsigned int v4 = a1[1+j];
  int v5 = 0x458BCD42*64;
  for ( int i = 0; i <= 0x3F; ++i )
  {  v4 -= (v3 + v5 + 20) ^ ((v3 << 6) + a2[2]) ^ ((v3 >> 9) + a2[3]) ^ 0x10;
     v3 -= (v4 + v5 + 11) ^ ((v4 << 6) + a2[0]) ^ ((v4 >> 9) + a2[1]) ^ 0x20;
    v5 -= 0x458BCD42;}
  a1[j] = v3;
  a1[1+j] = v4;
}for(int i=0;i<6;i++)
 cout<<hex<<a1[i];
 cout<<endl;
 for (int i = 0; i < 6; i++)
  printf("%c%c%c", *((char*)&a1[i]+2), *((char*)&a1[i] + 1), *(char*)&a1[i]);
}
```

34xor

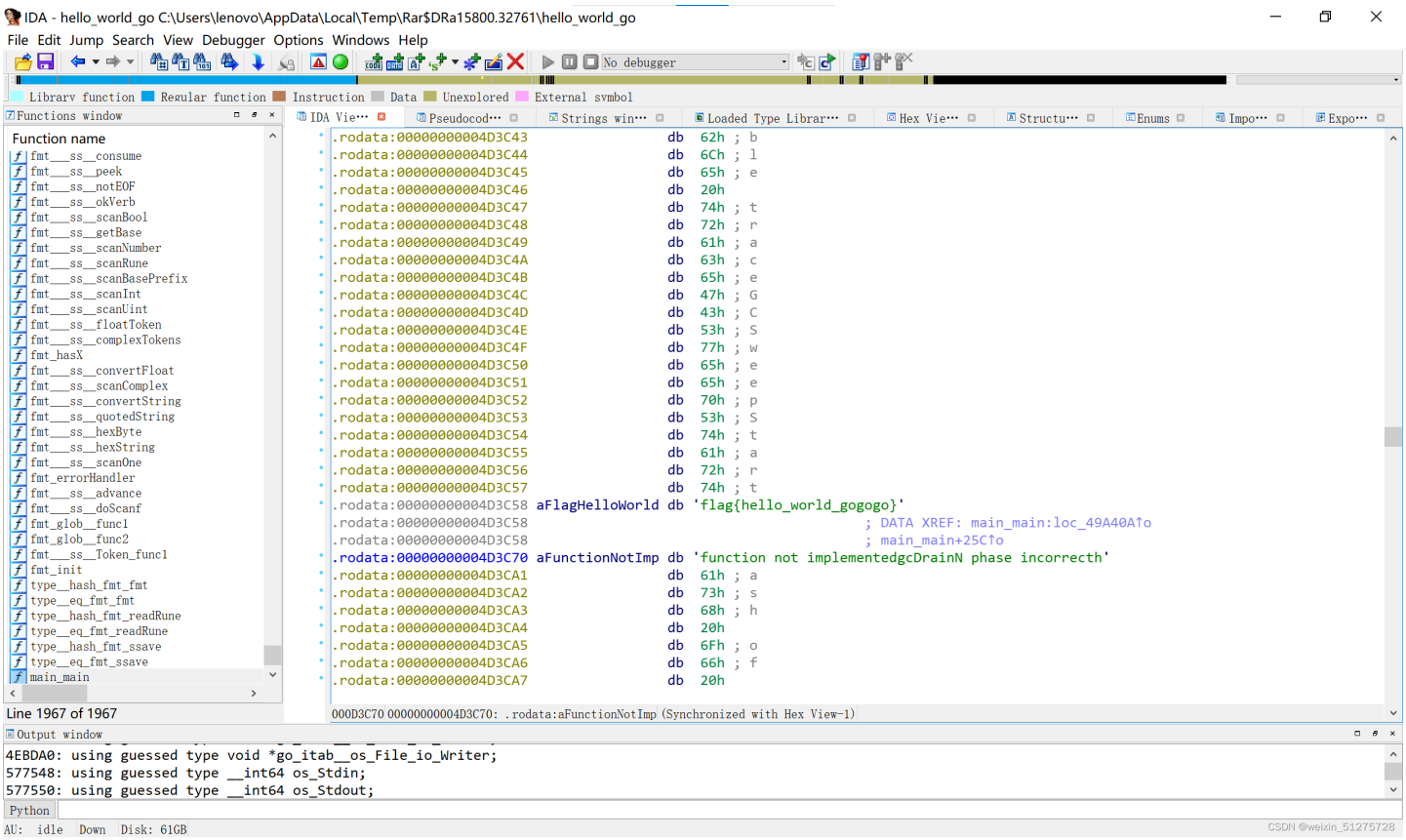
34 xor 就是简单的异或




```
#include < iostream>
using namespace std;
int main(){
 string a="MSAWB~FXZ:J:`tQJ\"N@ bpdd}8g";
 for(int i=0;i<a.length();i++)
{
char p=(a[i]^i);
cout<<p;}
}
```

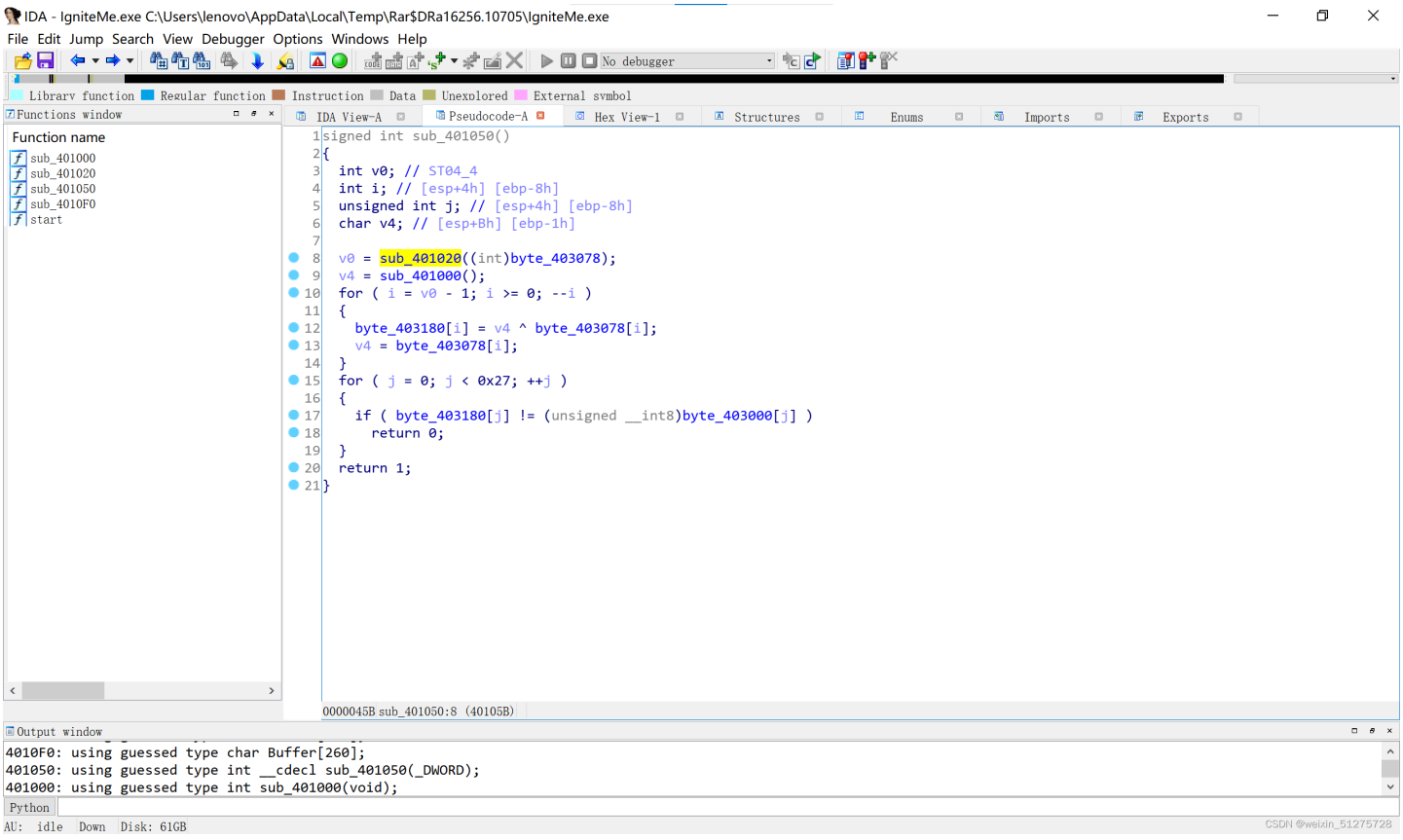
35 hello_world_go

35 hello_world_go 明码



36 igniteme

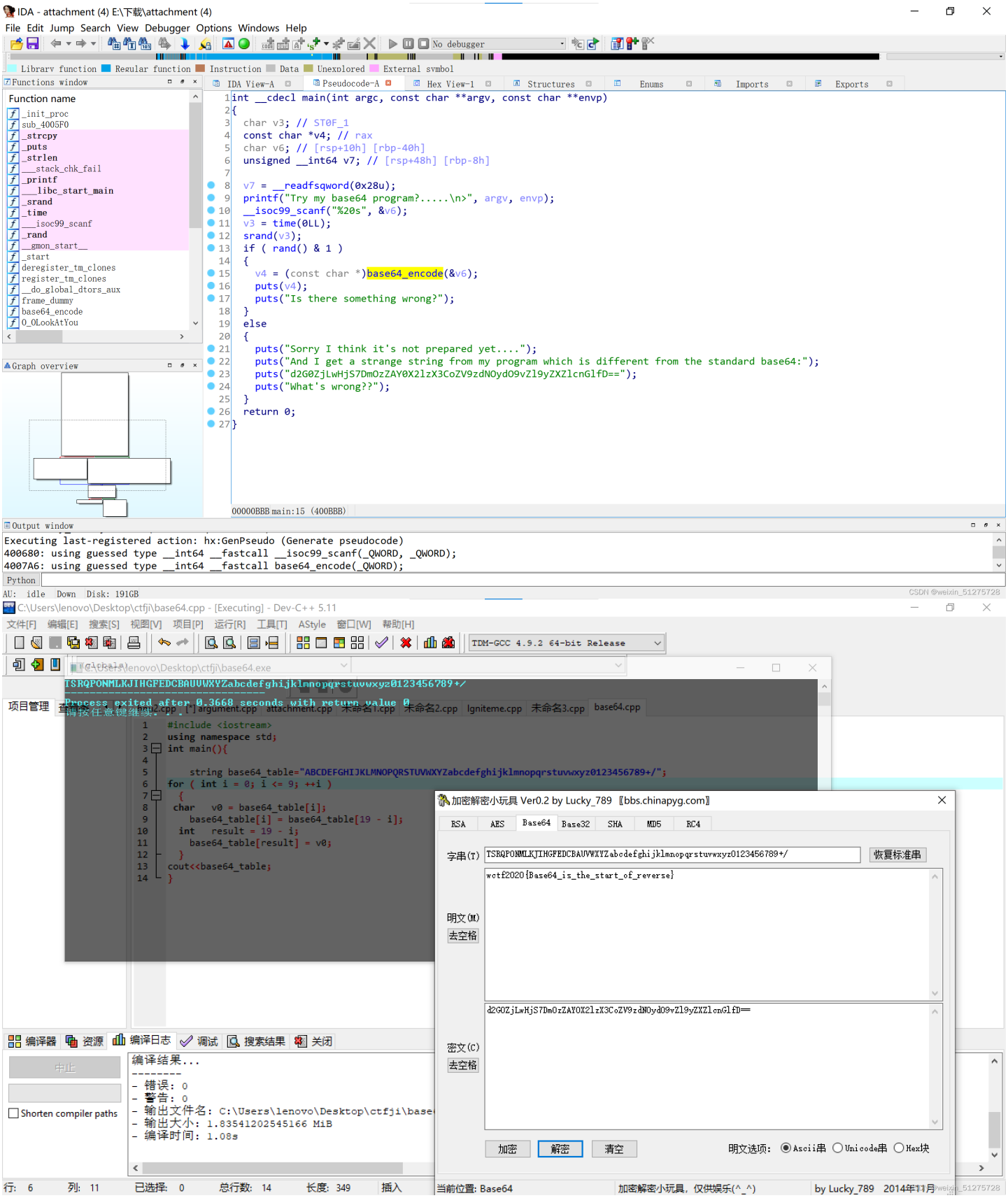
36 Igniteme,取了点巧，最后一位为m



```
#include < iostream>
#include <Windows.h>
using namespace std;
int main(){
int v4=0x69^’m’;
unsigned char ida_chars[] =
{
  0x0D, 0x26, 0x49, 0x45, 0x2A, 0x17, 0x78, 0x44, 0x2B, 0x6C,
  0x5D, 0x5E, 0x45, 0x12, 0x2F, 0x17, 0x2B, 0x44, 0x6F, 0x6E,
  0x56, 0x09, 0x5F, 0x45, 0x47, 0x73, 0x26, 0x0A, 0x0D, 0x13,
  0x17, 0x48, 0x42, 0x01, 0x40, 0x4D, 0x0C, 0x02, 0x69
};
for(int i=0x26;i>=0;i--)
 {for(int j=1;j<128;j++){
  char p=j^v4;
  {if(p!=ida_chars[i])
  continue;
  }
  char pp=j;
  cout<<pp;
  v4=j;
  break; } }
}
```

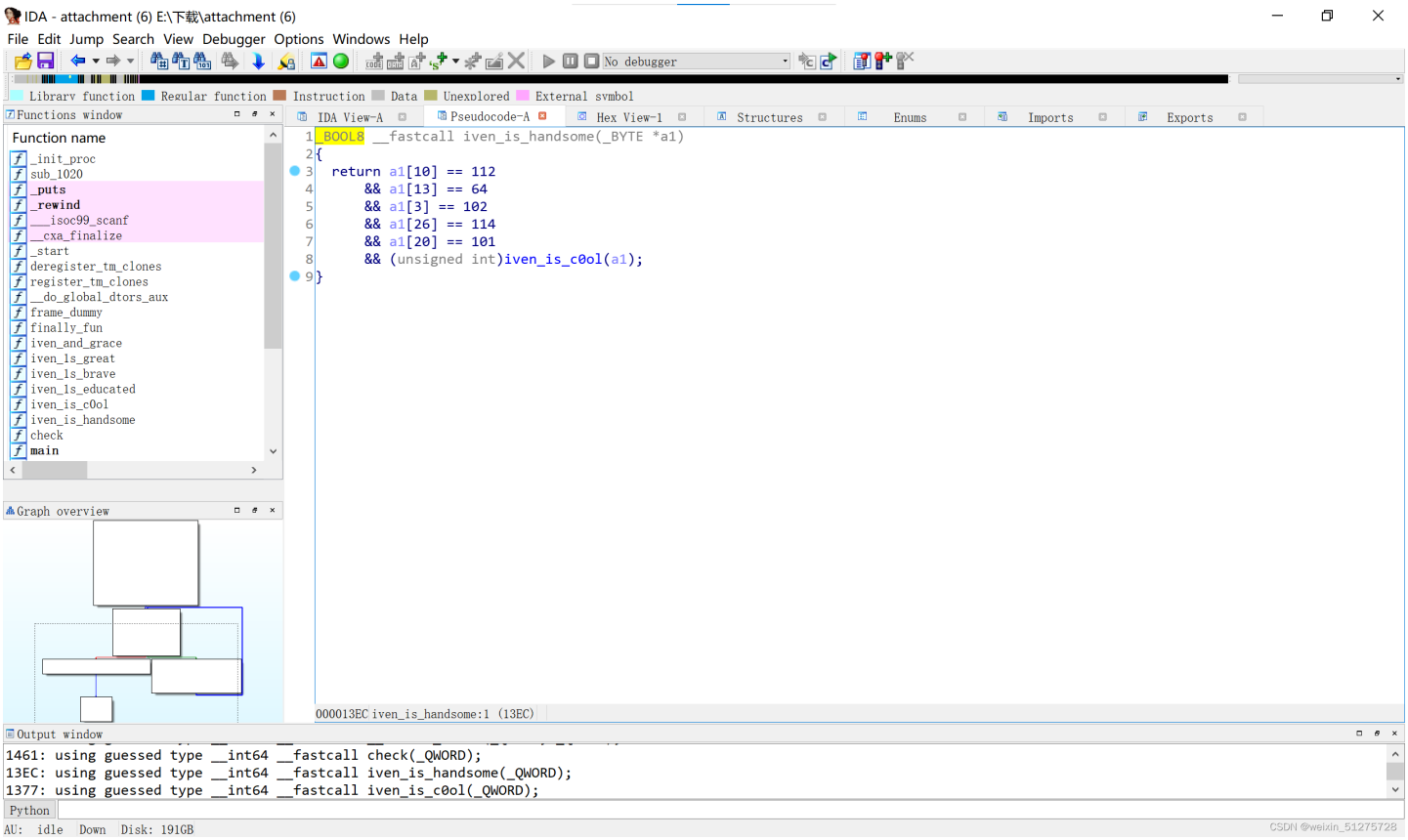
37 level3

37 level3 变表的base64 ,但我先开始没看出来哪里变了。就翻init,然后有发现



38 crossfun

38 crossfun 嵌套函数哈哈哈




