2021/12/12攻防世界reverse做题记录

原创

力扣没刷够五百道题不改名 于 2021-12-12 12:41:30 发布 2270 收藏

分类专栏： 攻防世界 reverse 网络安全学习 文章标签： 安全

本文链接：https://blog.csdn.net/weixin_56280430/article/details/121884579

攻防世界 同时被 3 个专栏收录

4 篇文章 0 订阅

订阅专栏

reverse

1 篇文章 0 订阅

订阅专栏

网络安全学习

2 篇文章 0 订阅

订阅专栏

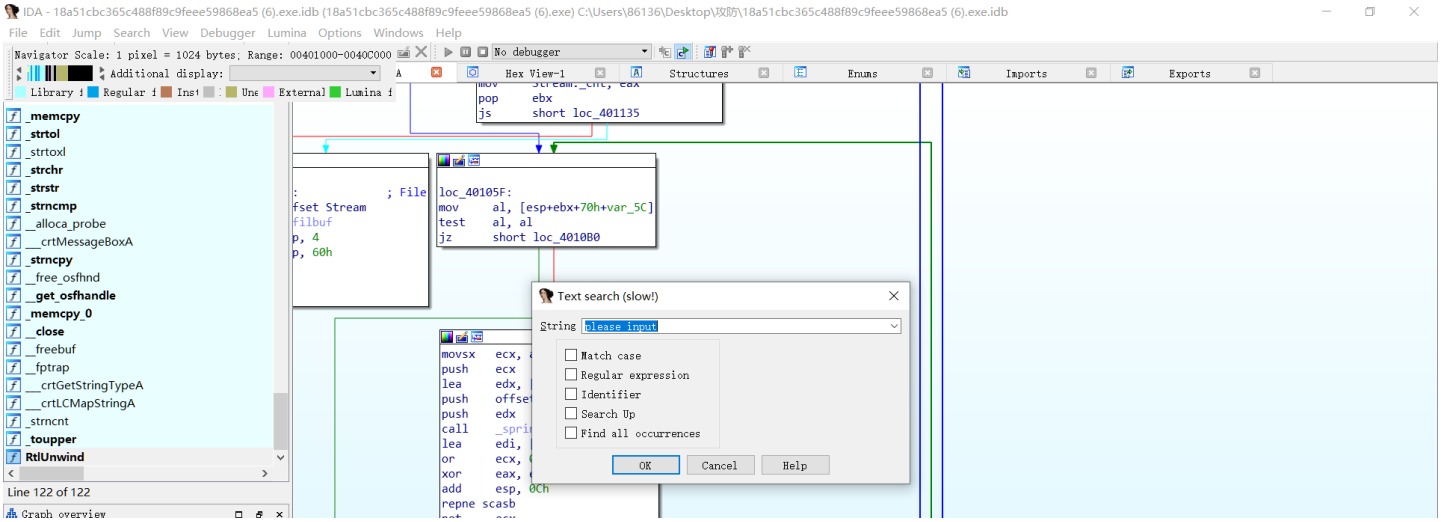
reverse

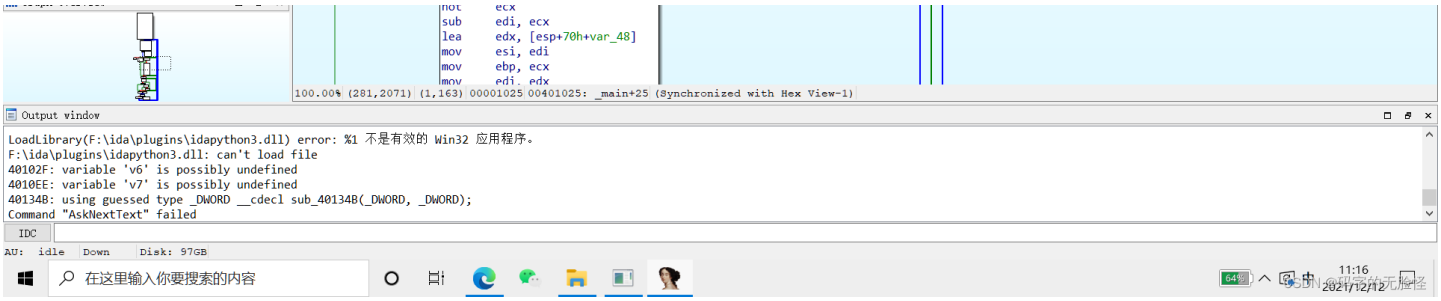
Hello, CTF

Hello, CTF

```
please input your serial:gjgc
wrong!
please input your serial:
```

