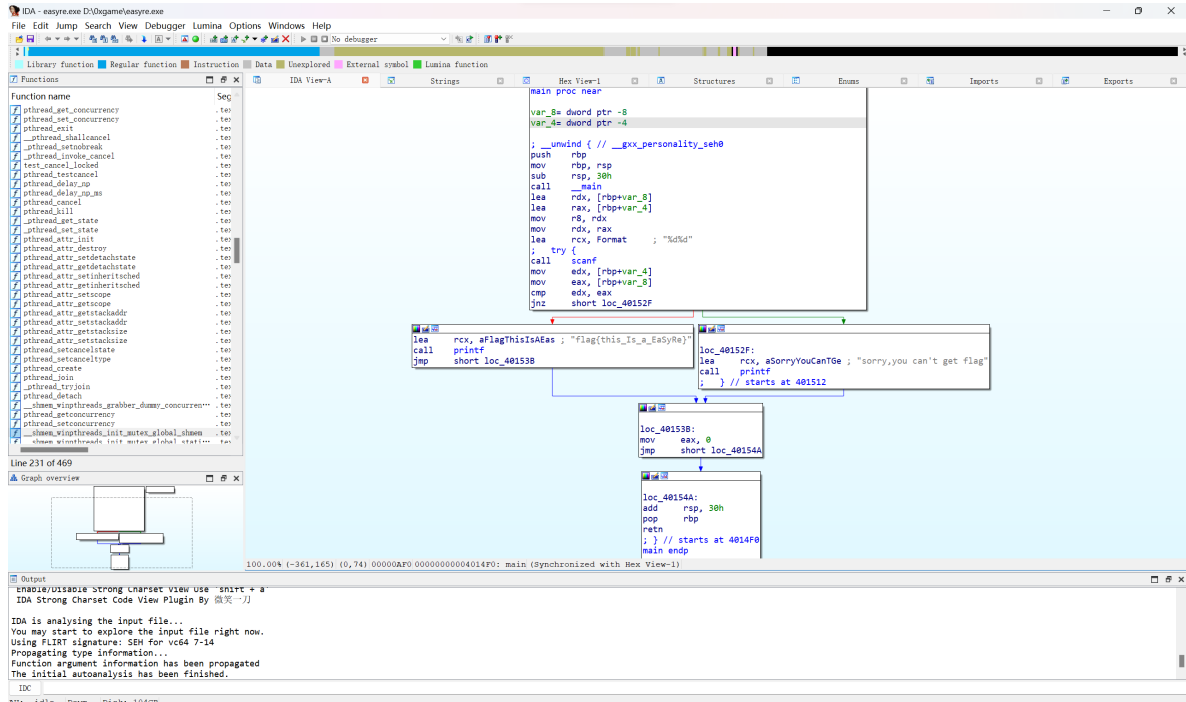
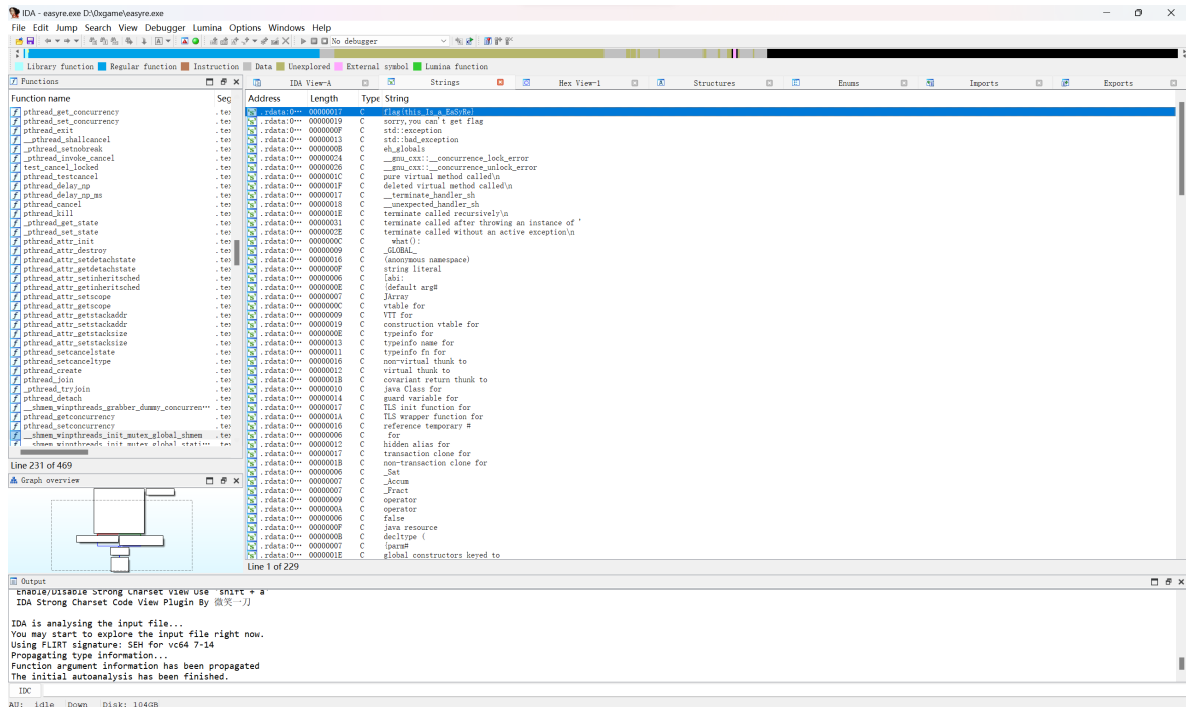