```
#include < iostream>
using namespace std;
int main(){
char a1[33];
a1[10] = 'p';
       a1[13] = '@';
       a1[3] ='f';
       a1[26] = 'r';
       a1[20] ='e';
a1[7] = 48;
     a1[16] =95;
      a1[11] = 112;
       a1[23] = 101;
       a1[30] = 117;
       a1[0] = 119 , a1[6] = 50 , a1[22] = 115 , a1[31] = 110 ,a1[12] = 95;
       a1[15] = 100;
      a1[8] = 123;
       a1[18] = 51;
       a1[28] = 95;
      a1[21] = 114;
      a1[2] = 116;
      a1[9] = 99;
    a1[32] = 125;
    a1[19] = 118;
     a1[5]= 48;
    a1[14] = 110;
    a1[4] = 50, a1[17] = 114, a1[29] = 102 , a1[17] = 114 , a1[24] = 95;
    a1[1] = 99, a1[25] = 64;
 a1[27] = 101;
cout<<a1;
}
```

39 overlong

39 overlong 直接转成c++,分析发现输出字符短了。答案：我就是死也不会告诉你答案：真香

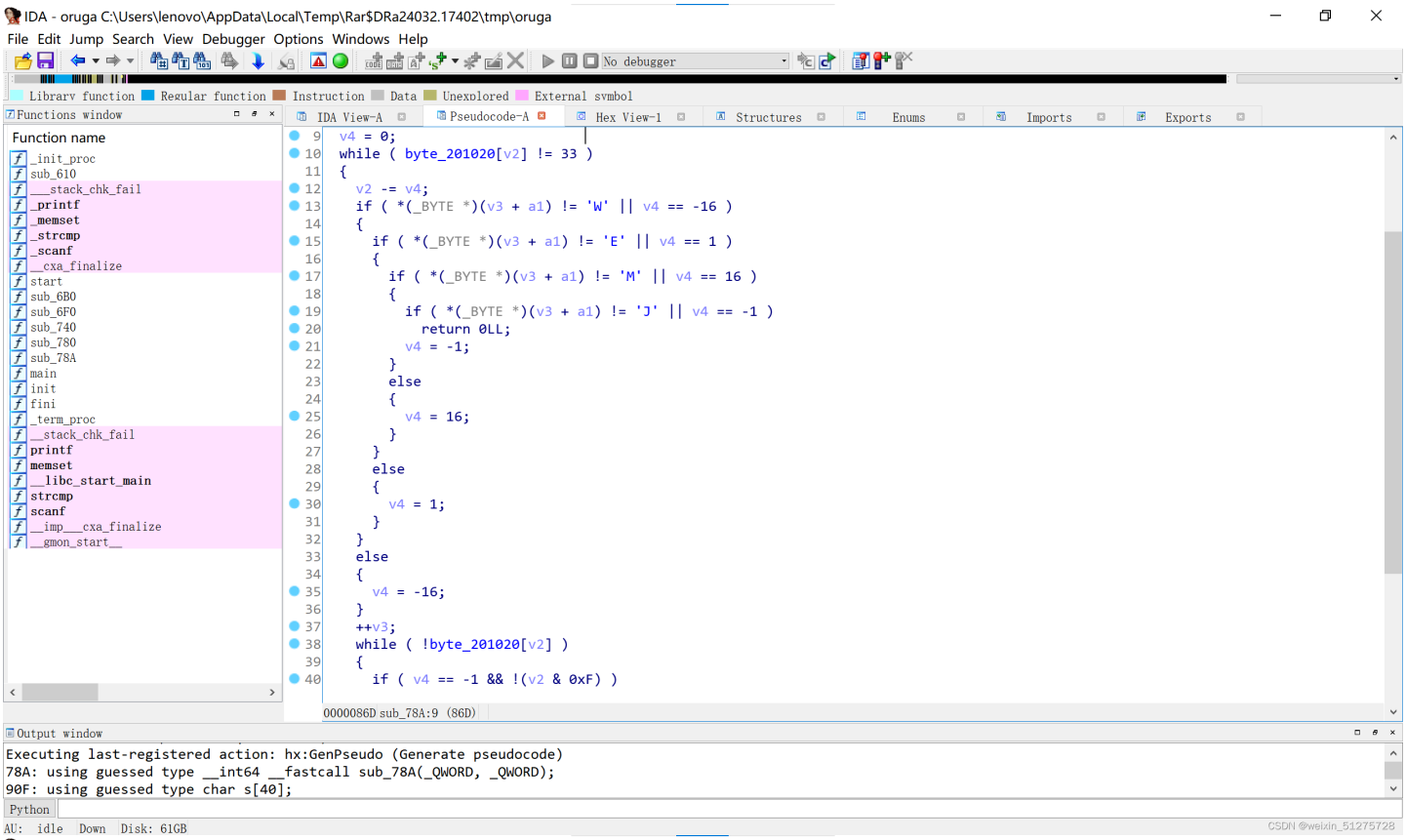
哈哈哈

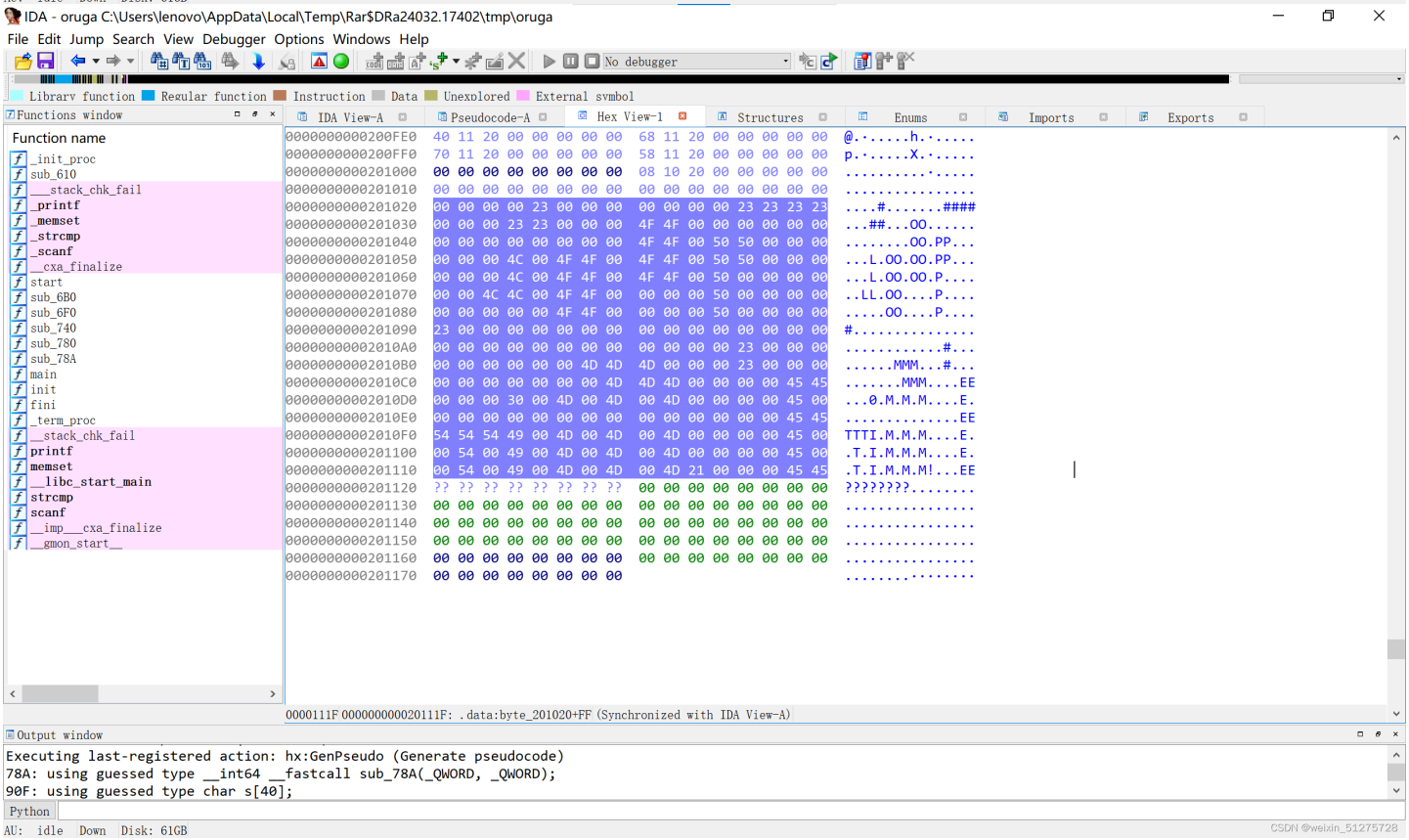
```
#include < iostream>
#include <Windows.h>
using namespace std;
int sub_401000(char *a1,unsigned  char * a2){
int v3; // [esp+0h] [ebp-8h]
  int v4; // [esp+4h] [ebp-4h]

  if ( *a2 >> 3 == 30 )
  {
    v4 = a2[3] & 0x3F | ((a2[2] & 0x3F) << 6) | ((a2[1] & 0x3F) << 12) | ((*a2 & 7) << 18);
    v3 = 4;
  }
  else if ( *a2 >> 4 == 14 )
  {
    v4 = a2[2] & 0x3F | ((a2[1] & 0x3F) << 6) | ((*a2 & 0xF) << 12);
    v3 = 3;
  }
  else if ( *a2 >> 5 == 6 )
  {
    v4 = a2[1] & 0x3F | ((*a2 & 0x1F) << 6);
    v3 = 2;
  }
  else
  {
    v4 = *a2;
    v3 = 1;
  }
  *a1 = v4;
  return v3;}
int sub_401160( char *a1, unsigned char *a2, int a3)
{
  int v3; // ST08_4
  unsigned int i; // [esp+4h] [ebp-4h]

