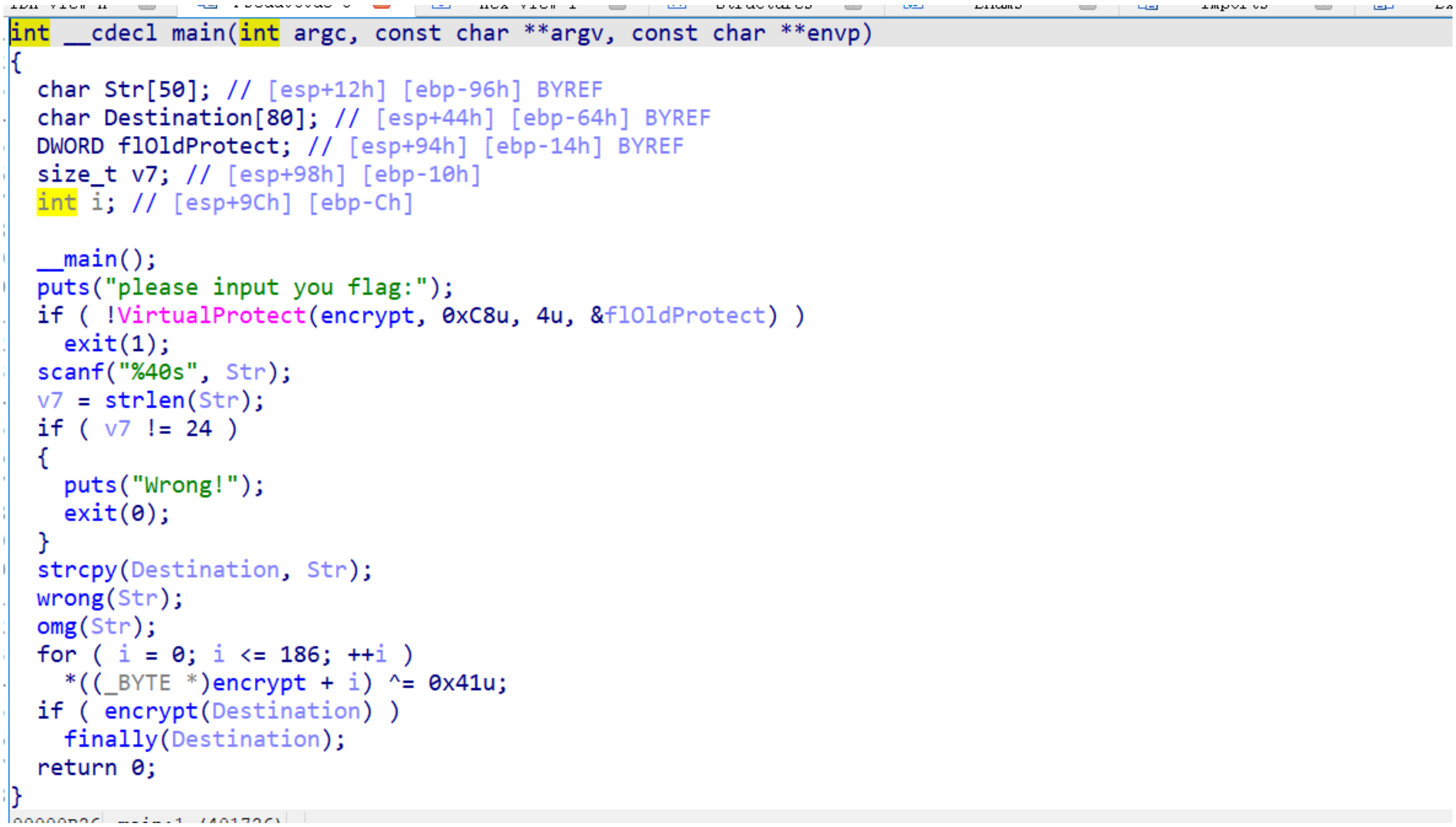
什么脑洞题，太屎了



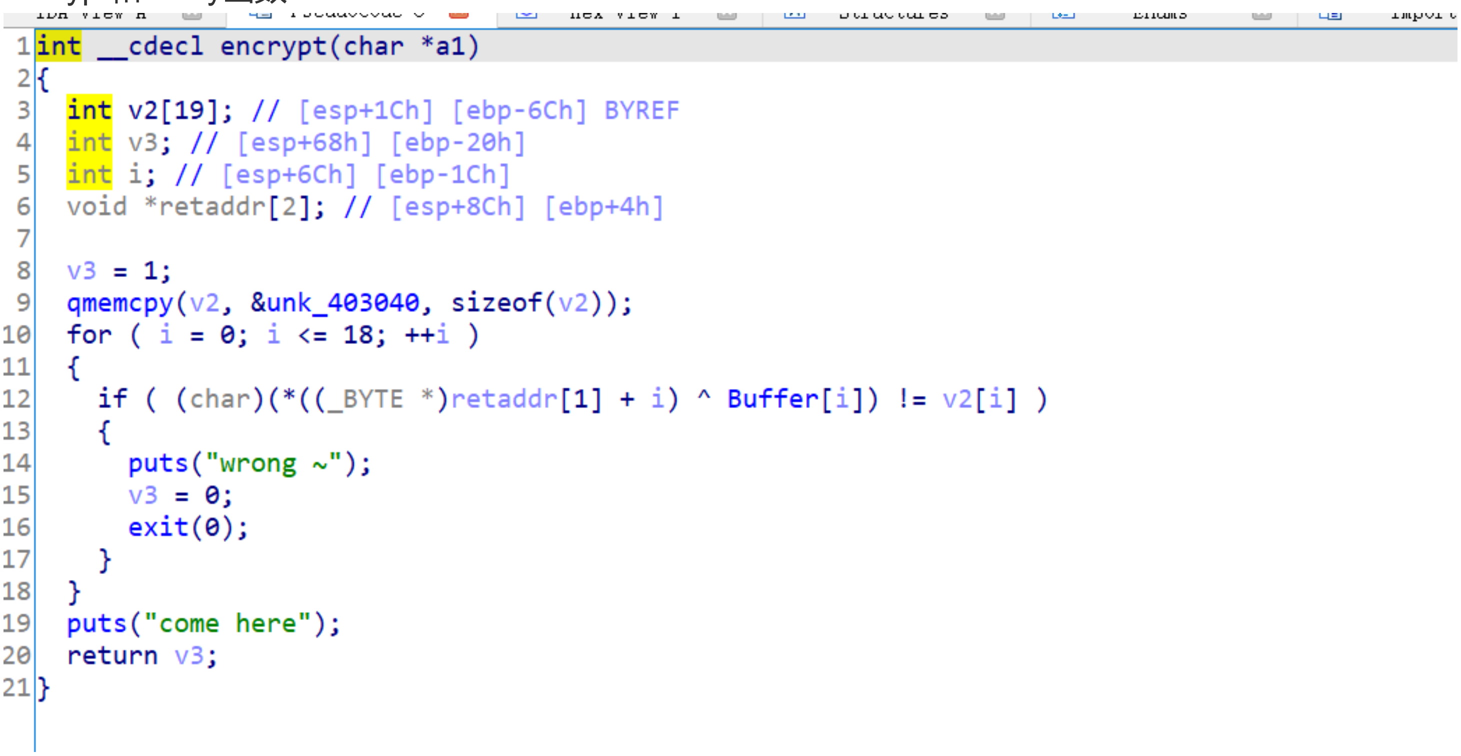
```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char Str[50]; // [esp+12h] [ebp-96h] BYREF
  char Destination[80]; // [esp+44h] [ebp-64h] BYREF
  DWORD flOldProtect; // [esp+94h] [ebp-14h] BYREF
  size_t v7; // [esp+98h] [ebp-10h]
  int i; // [esp+9Ch] [ebp-Ch]

  __main();
  puts("please input you flag:");
  if ( !VirtualProtect(encrypt, 0xC8u, 4u, &flOldProtect) )
    exit(1);
  scanf("%40s", Str);
  v7 = strlen(Str);
  if ( v7 != 24 )
  {
    puts("Wrong!");
    exit(0);
  }
  strcpy(Destination, Str);
  wrong(Str);
  omg(Str);
  for ( i = 0; i <= 186; ++i )
    *((_BYTE *)encrypt + i) ^= 0x41u;
  if ( encrypt(Destination) )
    finally(Destination);
  return 0;
}
```

wrong和omg都是假的，是屎

重点在smc后

encrypt和finally函数



```
int __cdecl encrypt(char *a1)
{
  int v2[19]; // [esp+1Ch] [ebp-6Ch] BYREF
  int v3; // [esp+68h] [ebp-20h]
  int i; // [esp+6Ch] [ebp-1Ch]
  void *retaddr[2]; // [esp+8Ch] [ebp+4h]

  v3 = 1;
  qmemcpy(v2, &unk_403040, sizeof(v2));
  for ( i = 0; i <= 18; ++i )
  {
    if ( (char)(*((_BYTE *)retaddr[1] + i) ^ Buffer[i]) != v2[i] )
    {
      puts("wrong ~");
      v3 = 0;
      exit(0);
    }
  }
  puts("come here");
  return v3;
}
```

这里能得到flag的前19个字符串

然而

后面就不行了

要靠猜

很离谱

因为最后一个字符是'}'所以才异或了71