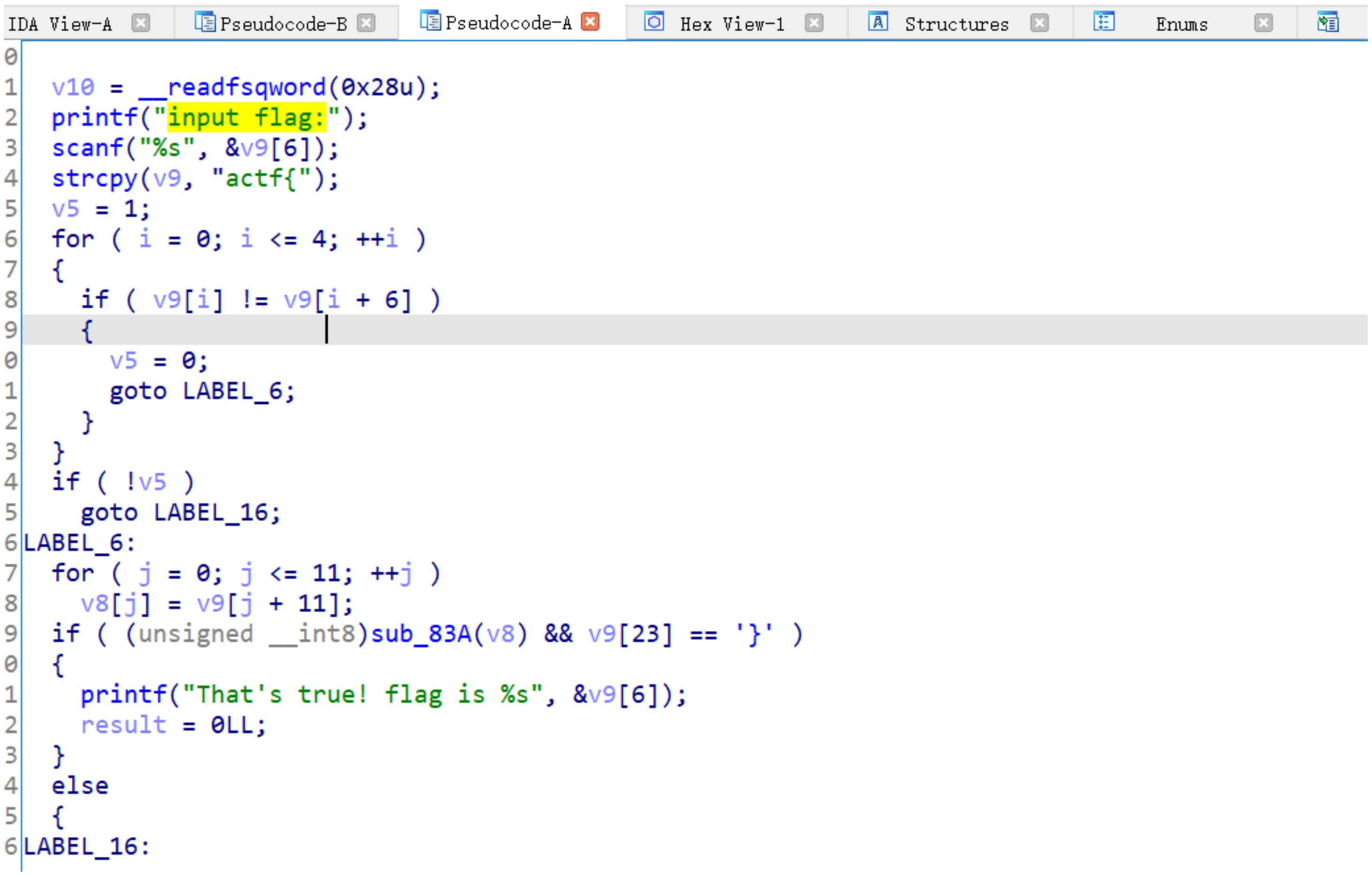
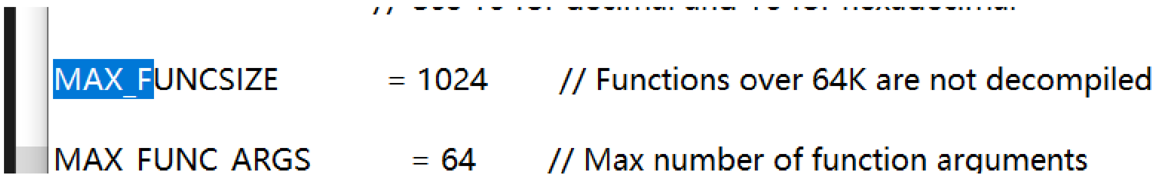
学到许多的re题目



```
  v10 = __readfsqword(0x28u);
  printf("input flag:");
  scanf("%s", &v9[6]);
  strcpy(v9, "actf{");
  v5 = 1;
  for ( i = 0; i <= 4; ++i )
  {
    if ( v9[i] != v9[i + 6] )
    {
      v5 = 0;
      goto LABEL_6;
    }
  }
  if ( !v5 )
    goto LABEL_16;
LABEL_6:
  for ( j = 0; j <= 11; ++j )
    v8[j] = v9[j + 11];
  if ( (unsigned __int8)sub_83A(v8) && v9[23] == '}' )
  {
    printf("That's true! flag is %s", &v9[6]);
    result = 0LL;
  }
  else
  {
LABEL_16:
```