  for ( i = 0; i < a3; ++i )
  {
    a2 += sub_401000(a1, a2);
    v3 = *a1++;
    if ( !v3 )
      break;
  }
  return i;}
int main(){
unsigned char ida_chars[] =
{
  0xE0, 0x81, 0x89, 0xC0, 0xA0, 0xC1, 0xAE, 0xE0, 0x81, 0xA5,
  0xC1, 0xB6, 0xF0, 0x80, 0x81, 0xA5, 0xE0, 0x81, 0xB2, 0xF0,
  0x80, 0x80, 0xA0, 0xE0, 0x81, 0xA2, 0x72, 0x6F, 0xC1, 0xAB,
  0x65, 0xE0, 0x80, 0xA0, 0xE0, 0x81, 0xB4, 0xE0, 0x81, 0xA8,
  0xC1, 0xA5, 0x20, 0xC1, 0xA5, 0xE0, 0x81, 0xAE, 0x63, 0xC1,
  0xAF, 0xE0, 0x81, 0xA4, 0xF0, 0x80, 0x81, 0xA9, 0x6E, 0xC1,
  0xA7, 0xC0, 0xBA, 0x20, 0x49, 0xF0, 0x80, 0x81, 0x9F, 0xC1,
  0xA1, 0xC1, 0x9F, 0xC1, 0x8D, 0xE0, 0x81, 0x9F, 0xC1, 0xB4,
  0xF0, 0x80, 0x81, 0x9F, 0xF0, 0x80, 0x81, 0xA8, 0xC1, 0x9F,
```

```
  0xF0, 0x80, 0x81, 0xA5, 0xE0, 0x81, 0x9F, 0xC1, 0xA5, 0xE0,
  0x81, 0x9F, 0xF0, 0x80, 0x81, 0xAE, 0xC1, 0x9F, 0xF0, 0x80,
  0x81, 0x83, 0xC1, 0x9F, 0xE0, 0x81, 0xAF, 0xE0, 0x81, 0x9F,
  0xC1, 0x84, 0x5F, 0xE0, 0x81, 0xA9, 0xF0, 0x80, 0x81, 0x9F,
  0x6E, 0xE0, 0x81, 0x9F, 0xE0, 0x81, 0xA7, 0xE0, 0x81, 0x80,
  0xF0, 0x80, 0x81, 0xA6, 0xF0, 0x80, 0x81, 0xAC, 0xE0, 0x81,
  0xA1, 0xC1, 0xB2, 0xC1, 0xA5, 0xF0, 0x80, 0x80, 0xAD, 0xF0,
  0x80, 0x81, 0xAF, 0x6E, 0xC0, 0xAE, 0xF0, 0x80, 0x81, 0xA3,
  0x6F, 0xF0, 0x80, 0x81, 0xAD, 0x00
};

 unsigned int v4; // eax
 char Text[128]; // [esp+0h] [ebp-84h]
  unsigned int v7; // [esp+80h] [ebp-4h]
  v4 = sub_401160(Text, ida_chars, 180);//180换成其他数也行
  v7 = v4;
  Text[v4] = 0;
  MessageBoxA(0, Text,"Output", 0);
  return 0;