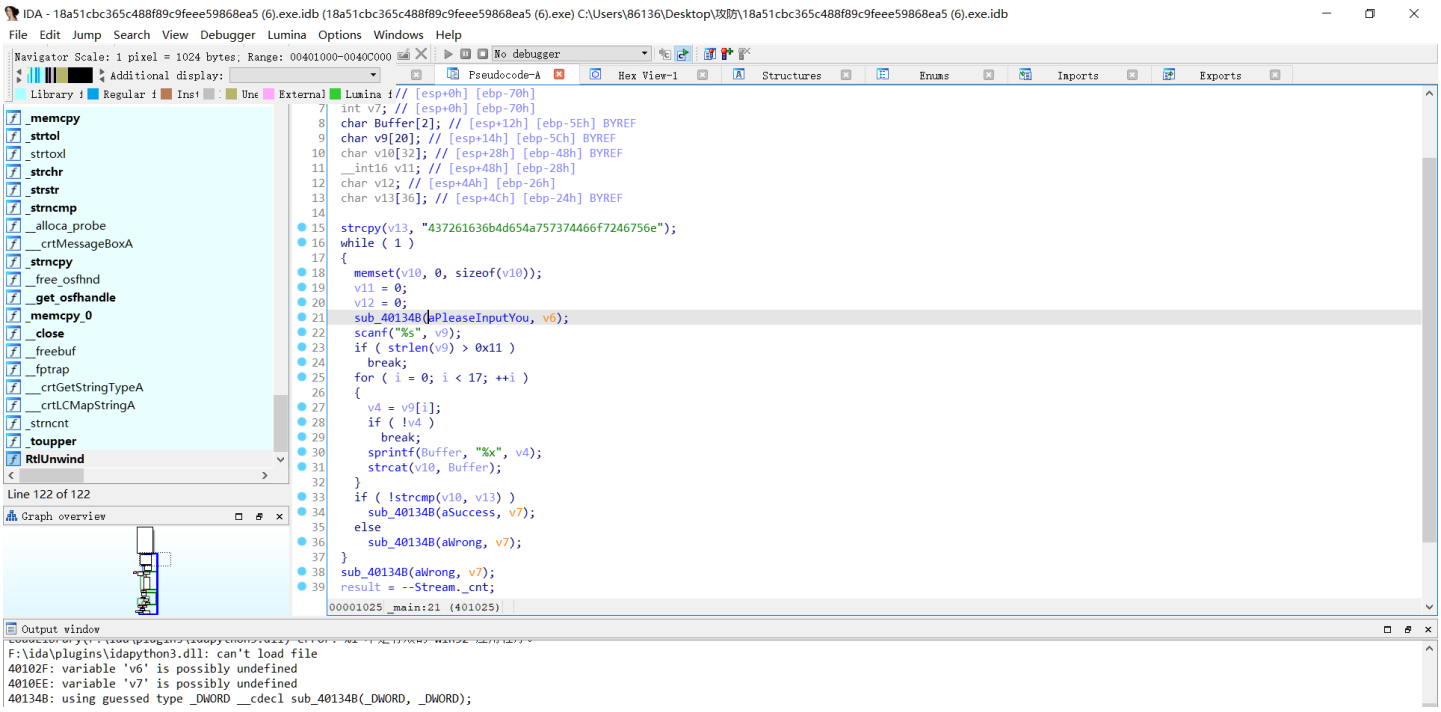
用idea打开，

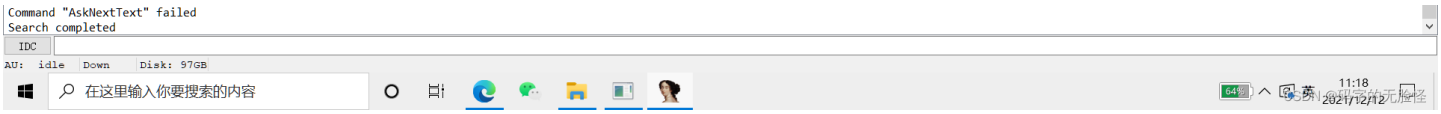




找命令行输入的please input

点击F5跳转





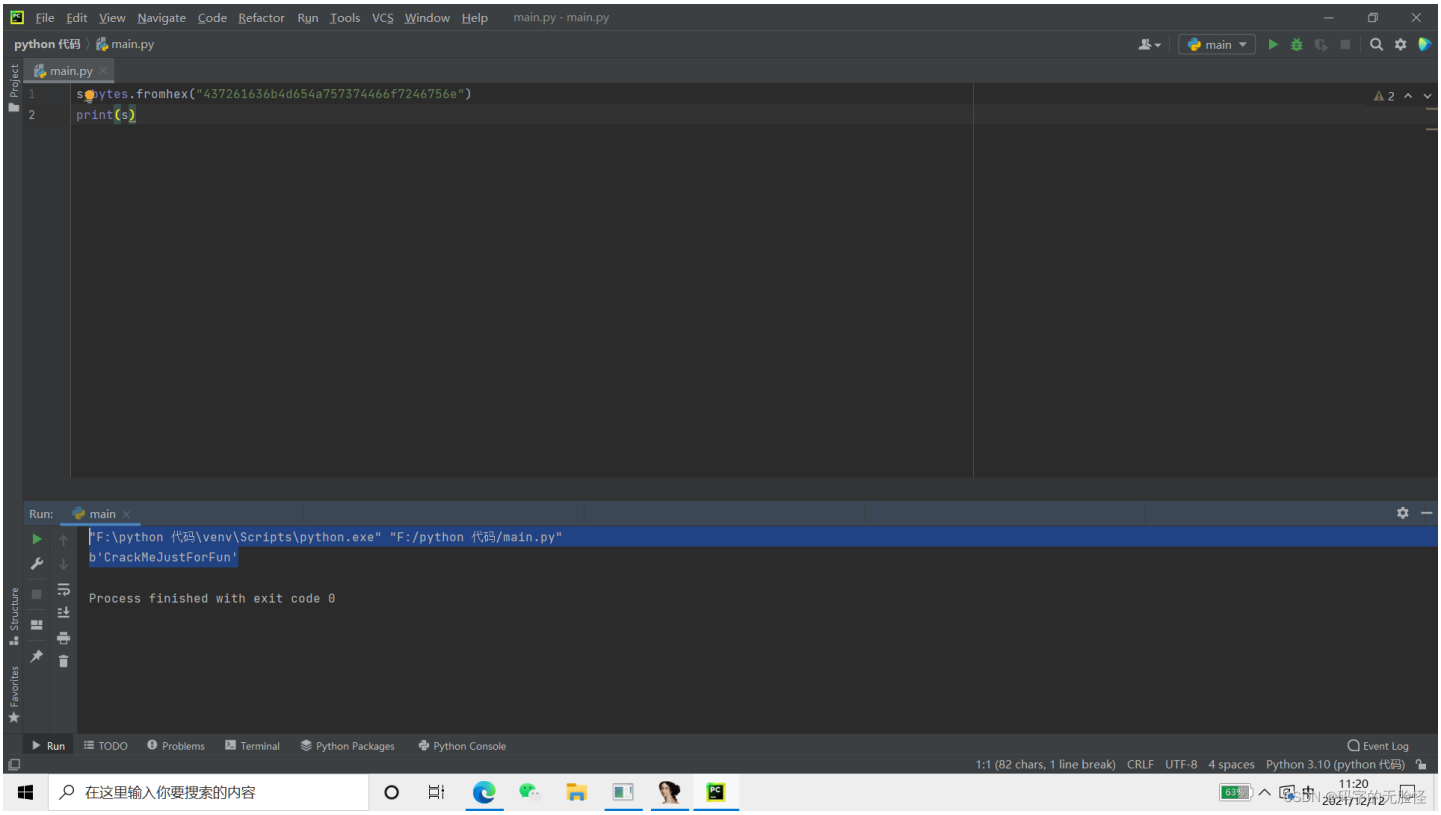
看网上说，这一段的意思是：大致逻辑为将用户输入的字符单个与v13字串单个进行比对，然后判断是否输入正确，v13对应的字串是16进制，直接用python转换过来即可

```
strcpy(v13, "437261636b4d654a757374466f7246756e");
while ( 1 )
{
  memset(v10, 0, sizeof(v10));
  v11 = 0;
  v12 = 0;
  sub_40134B(aPleaseInputYou, v6);
  scanf("%s", v9);
  if ( strlen(v9) > 0x11 )
    break;
  for ( i = 0; i < 17; ++i )
  {
    v4 = v9[i];
    if ( !v4 )
      break;
    sprintf(Buffer, "%x", v4);
    strcat(v10, Buffer);
  }
  if ( !strcmp(v10, v13) )
    sub_40134B(aSuccess, v7);
  else
    sub_40134B(aWrong, v7);
}
sub_40134B(aWrong, v7);
result = --Stream._cnt;
if ( Stream._cnt < 0 )
  return _filbuf(&Stream);
++Stream._ptr;
return result;
```