— easyre

1.查壳，放入ida64



2.shift+f12调出字符串，得到答案 **flag{this_Is_a_EaSyRe}**

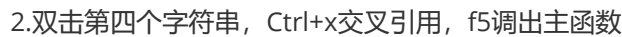


```

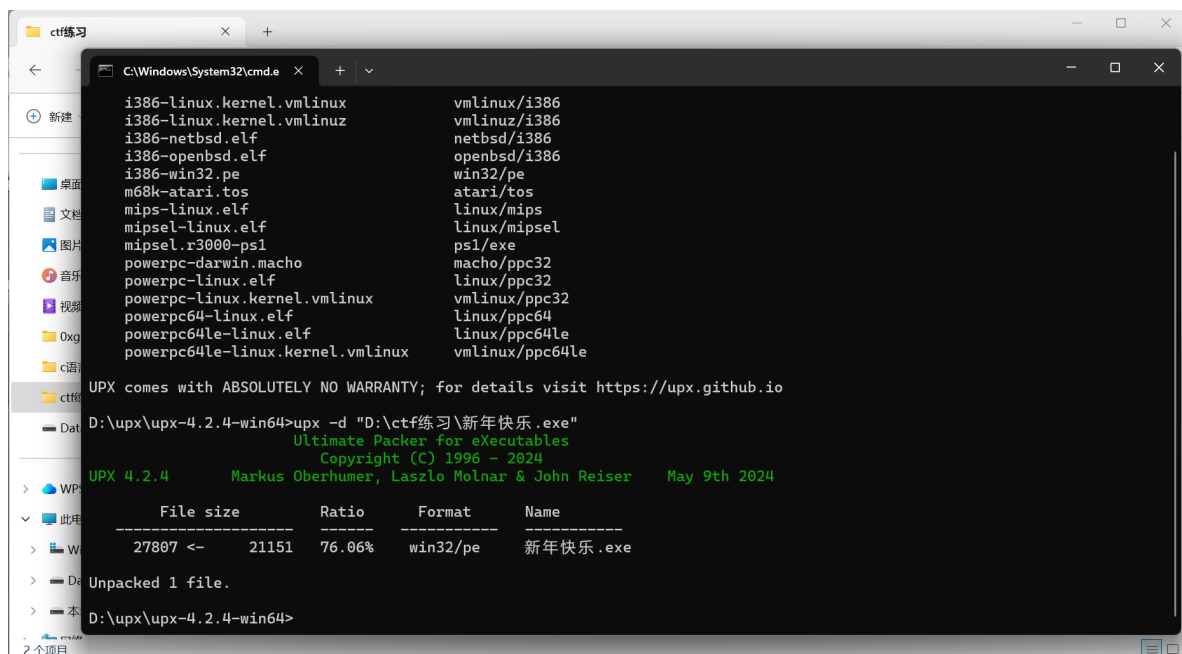
= reserve1