}
```

41 orgua

41 orgua 不知道大家有没有玩过三月之庭，看到图时我感觉很熟悉





就是说走一个方向时在遇到障碍物前不会停，最后到感叹号。最开始在左上角。

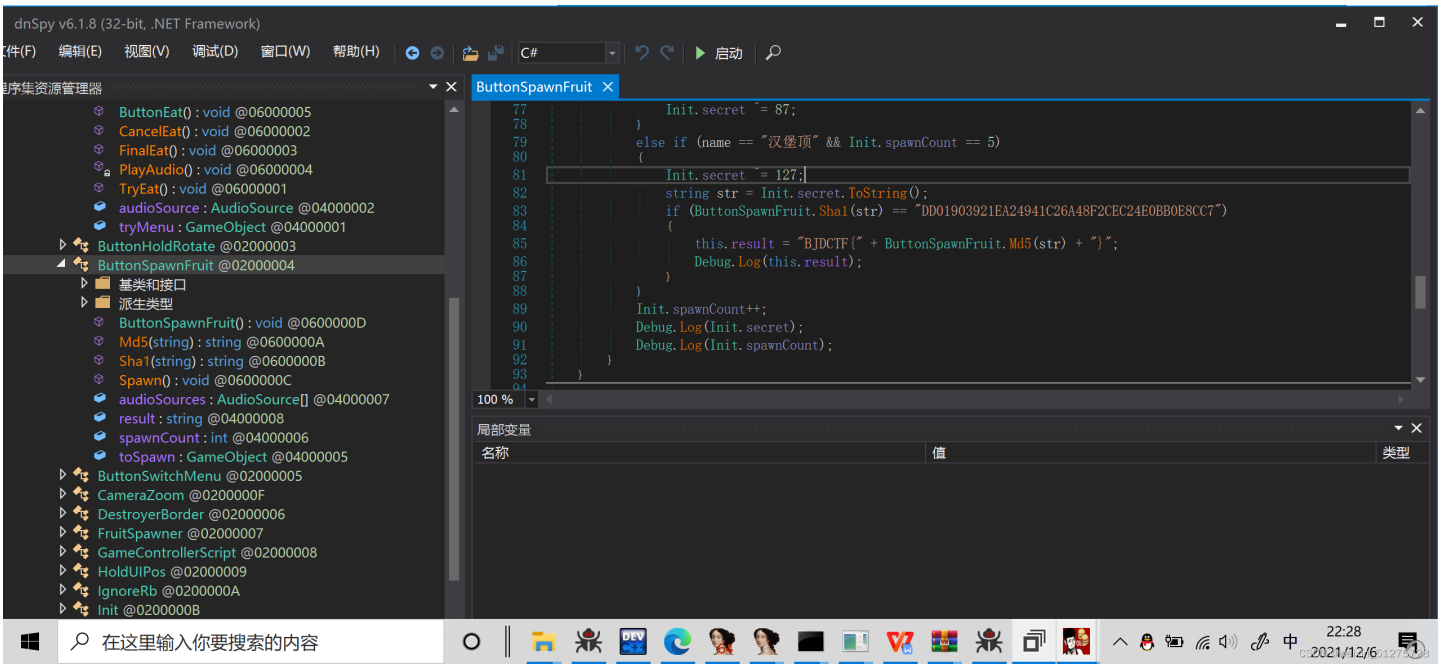
就一个迷宫，也想难倒我？然后卡了十多分钟。。。血压都上来了。

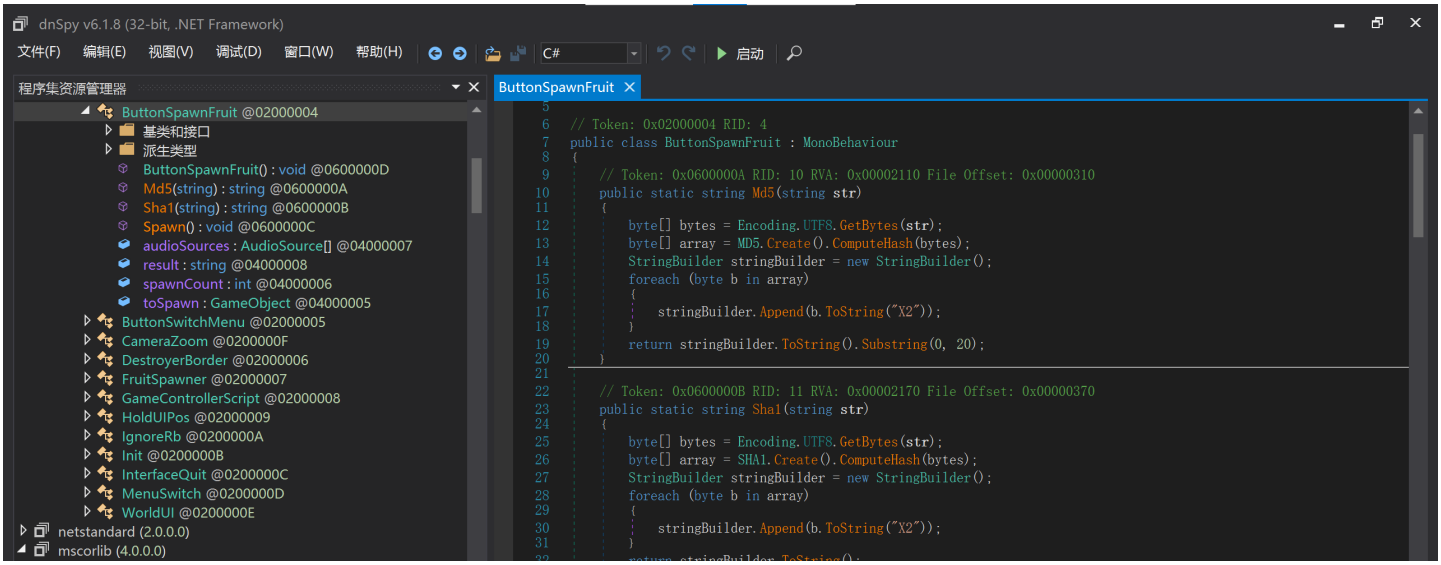
MEWEMEWJMEWJM

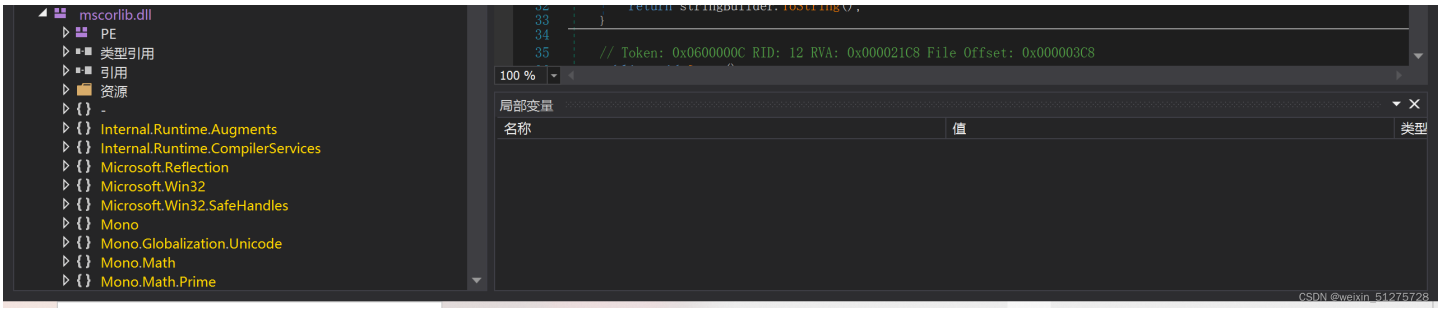
40 BJD hamburger competiton

40 [BJDCTF2020]BJD hamburger competition

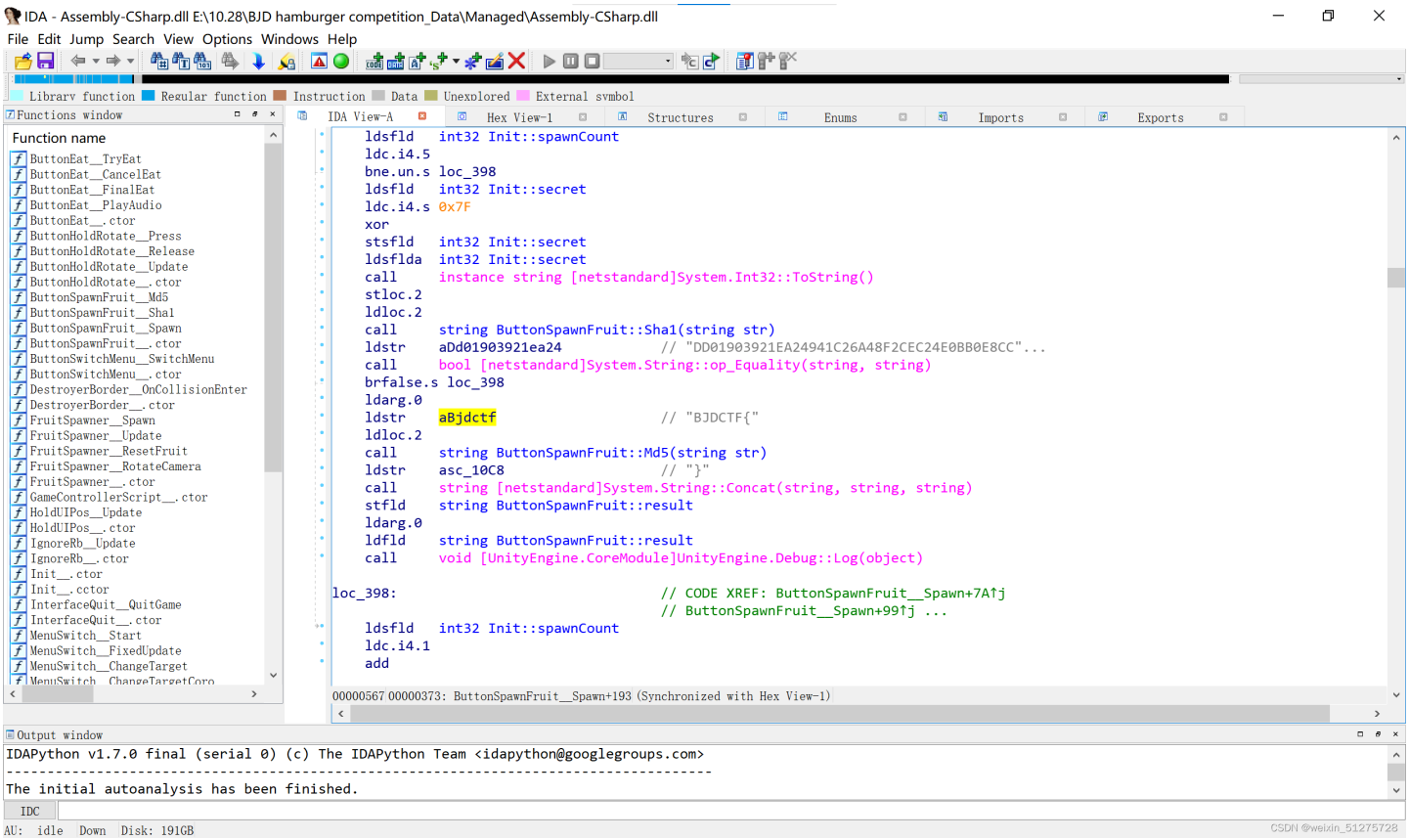

中文 | English | Россия 未登录
首页
在线解密
免费批量解密工具
会员中心
广告位长期招租
QQ:1878399009
定位精准用户
多关键词百度首页
常用哈希加密解密 >> sha1在线加密 | sha1在线解密
DD01903921EA24941C26A48F2CEC24E0BB0E8CC7
在线加密
在线解密
解密成功，结果是：1001
备案号：闽ICP备16008567号-1
ttmd5@yahoo.com QQ:1878399009 md5互助群:303488034
友情链接：上班摸鱼神器-单行阅读器 奇乐浓浓小说网 SecSilo 华域联盟 Mrxn's Blog 七行者博客 Arvin's Blog


dnSpy v6.1.8 (32-bit, .NET Framework)
ButtonSpawnFruit
Init.secret *= 87;
else if (name == "汉堡顶" && Init.spawnCount == 5)
Init.secret *= 127;
string str = Init.secret.ToString();
if (ButtonSpawnFruit.Sha1(str) == "DD01903921EA24941C26A48F2CEC24E0BB0E8CC7")
this.result = "BJDCTF{" + ButtonSpawnFruit.Md5(str) + "}";
Debug.Log(this.result);
Init.spawnCount++;
Debug.Log(Init.secret);
Debug.Log(Init.spawnCount);
局部变量
名称
值
类型
在这里输入你要搜索的内容
22:28
2021/12/6


dnSpy v6.1.8 (32-bit, .NET Framework)
ButtonSpawnFruit
// Token: 0x02000004 RID: 4
public class ButtonSpawnFruit : MonoBehaviour
// Token: 0x0600000A RID: 10 RVA: 0x00002110 File Offset: 0x00000310
public static string Md5(string str)
byte[] bytes = Encoding.UTF8.GetBytes(str);
byte[] array = MD5.Create().ComputeHash(bytes);
StringBuilder stringBuilder = new StringBuilder();