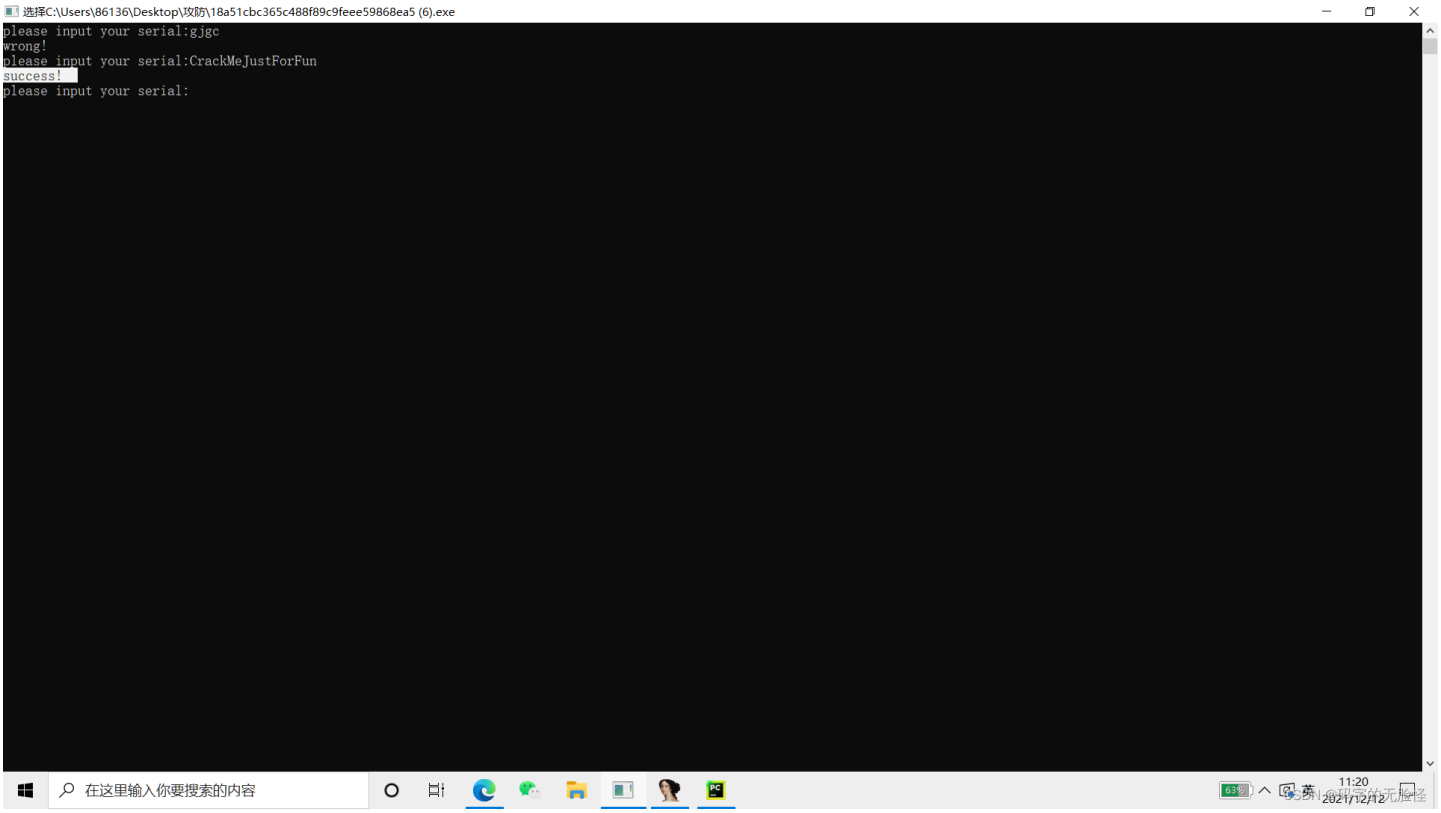

于是我从他们写的writeup的python脚本复制了过来（没办法，我还没有学python，什么都看不懂）

```
s=bytes.fromhex("437261636b4d654a757374466f7246756e")
print(s)
```