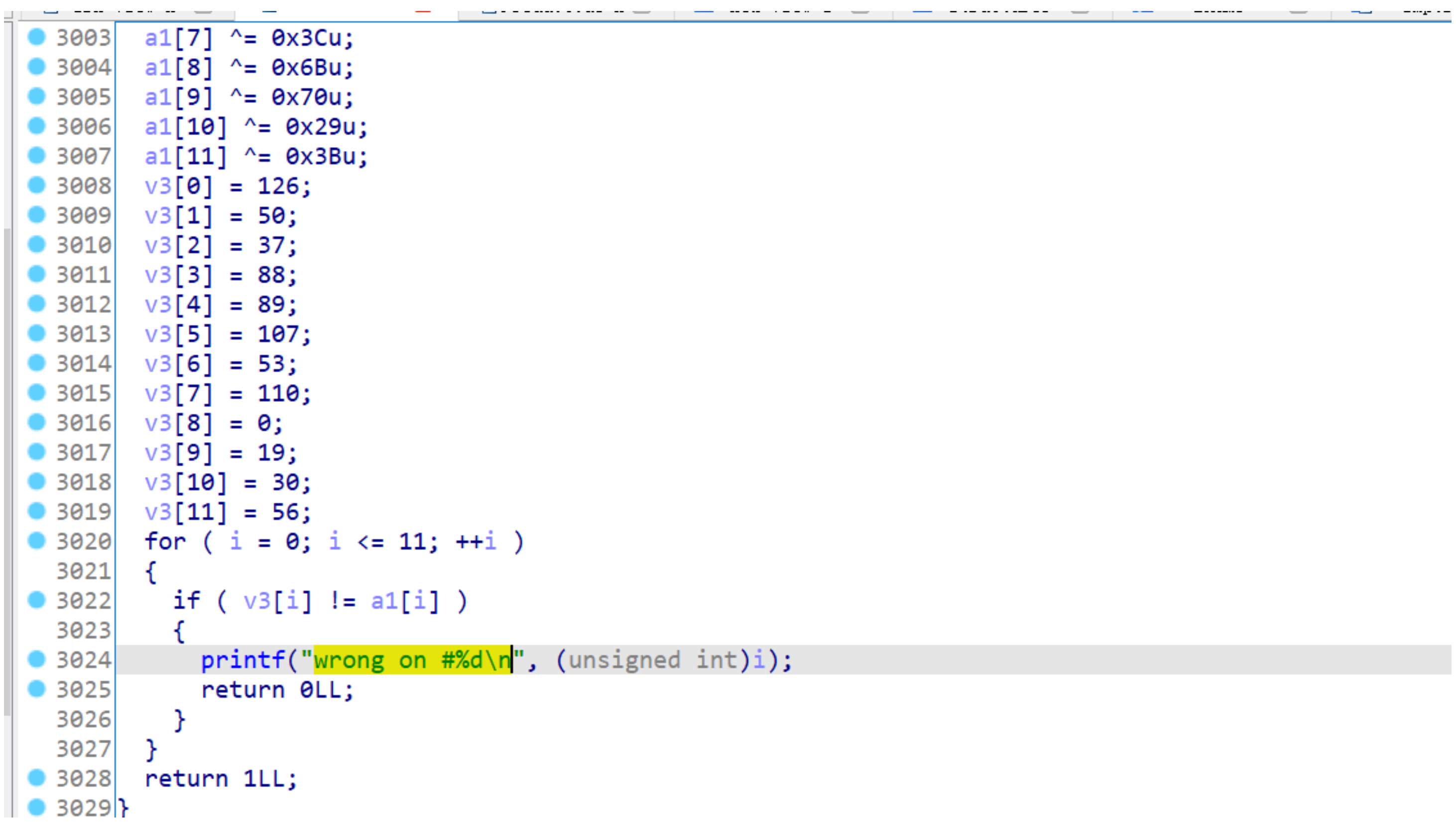
首先看这个
逻辑很清晰
很快就能明确到sub_83A是关键函数。
一开始因为sub_83A函数太大了，需要进入ida目录下的cfg目录中的MAX_FUNCSIZE从64改为1024



```
MAX_FUNCSIZE          = 1024      // Functions over 64K are not decompiled

MAX_FUNC_ARGS          = 64       // Max number of function arguments
```

然后可以编译了



```
  a1[7] ^= 0x3Cu;
  a1[8] ^= 0x6Bu;
  a1[9] ^= 0x70u;
  a1[10] ^= 0x29u;
  a1[11] ^= 0x3Bu;
  v3[0] = 126;
  v3[1] = 50;
  v3[2] = 37;
  v3[3] = 88;
  v3[4] = 89;
  v3[5] = 107;
  v3[6] = 53;
  v3[7] = 110;
  v3[8] = 0;
  v3[9] = 19;
  v3[10] = 30;
  v3[11] = 56;
  for ( i = 0; i <= 11; ++i )
  {
    if ( v3[i] != a1[i] )
    {
      printf("wrong on #%d\n", (unsigned int)i);
      return 0LL;
    }
  }
  return 1LL;
}
```

可以看到，第几个字符错误，就会告诉你第几个字符出错了，这时候爆破就很有效

```
from pwn import *
import re
flag = 'actf{'
for i in range(12):
    for j in range(33,127):
    test_flag = flag
    io = process('./SoulLike')
    test_flag += chr(j)
    if i == 11:
        test_flag += '}'
    print(test_flag)
    io.sendline(test_flag)
    msg = io.recvuntil('\n')
    r = re.findall('wrong on #(.*?)\n',msg)
    r = r[0]
    r = int(r)
    io.close()
    if r == i:
        continue
    else:
        flag += chr(j)
        break
```

原来pwntools还能这样用，实在是涨姿势了啊