foreach (byte b in array)
stringBuilder.Append(b.ToString("X2"));
return stringBuilder.ToString().Substring(0, 20);
// Token: 0x0600000B RID: 11 RVA: 0x00002170 File Offset: 0x00000370
public static string Sha1(string str)
byte[] bytes = Encoding.UTF8.GetBytes(str);
byte[] array = SHA1.Create().ComputeHash(bytes);
StringBuilder stringBuilder = new StringBuilder();
foreach (byte b in array)
stringBuilder.Append(b.ToString("X2"));



这道题第一次做时不会，ida不行啊。



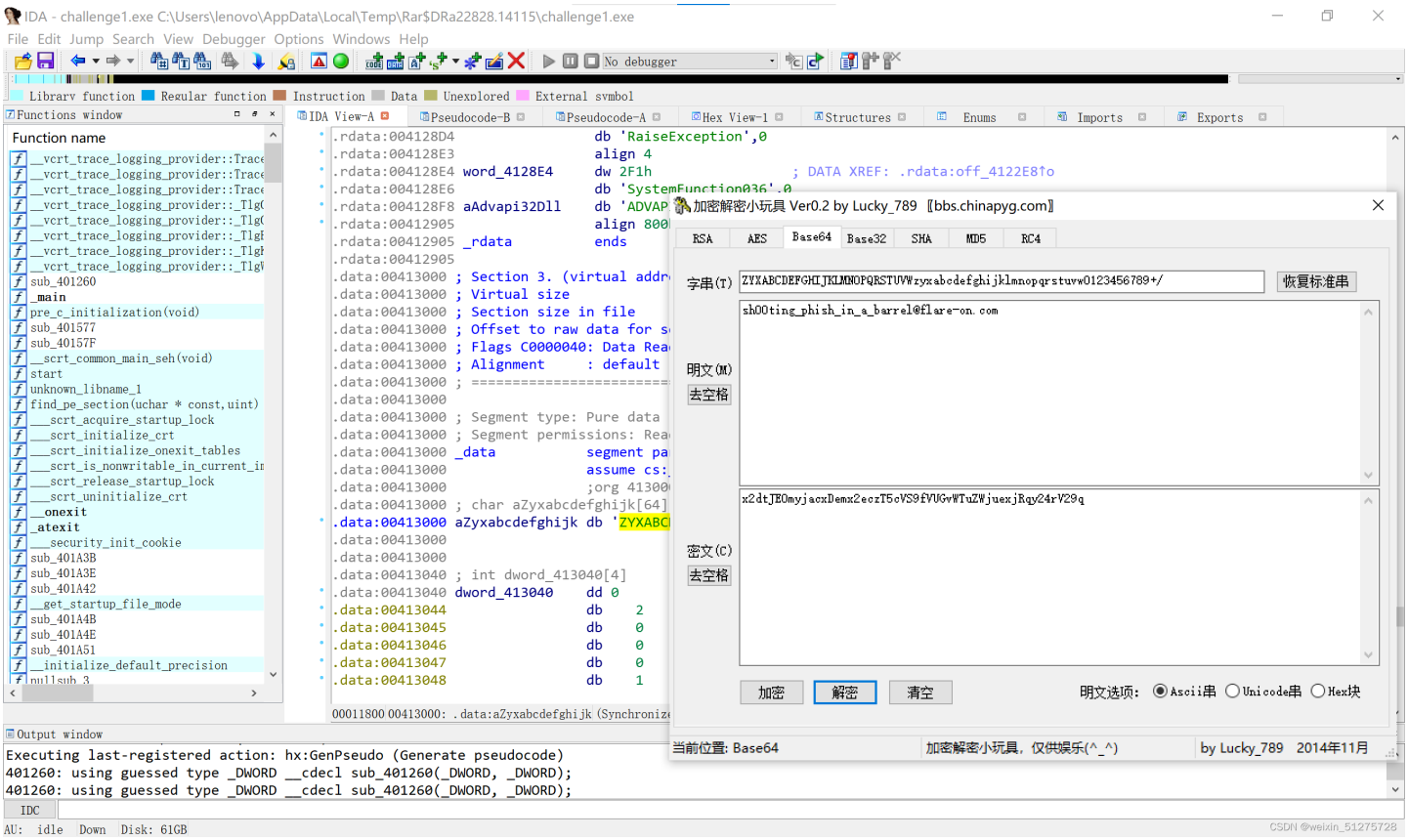
看别人题解才知道要用dnspy.而且代码要看全

Sha解密后md5加密

应该是转换大写了,且读取20个字符。

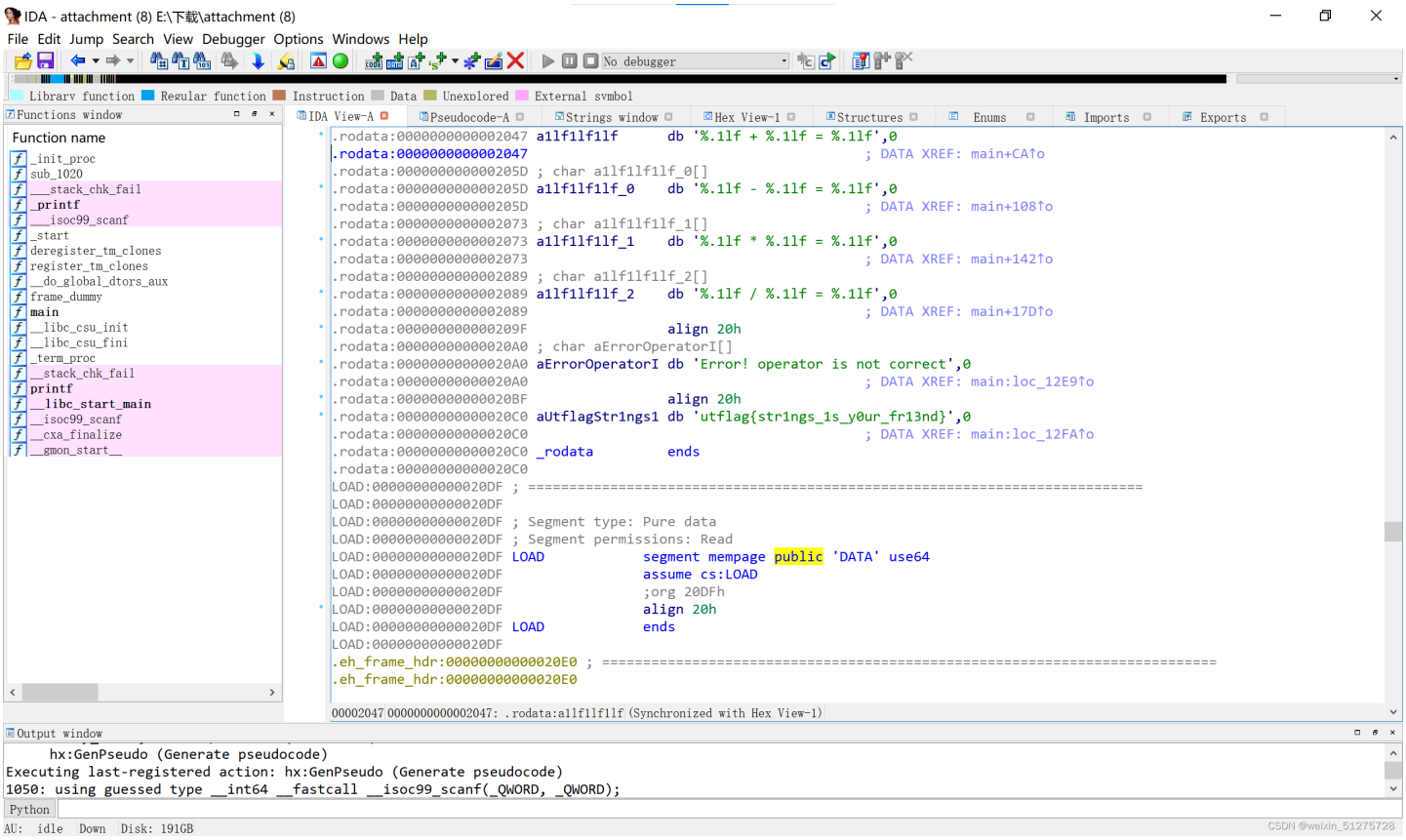
42 challenge

42 challenge base64 变表



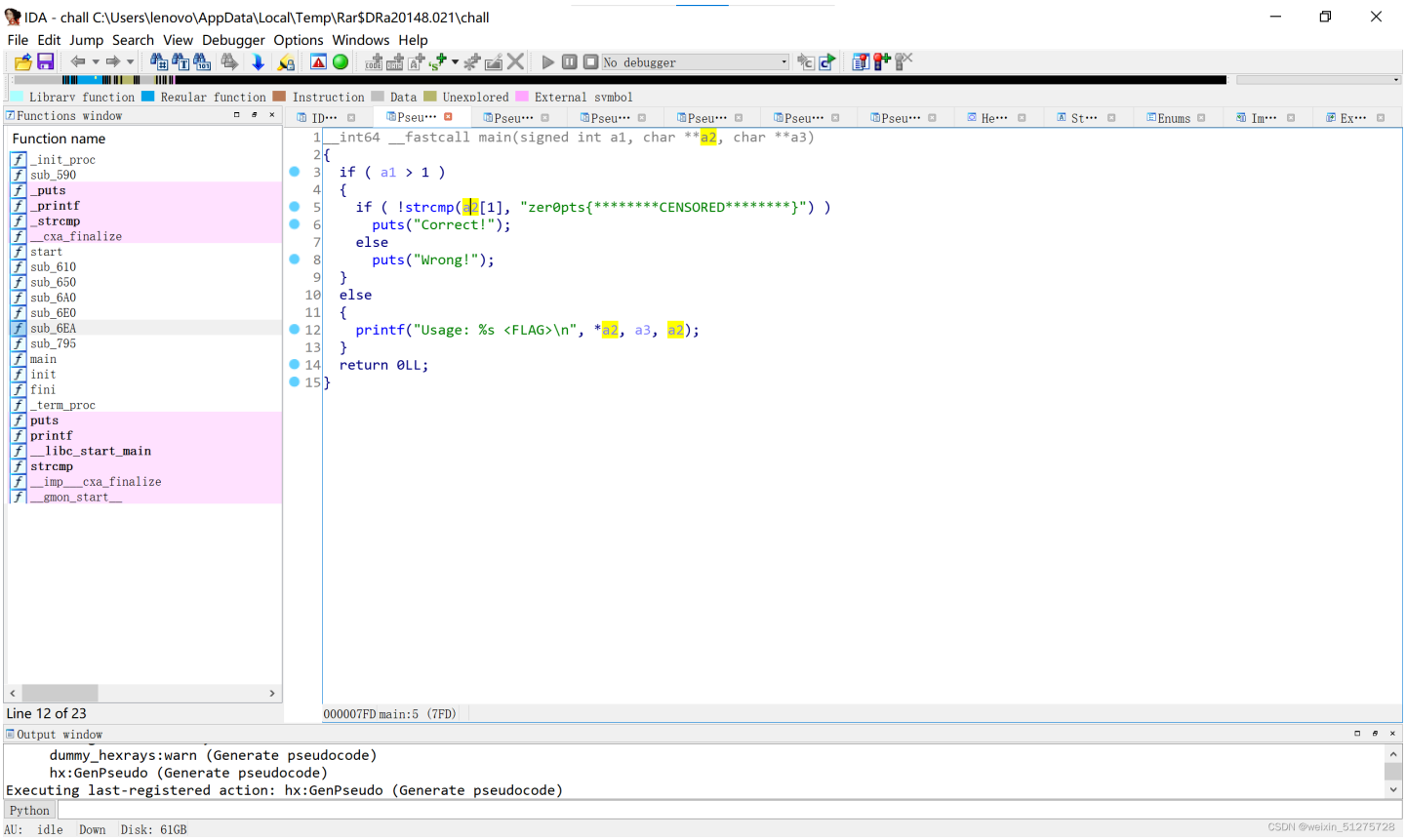
43 base-re

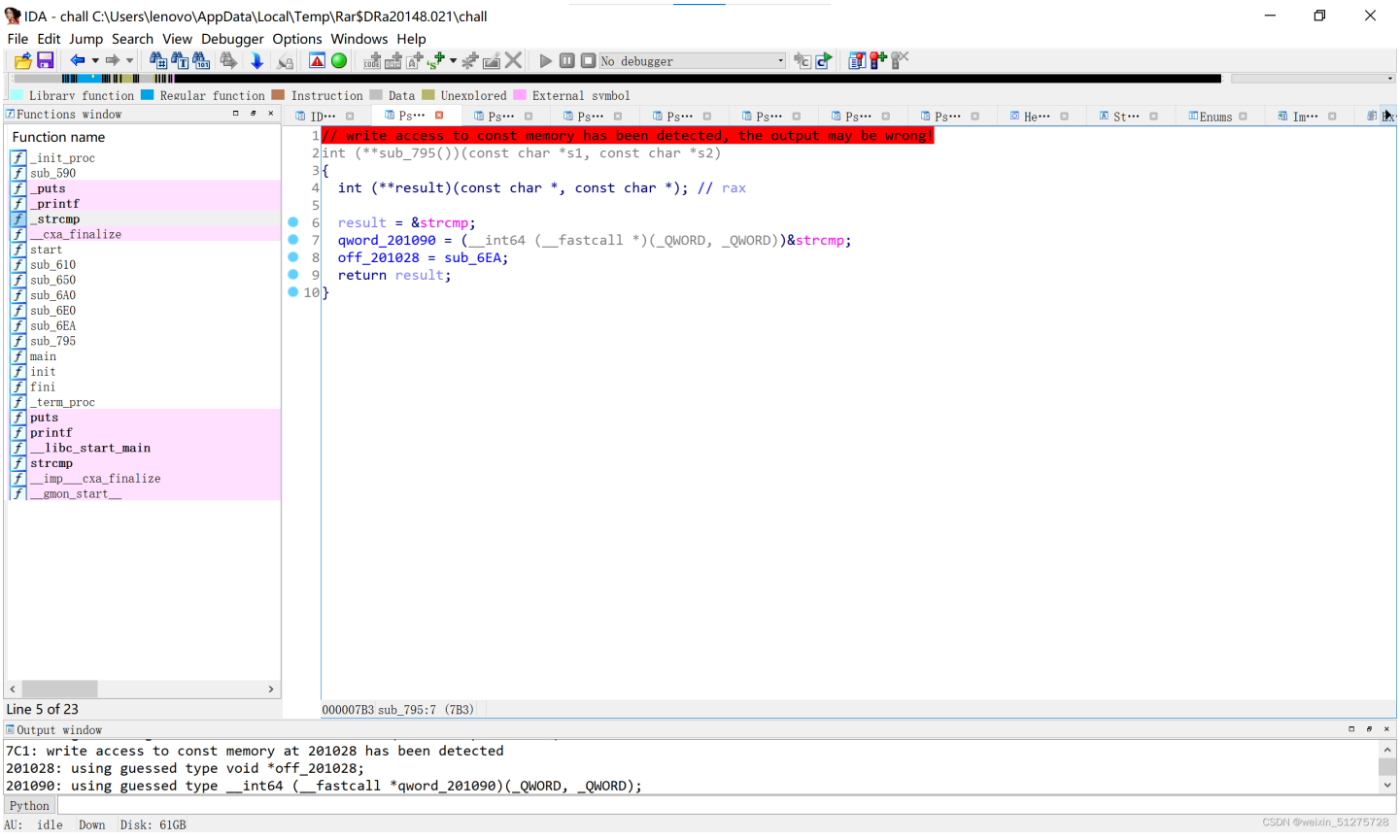
43 base-re 明码



44 easy strcmp

44[Zer0pts2020]easy strcmp 一点都不easy 找了半天没找到,我就乱翻函数





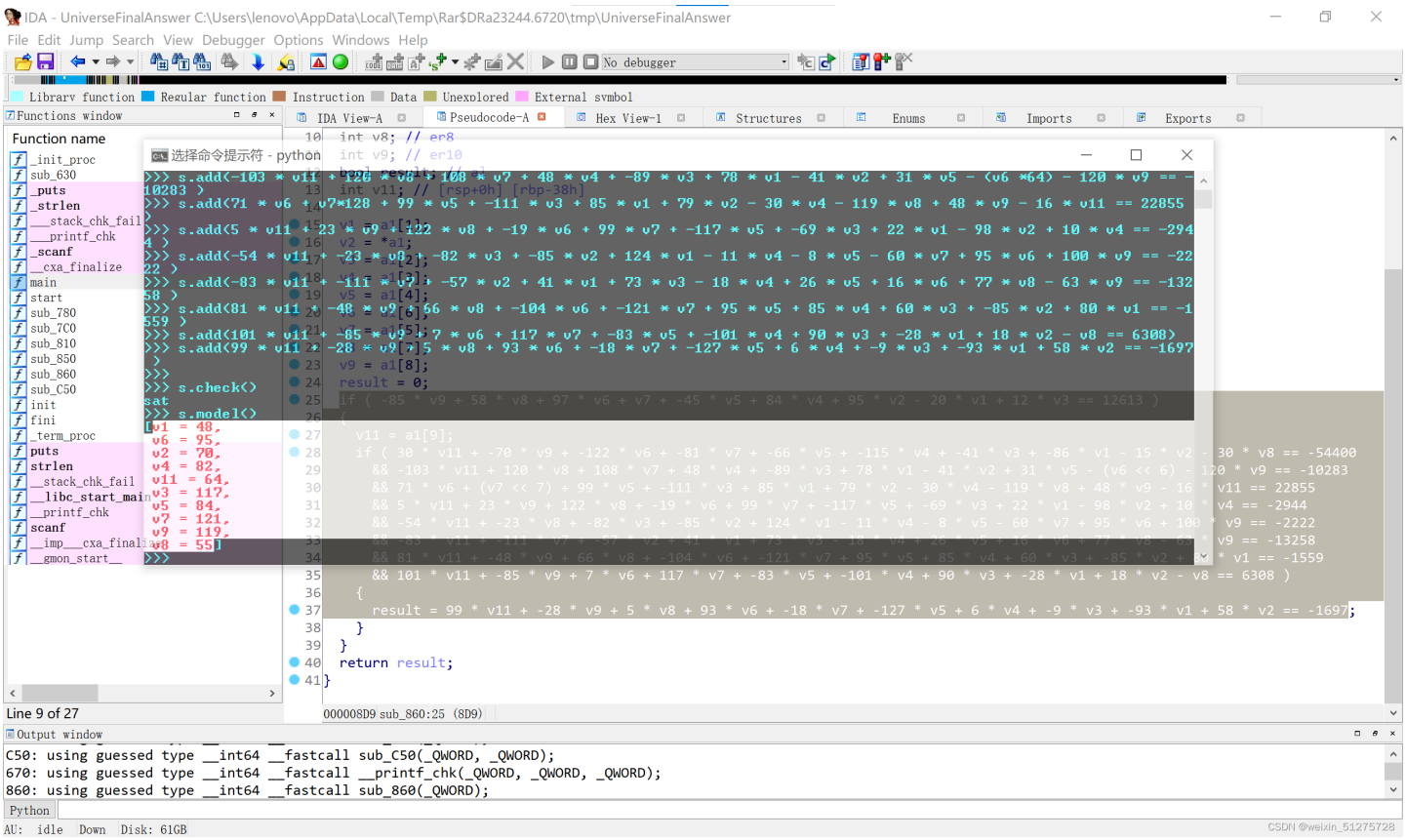
201090->795

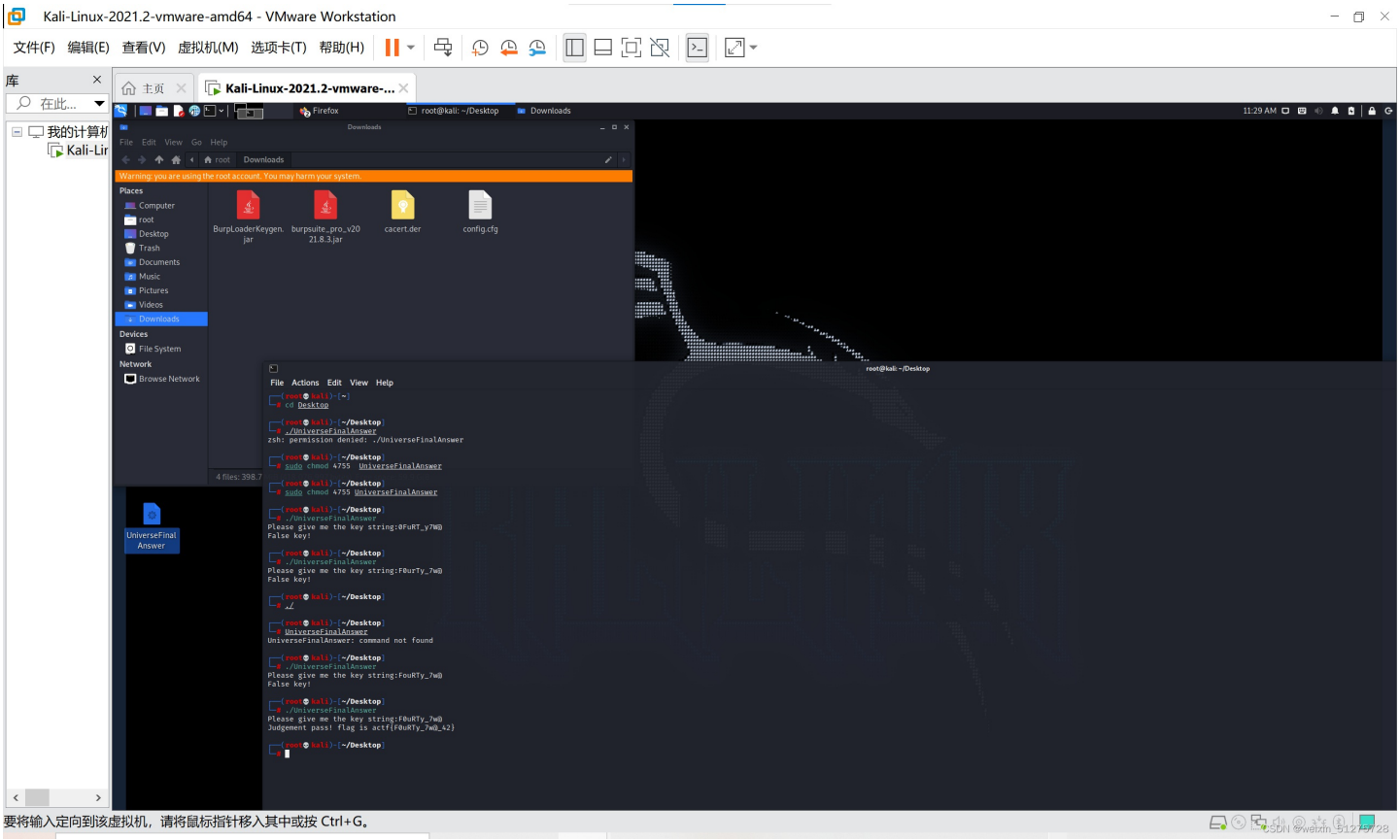
201028->6ea

```
#include < iostream>
using namespace std;
int main(){
 string p="zer0pts{********CENSORED********}";
 char a[24]={ 0x42, 0x09,
  0x4A, 0x49, 0x35, 0x43, 0x0A, 0x41, 0xF0, 0x19, 0xE6, 0x0B,
  0xF5, 0xF2, 0x0E, 0x0B, 0x2B, 0x28, 0x35, 0x4A, 0x06, 0x3A,
  0x0A, 0x4f};
  for(int i=0;i<24;i++){
   p[i+8]+=a[i];
  }
  cout<<p;}
```

45 UniverseFinalAnswer

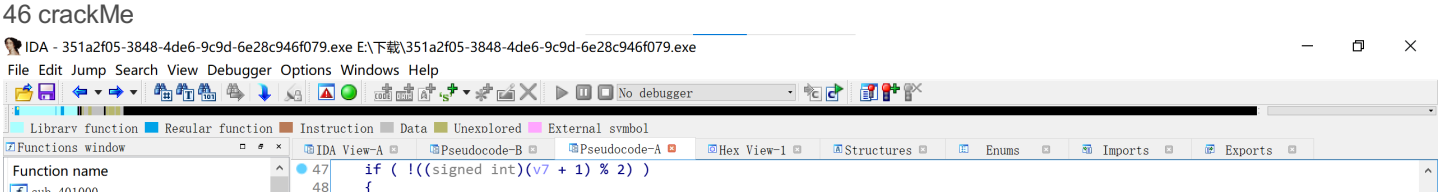
45UniverfinalanswerZ3yyds



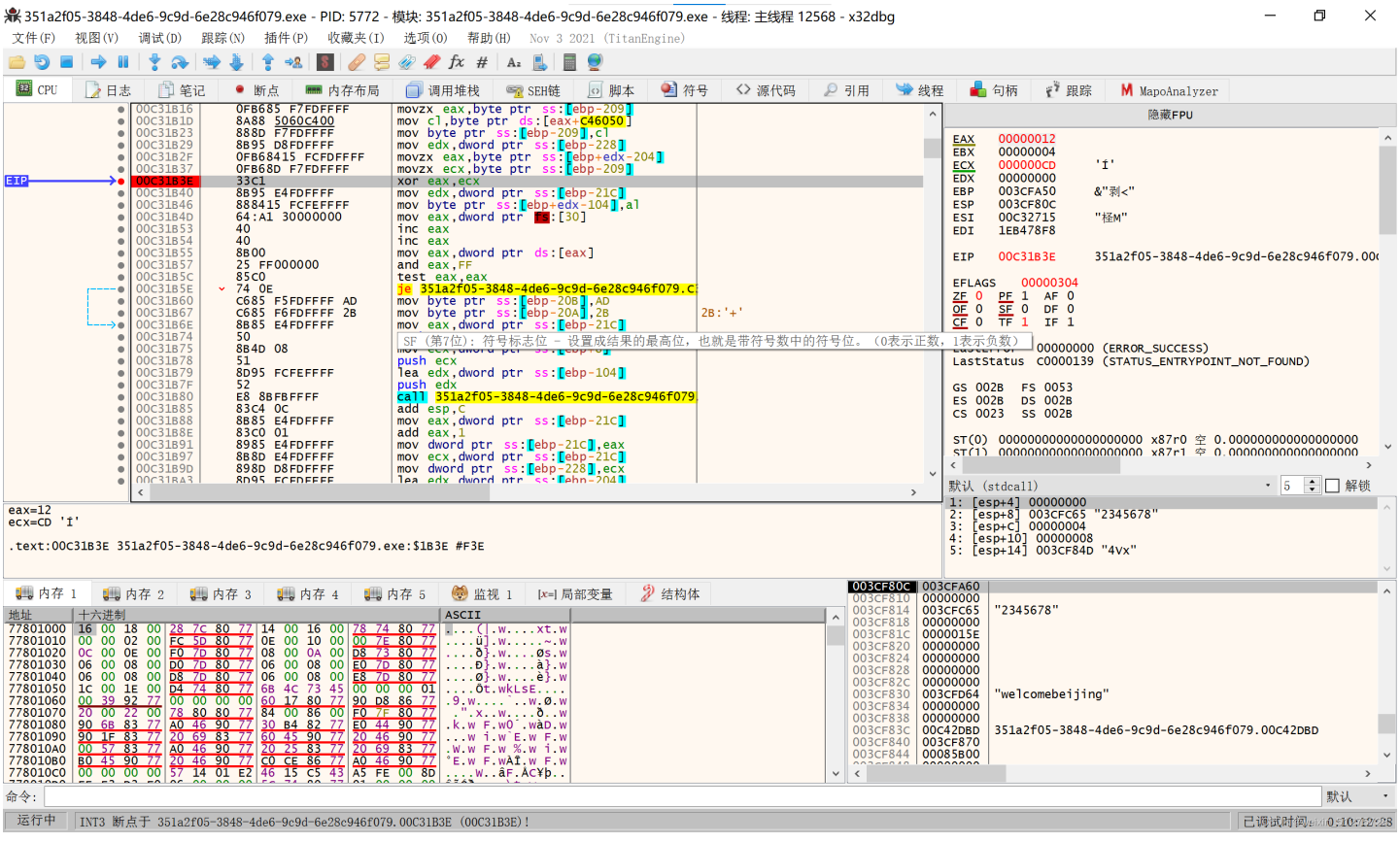


46 crackMe

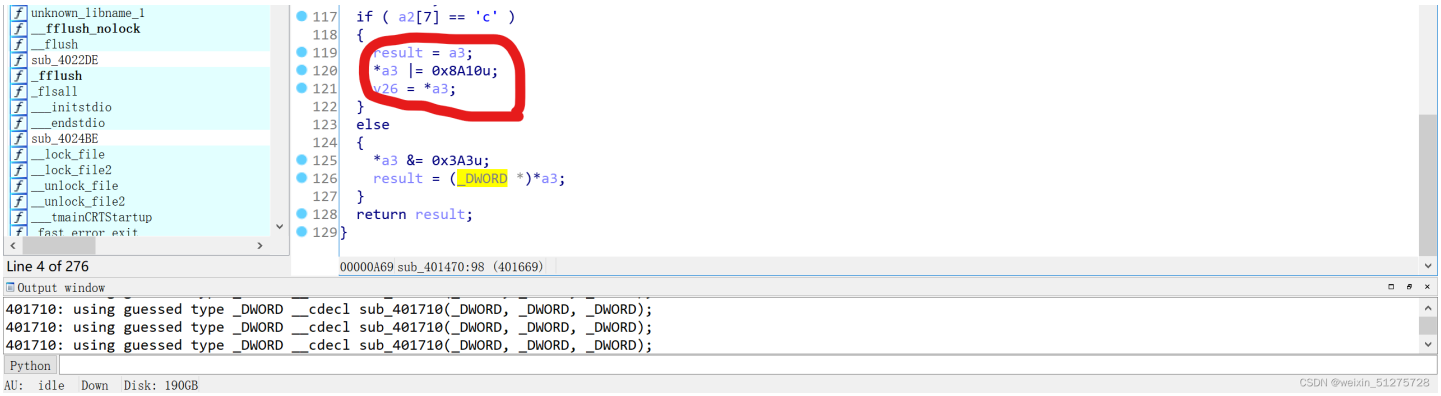
46 crackMe




sub_401090
sub_4011A0
sub_401470
sub_401710