CrackMeJustForFun

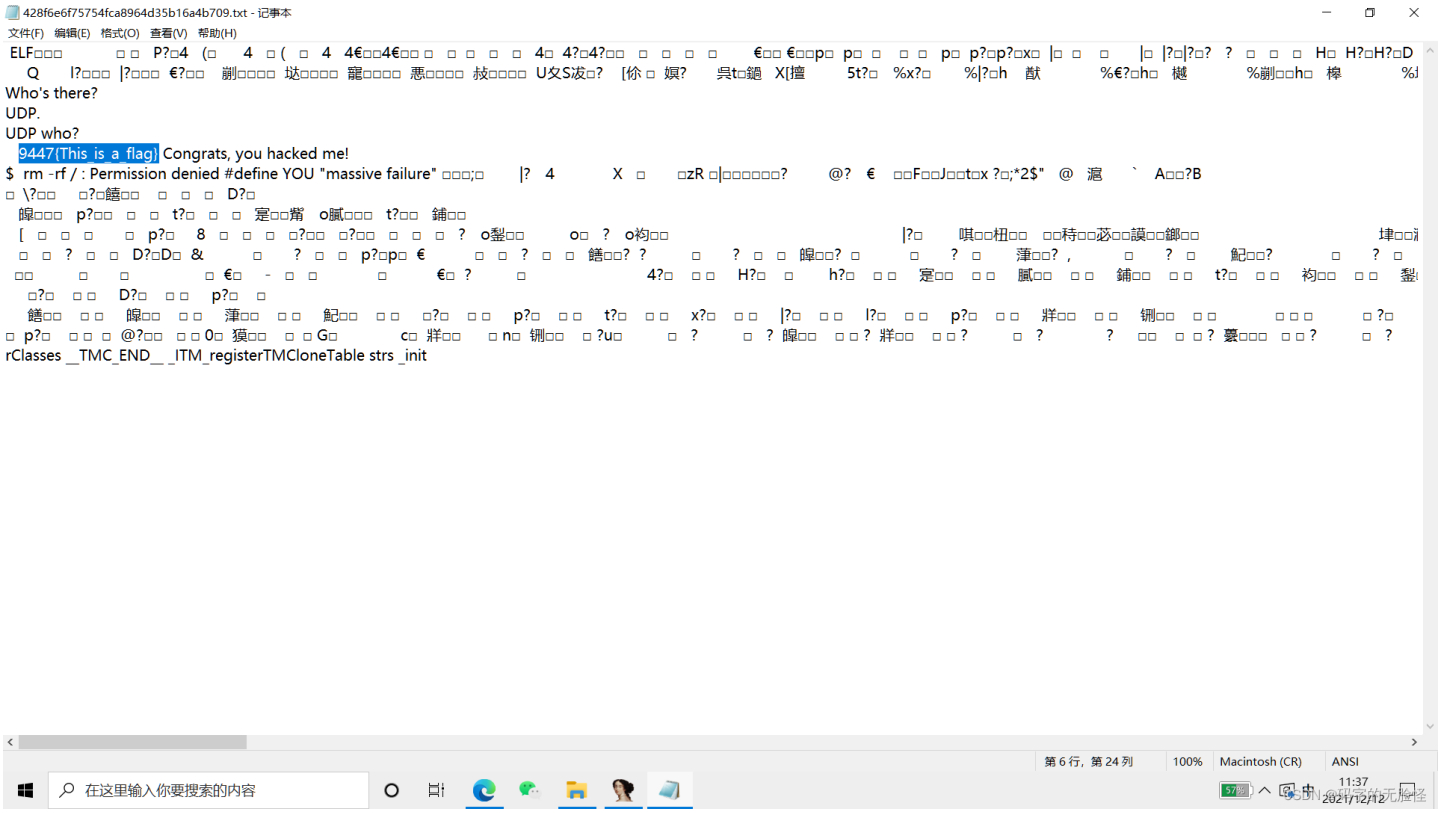


最后的答案是CrackMeJustForFun



insanity

我随便改成了txt后缀，看到了

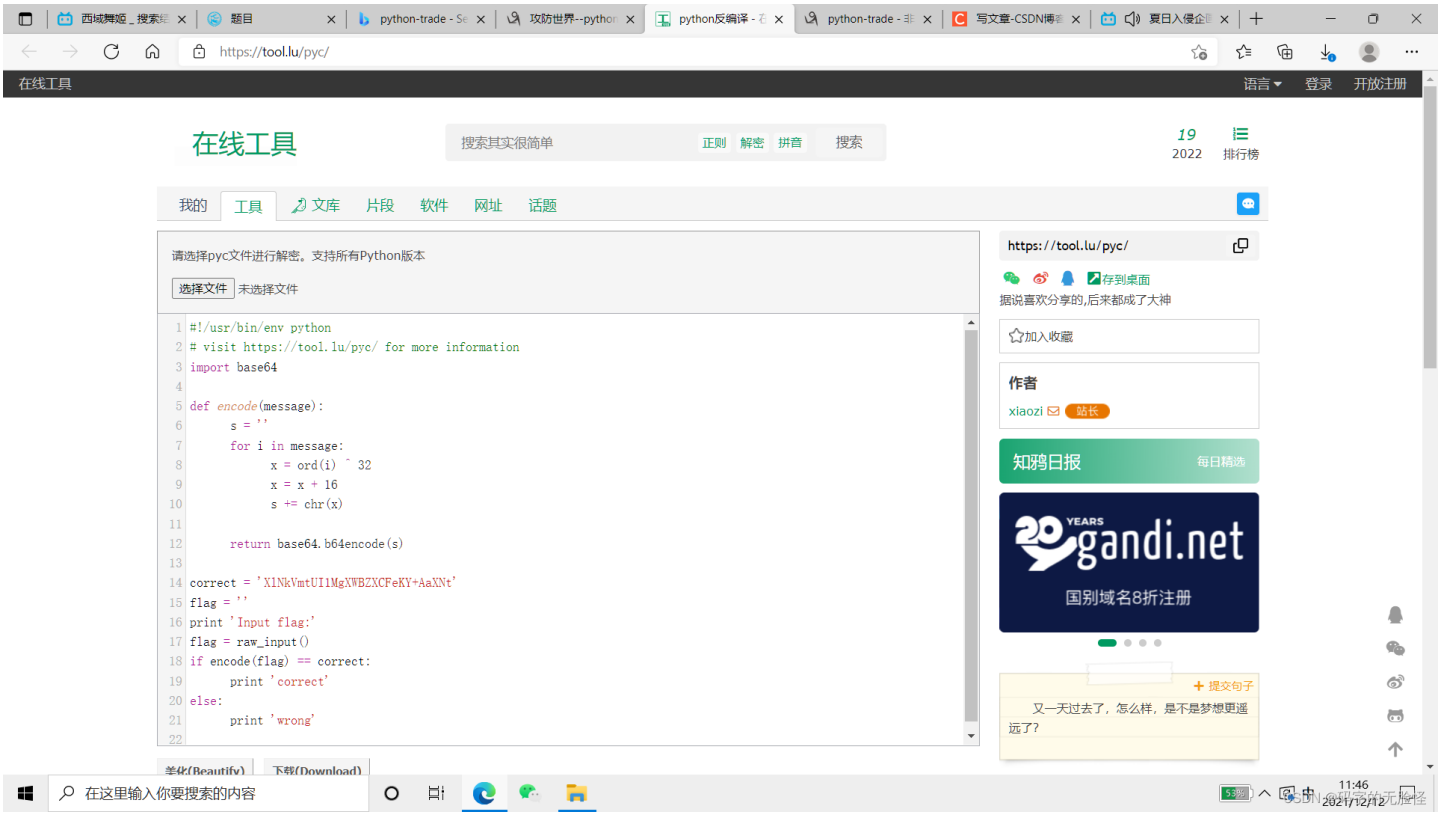


然后复制了过去9447{This_is_a_flag}

居然对了！

amazing

python-trade



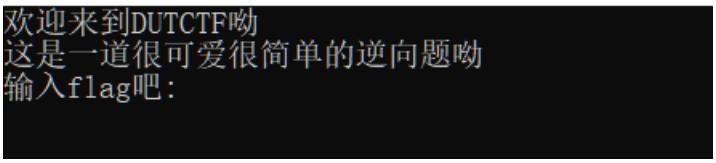
这里需要用到一个pyc文件反编译的工具，可以使用在线python 反编译

听说kali可以下载安装，用命令pip install uncompyle，用空自己去试试

得到这个XlNkVmtUI1MgXWBZXCFeKY+AaXNt

听他们说要反过来执行，搞了半天我也打不出代码，暂时放弃。

re1



用idea打开

按F5进入伪代码

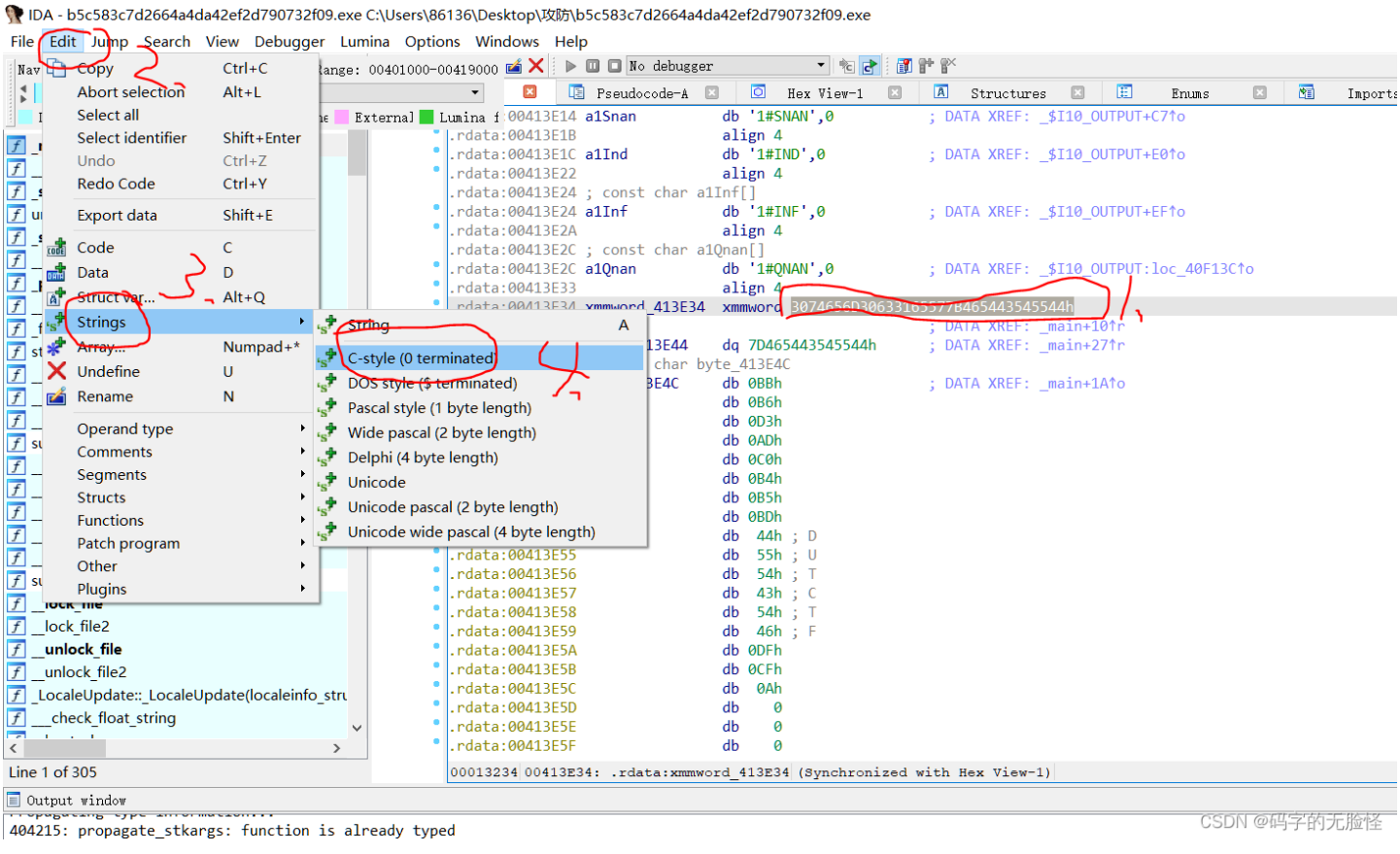
```
nal Lumina f l main(int argc, const char **argv, const char **envp)
{
  int v3; // eax
  __m128i v5; // [esp+0h] [ebp-44h] BYREF
  int v6; // [esp+1Ch] [ebp-28h]
  char v7[32]; // [esp+20h] [ebp-24h] BYREF

  v5 = _mm_loadu_si128((const __m128i *)&xmmword_413E34);
  LOWORD(v6) = 0;
  printf(&byte_413E4C, v5.m128i_i64[0], v5.m128i_i64[1], 1129600324, 8210004, 0, v6);
  printf(&byte_413E60);
  printf(&byte_413E80);
  scanf("%s", v7);
  v3 = strcmp(v5.m128i_i8, v7);
  if ( v3 )
    v3 = v3 < 0 ? -1 : 1;
  if ( v3 )
    printf(aFlag);
  else
    printf(aFlagGet);
```

```
    printf(aFlagGet);
    system("pause");
    return 0;
}
```