sub_401830
_wmain
__security_check_cookie(x)
_LocaleUpdate::_LocaleUpdate(locale
__isalnum_l
__isdigit_l
__isspace_l
__isxdigit_l
_isalnum
_isdigit
_isspace
_isxdigit
__get_printf_count_output
_printf
_scanf
unknown_libname_1
__fflush_nolock
__flush
sub_4022DE
_fflush
_flsall
__initstdio
__endstdio
sub_4024BE
__lock_file
__lock_file2
__unlock_file
__unlock_file2
__tmainCRTStartup
_fast_error_exit
Line 3 of 276
00000F2F sub_401830:67 (401B2F)
Output window
414000: using guessed type void *off_414000;
401251: inconsistent fpu stack
401710: using guessed type _DWORD __cdecl sub_401710(_DWORD, _DWORD, _DWORD);
Python
AU: idle Down Disk: 190GB
CSDN @weixin_51275728


351a2f05-3848-4de6-9c9d-6e28c946f079.exe - PID: 5772 - 模块: 351a2f05-3848-4de6-9c9d-6e28c946f079.exe - 线程: 主线程 12568 - x32dbg
文件(F) 视图(V) 调试(D) 跟踪(N) 插件(P) 收藏夹(I) 选项(O) 帮助(H) Nov 3 2021 (TitanEngine)
CPU 日志 笔记 断点 内存布局 调用堆栈 SEH链 脚本 符号 源代码 引用 线程 句柄 跟踪 MapoAnalyzer
隐藏FPU
EAX 00000012
EBX 00000004
ECX 000000CD 'Í'
EDX 00000000
EBP 003CFA50 &"剥<"
ESP 003CF80C
ESI 00C32715 "桉M"
EDI 1EB478F8
EIP 00C31B3E 351a2f05-3848-4de6-9c9d-6e28c946f079.00
EFLAGS 00000304
ZF 0 PF 1 AF 0
OF 0 SF 0 DF 0
CF 0 TF 1 IF 1
00000000 (ERROR_SUCCESS)
LastStatus C0000139 (STATUS_ENTRYPOINT_NOT_FOUND)
GS 002B FS 0053
ES 002B DS 002B
CS 0023 SS 002B
ST(0) 00000000000000000000 x87r0 空 0.00000000000000000000
默认 (stdcall)
1: [esp+4] 00000000
2: [esp+8] 003CFC65 "2345678"
3: [esp+C] 00000004
4: [esp+10] 00000008
5: [esp+14] 003CF84D "4vx"
eax=12
ecx=CD 'Í'
.text:00C31B3E 351a2f05-3848-4de6-9c9d-6e28c946f079.exe:$1B3E #F3E
内存 1 内存 2 内存 3 内存 4 内存 5 监视 1 局部变量 结构体
地址 十六进制 ASCII
003CF80C 003CFA60
003CF810 00000000
003CF814 003CFC65 "2345678"
003CF818 00000000
003CF81C 0000015E
003CF820 00000000
003CF824 00000000
003CF828 00000000
003CF82C 00000000
003CF830 003CFD64 "welcomebeijing"
003CF834 00000000
003CF838 00000000
003CF83C 00C42DBD 351a2f05-3848-4de6-9c9d-6e28c946f079.00C42DBD
003CF840 003CF870
003CF844 00085B00
命令:
运行中
INT3 断点于 351a2f05-3848-4de6-9c9d-6e28c946f079.00C31B3E (00C31B3E)!


IDA - 351a2f05-3848-4de6-9c9d-6e28c946f079.exe E:\下载\351a2f05-3848-4de6-9c9d-6e28c946f079.exe
File Edit Jump Search View Debugger Options Windows Help
No debugger
Library function Regular function Instruction Data Unexplored External symbol
Functions window
Function name
IDA View-A Pseudocode-B Pseudocode-A Hex View-1 Structures Enums Imports Exports
sub_401000
sub_401090
sub_4011A0
sub_401470
sub_401710
sub_401830
_wmain
__security_check_cookie(x)
_LocaleUpdate::_LocaleUpdate(locale
__isalnum_l
__isdigit_l
__isspace_l
__isxdigit_l
_isalnum
_isdigit
_isspace
_isxdigit
__get_printf_count_output
_printf
_scanf



401830:A3->v9>v10->v15–>v17

401000: 检验字符

401090：对416050处理

4011a0:congraulations /try again

画红圈的地方注意result该在a3异或的下面

dbappsec的来源

```
#include < iostream>
using namespace std;
 int main(){
 int result;
string a2="dbappsec";
int a3=0;
  if ( a2[0] == 'd' )
  {
    a3 |= 4;
  }
  else
  {
    a3 ^= 3;
  }
  if ( a2[1] == 'b' )
  {
    a3 |= 0x14;
  }
  else
  {
    a3 &= 0x61;