点击红框的，按确认键

```
.rdata:00413E1B                 align 4
.rdata:00413E1C a1Ind           db '1#IND',0            ; DATA XREF: _$I10_OUTPUT+E0↑o
.rdata:00413E22                 align 4
.rdata:00413E24 ; const char a1Inf[]
.rdata:00413E24 a1Inf           db '1#INF',0            ; DATA XREF: _$I10_OUTPUT+EF↑o
.rdata:00413E2A                 align 4
.rdata:00413E2C ; const char a1Qnan[]
.rdata:00413E2C a1Qnan          db '1#QNAN',0           ; DATA XREF: _$I10_OUTPUT:loc_40F13C↑o
.rdata:00413E33                 align 4
.rdata:00413E34 xmmword_413E34  xmmword 3074656D30633165577B465443545544h
.rdata:00413E34                                         ; DATA XREF: _main+10↑r
.rdata:00413E44 qword_413E44    dq 7D465443545544h      ; DATA XREF: _main+27↑r
.rdata:00413E4C ; const char byte_413E4C
.rdata:00413E4C byte_413E4C     db 0BBh                 ; DATA XREF: _main+1A↑o
.rdata:00413E4D                 db 0B6h
.rdata:00413E4E                 db 0D3h
.rdata:00413E4F                 db 0ADh
.rdata:00413E50                 db 0C0h
.rdata:00413E51                 db 0B4h
.rdata:00413E52                 db 0B5h
.rdata:00413E53                 db 0BDh
.rdata:00413E54                 db  44h ; D
.rdata:00413E55                 db  55h ; U
.rdata:00413E56                 db  54h ; T
.rdata:00413E57                 db  43h ; C
.rdata:00413E58                 db  54h ; T
.rdata:00413E59                 db  46h ; F
.rdata:00413E5A                 db 0DFh
.rdata:00413E5B                 db 0CFh
.rdata:00413E5C                 db  0Ah
.rdata:00413E5D                 db    0
.rdata:00413E5E                 db    0
.rdata:00413E5F                 db    0
00013234 00413E34: .rdata:xmmword_413E34 (Synchronized with Hex View-1)
```