  }
  if ( a2[2] == 'a' )
  {
    a3 |= 0x84;
  }
  else
  {
    a3 &= 0xA;
  }
  if ( a2[3] == 'p' )
  {
    a3 |= 0x114  ;
  }
  else
  {
    a3 >>= 7;
  }
  if ( a2[4] == 'p' )
  {
```

```
  {
    a3 |= 0x380;
  }
  else
  {
    a3 *= 2;
  }
 if ( a2[5] == 'f' )
    {
      a3 |= 0x2DC;

    }
  if ( a2[5] == 's' )
  {
    a3 |= 0xA04;
  }
  else{

    a3 ^= 0x1AD;
  }
  if ( a2[6] == 'e' )
  {
    a3 |= 0x2310;
  }
  else
  {
    a3 |= 0x4A;
  }
  if ( a2[7] == 'c' )
  { a3 |= 0x8A10;
    result = a3;
  }
  else
  {
    a3 &= 0x3A3;
    result = a3;
  }
cout<<hex<<result<<endl;
char x[8]={0x2a,0xd7,0x92,0xe9,0x53,0xe2,0xc4,0xcd};
for(int i=0;i<8;i++){
x[i]^=a2[i];
int pp=x[i]&0xff;
 cout<<pp;
}
}
```

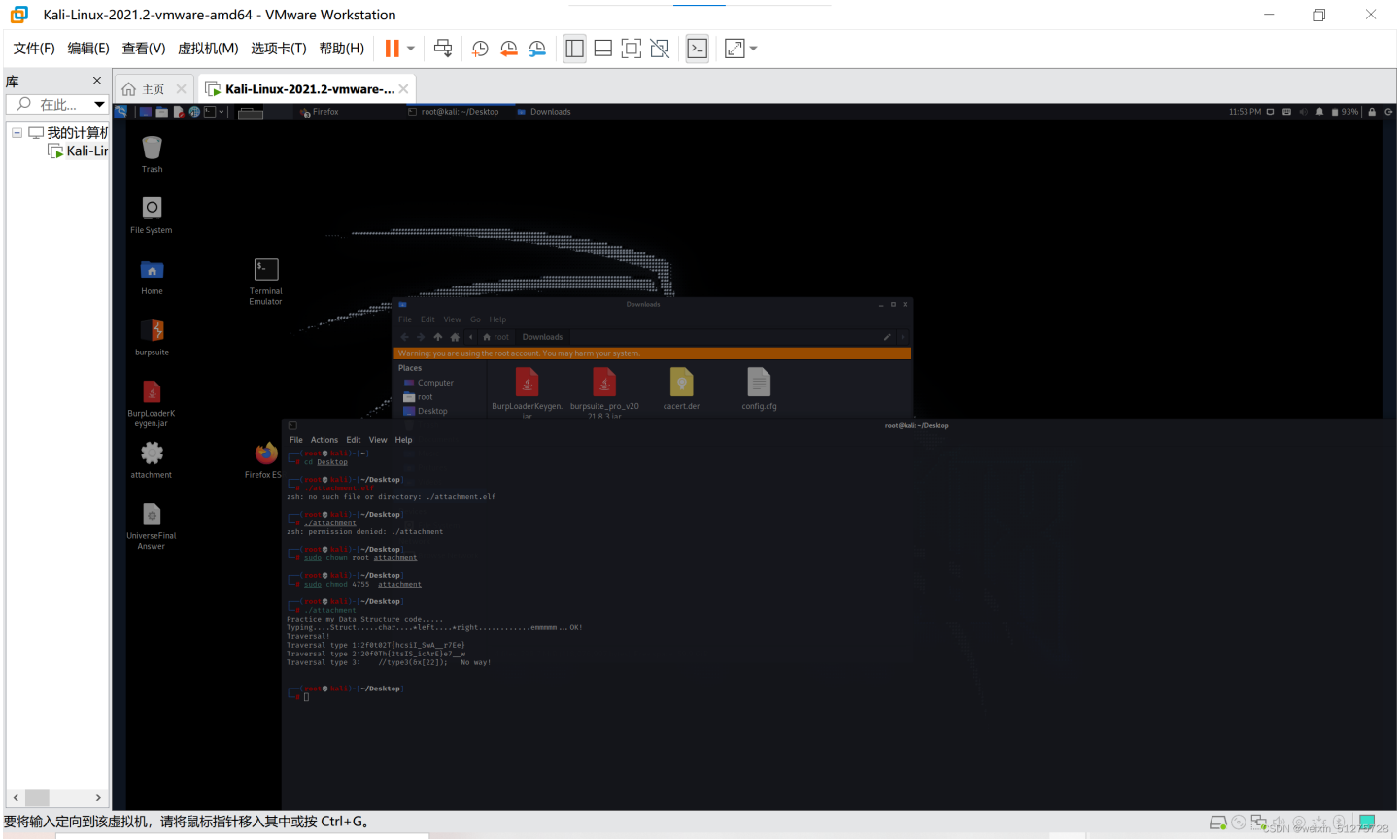
416050太复杂了 用动调发现反调试三个isdebuggerpresent 、一个isprocessorfeaturepresent。但strongod就可以不用修改

v15

最后md5

47 level4

47 level4 二叉树的三种遍历方式



20f0Th{2tsIS_icArE}e7__w 后序遍历结果

2f0t02T{hcsiI_SwA__r7Ee} 中序遍历结果

w： 2f0t02T{hcsiI_S A__r7we}

c： 2f0t02T{h siI_S

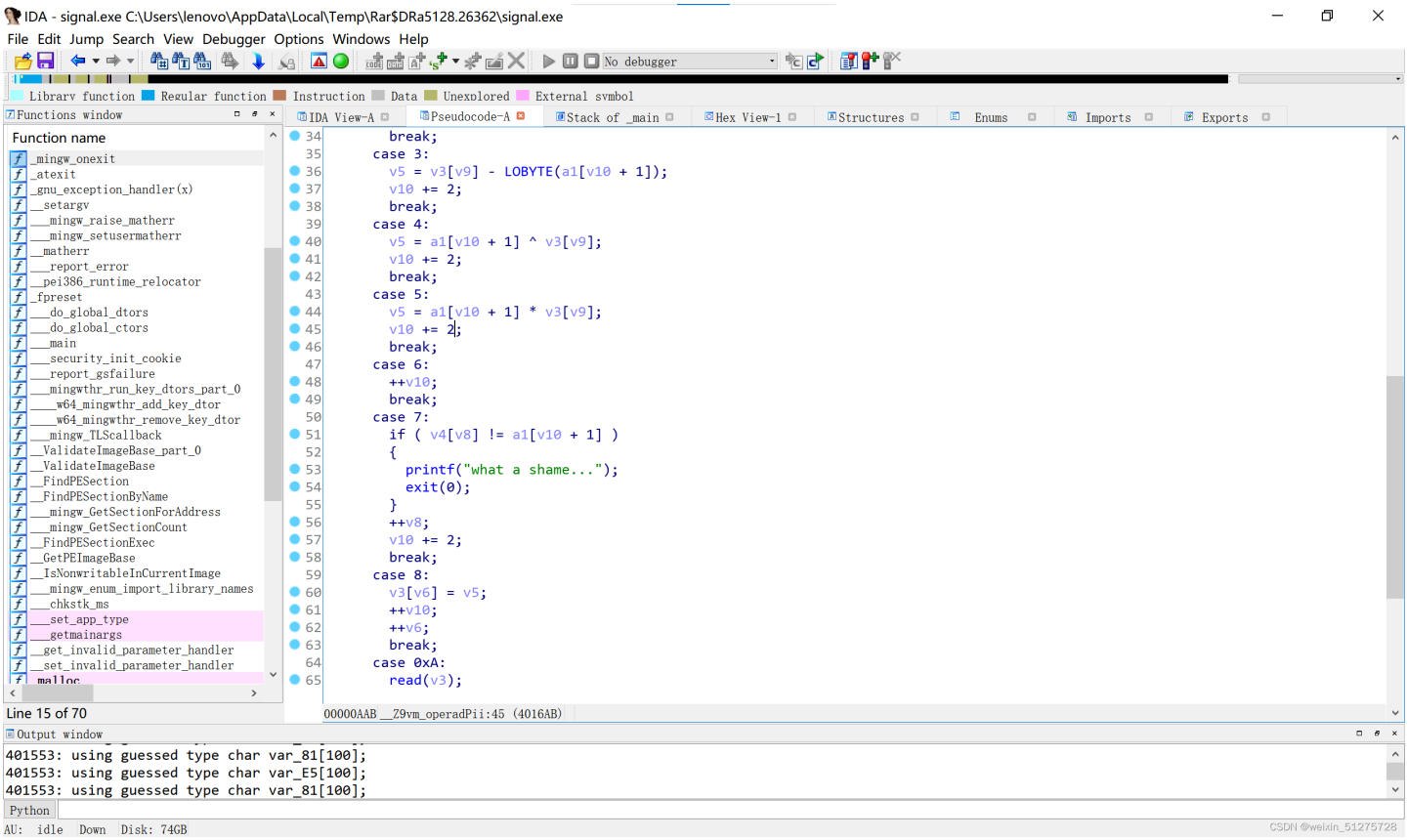
t：2f0 t02T{h

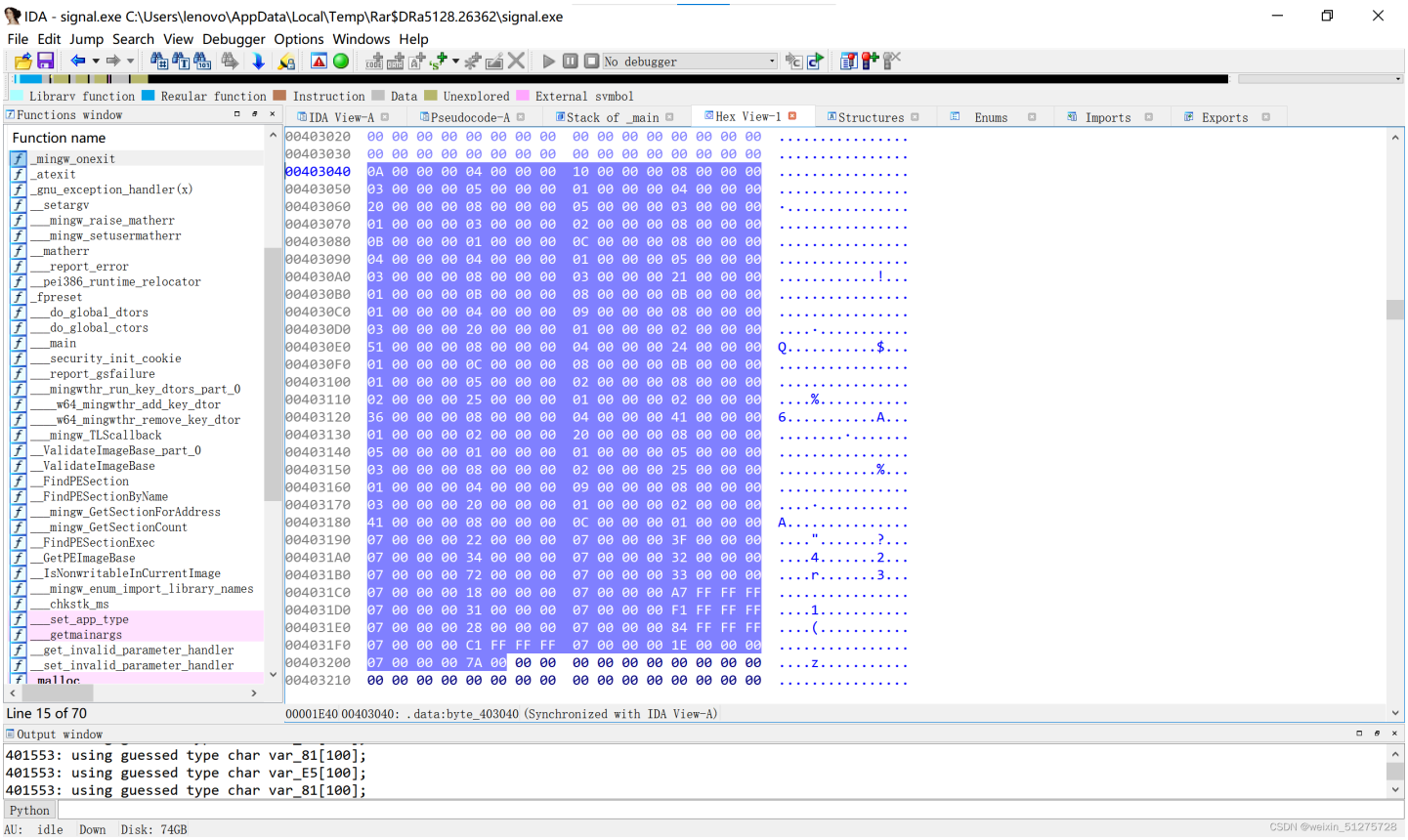
…这么一直分下去就有了

48 singal

48singal手算

a输入 然后进行运算方式及运算的数最后与07后面的数字比较。





```
#include < iostream>
using namespace std;
int main(){
char a[15];
a[0]^0x10-5=0x22;
a[1]^0x20*3=0x3f;
a[2]-2-1=0x34;
a[3]+1^4=0x32;
a[4]*3-0x21=0x72;
a[5]-2=0x33;
a[6]^9-0x20=0x18;
a[7]+0x51^0x24=0xa7;
a[8]=0x31;
a[9]*2+0x25=0xf1;
a[10]+0x36^0x41=0x28;
a[11]+0x20=0x84;
a[12]*3+0x25=0xc1;
a[13]*9-0x20=0x1e;
a[14]+0x42=0x7a;
cout<<a;
}
```