按如图步骤进行，转换为ascill码。



得到解码，去试试DUTCTF{We1c0met0DUTCTF}

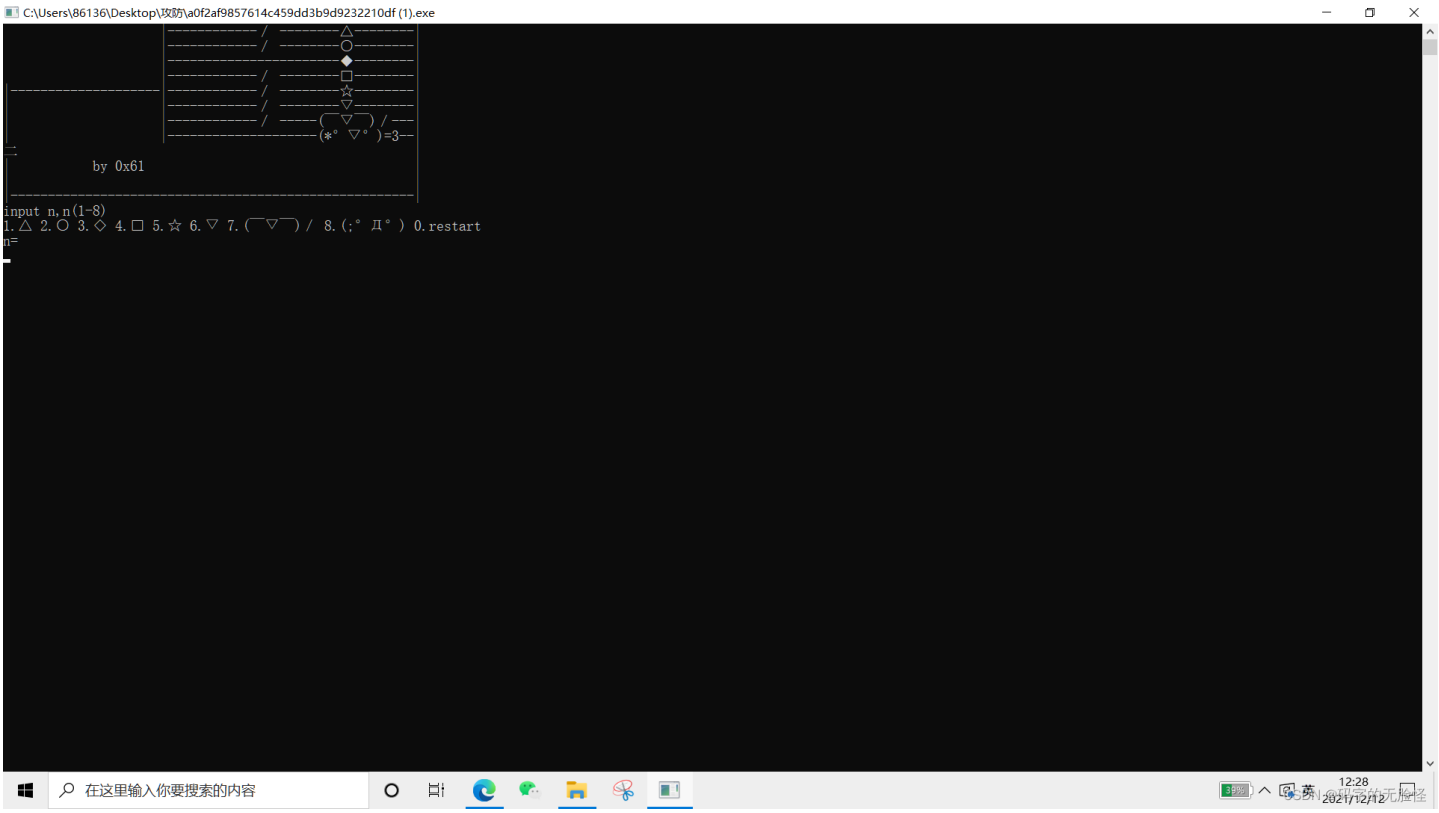
```
Lumina f 00413E14 a1Snan          db  1#SNAN ,0          ; DATA XREF: _$I10_OUTPUT+C7↑o
.rdata:00413E1B                   align 4
.rdata:00413E1C a1Ind             db '1#IND',0            ; DATA XREF: _$I10_OUTPUT+E0↑o
.rdata:00413E22                   align 4
.rdata:00413E24 ; const char a1Inf[]
.rdata:00413E24 a1Inf             db '1#INF',0            ; DATA XREF: _$I10_OUTPUT+EF↑o
.rdata:00413E2A                   align 4
.rdata:00413E2C ; const char a1Qnan[]
.rdata:00413E2C a1Qnan            db '1#QNAN',0           ; DATA XREF: _$I10_OUTPUT:loc_40F13C↑o
.rdata:00413E33                   align 4
.rdata:00413E34 aDutctfWe1c0met db 'DUTCTF{We1c0met0DUTCTF}',0
.rdata:00413E34                                           ; DATA XREF: _main+10↑r
.rdata:00413E4C ; const char byte_413E4C
.rdata:00413E4C byte_413E4C       db 0BBh                 ; DATA XREF: _main+1A↑o
.rdata:00413E4D                   db 0B6h
.rdata:00413E4E                   db 0D3h
.rdata:00413E4F                   db 0ADh
.rdata:00413E50                   db 0C0h
.rdata:00413E51                   db 0B4h
.rdata:00413E52                   db 0B5h
.rdata:00413E53                   db 0BDh
.rdata:00413E54                   db  44h ; D
.rdata:00413E55                   db  55h ; U
.rdata:00413E56                   db  54h ; T
.rdata:00413E57                   db  43h ; C
.rdata:00413E58                   db  54h ; T
.rdata:00413E59                   db  46h ; F
.rdata:00413E5A                   db 0DFh
.rdata:00413E5B                   db 0CFh
.rdata:00413E5C                   db  0Ah
.rdata:00413E5D                   db    0
.rdata:00413E5E                   db    0
.rdata:00413E5F                   db    0
.rdata:00413E60 ; const char byte_413E60
00013234 00413E34: .rdata:aDutctfWe1c0met (Synchronized with Hex View-1)
```



```
欢迎来到DUTCTF呦
这是一道很可爱很简单的逆向题呦
输入flag吧:DUTCTF{We1c0met0DUTCTF}
flag get √
请按任意键继续. . .
```

game

这个题目奇奇怪怪



先用idea打开试试
找flag一脸茫然

好了，先不写，下次写，现在没空，然后小结一下：
第一次写逆向的题目，感受是（1）每次打开都是这种命令行，可以检验你找的flag是否正确
（2）打开idea，找到main或者题目给的需要你输入什么，然后按F5进入到伪代码，看这个代码是如何得到flag的，可能falg就藏匿在里面的某个变量，然后你就开始找，在string里面搜索，或者什么样，感觉这个经常会用到python才可以看懂
（3）一般来说他的flag，不是什么falg{}这样子的格式，所以说找不到flag格式的也正正常，多做做说不定就可以碰巧蒙对，像我第二个直接用txt打开，有一个长一点的又比较像信息的直接正确了。
（4）感觉一般比较常用到的工具是idea（还好上课用过），python，好像说olldybge也能用，下次试试。