```

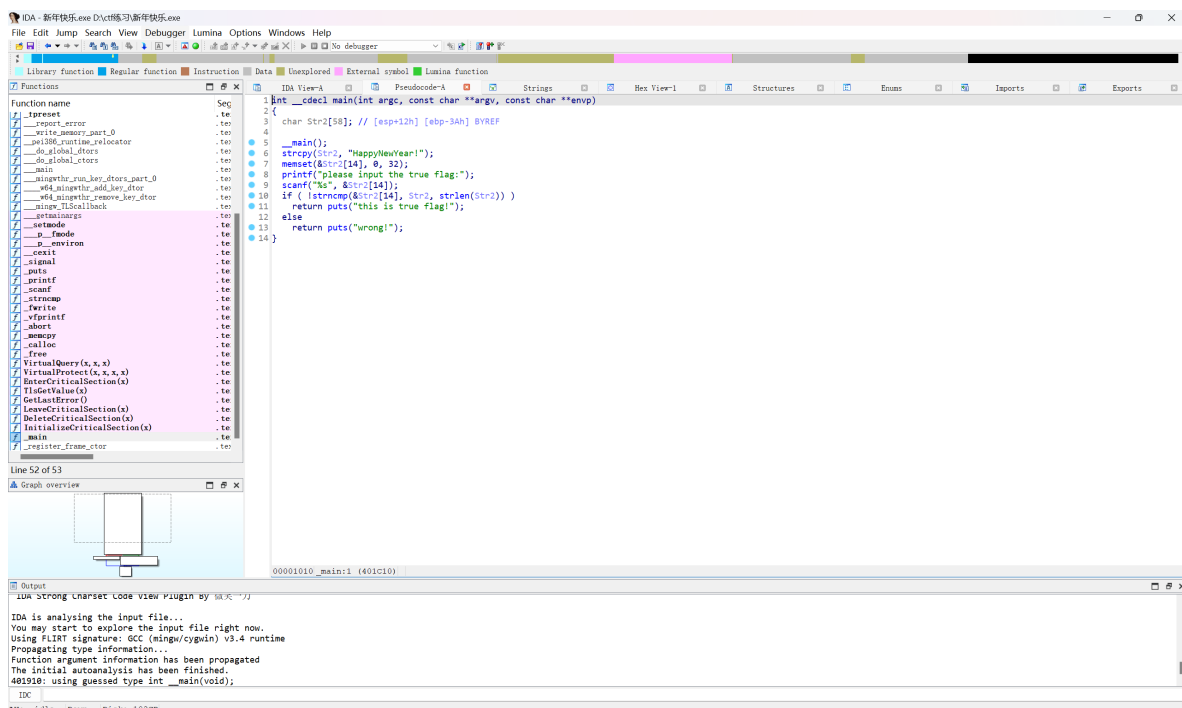
1.查壳，放入ida64，shift+f12调出字符串







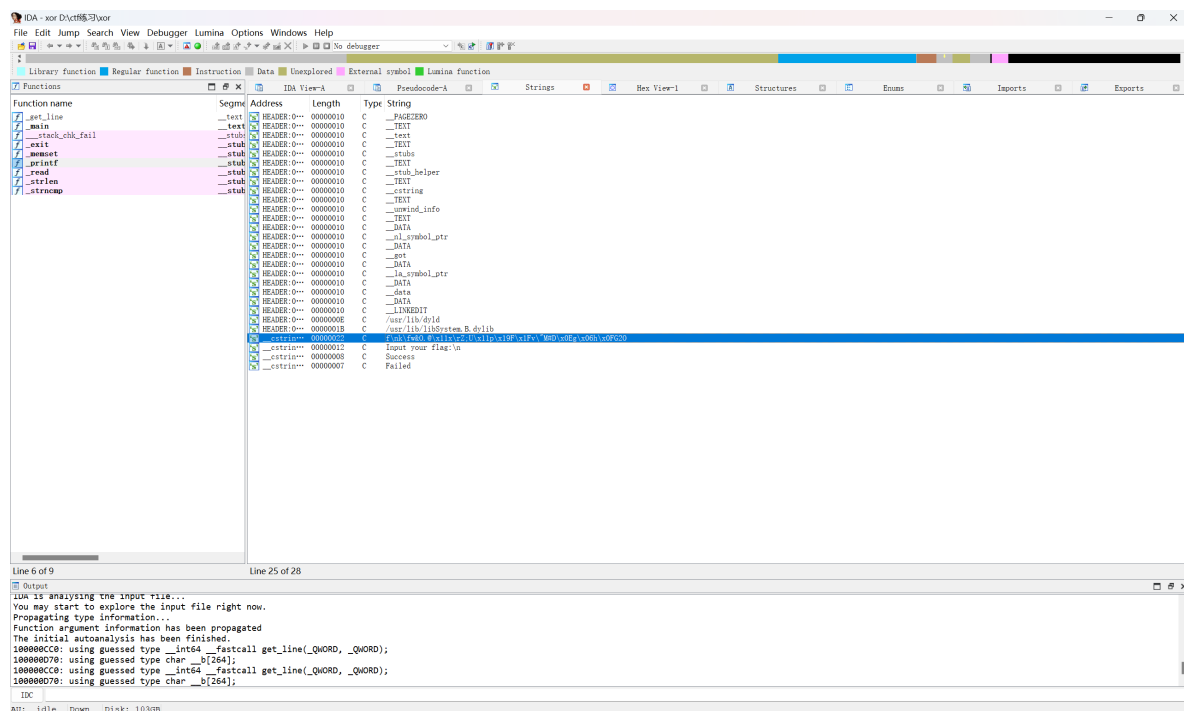
2.放入ida32，找到主函数，f5反汇编



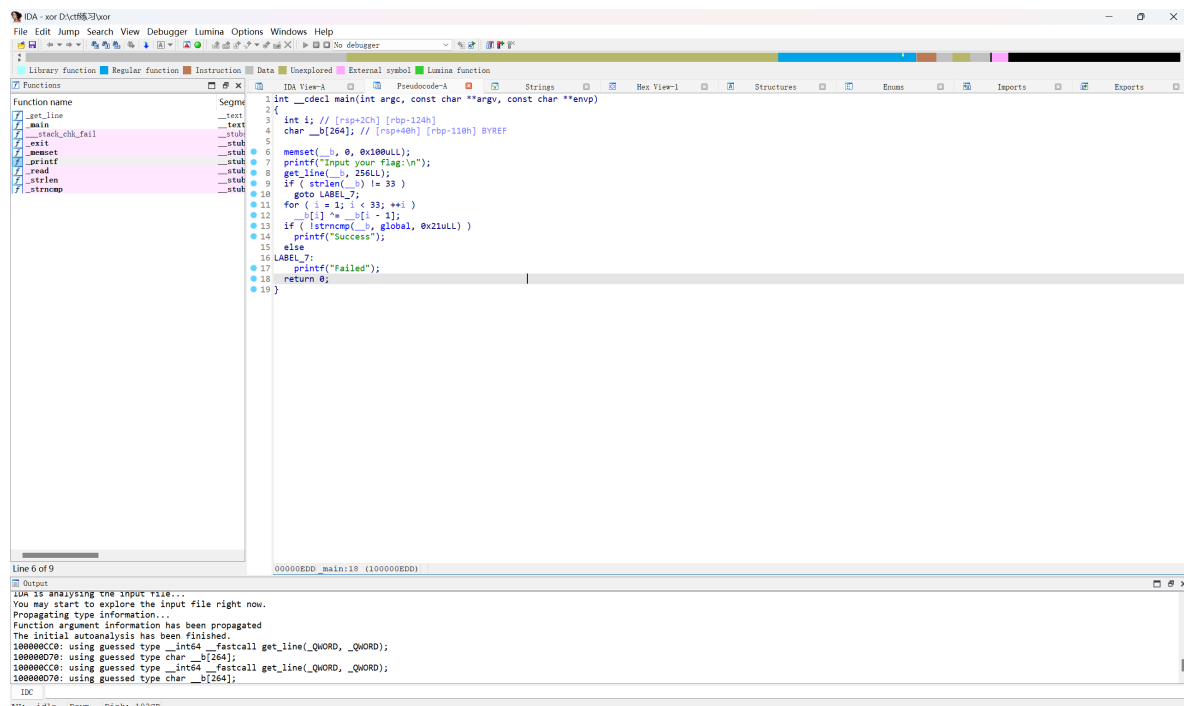
3.得到答案flag{HappyNewYear!}

六 xor

1.查壳，放入ida64，shift+f12调出字符串



2. ctrl+x交叉引用, f5反汇编调出主函数



3. 由于是异或，只要再进行异或一次即可获得原文

```
chars = [
    'f', 0x0A, 'k', 0x0C, 'w', '&', 'O', '.', '@', 0x11,
    'x', 0x0D, 'Z', ';', 'U', 0x11, 'p', 0x19, 'F', 0x1F,
    'v', '"', 'M', '#', 'D', 0x0E, 'g', 6, 'h', 0x0F, 'G', '2', 'O'
]
```

```
ascii_values = [ord(c) if isinstance(c, str) else c for c in chars]
```

```
result = []
```

```
for i in range(1, len(ascii_values)):
```

```
    xor_result = ascii_values[i] ^ ascii_values[i - 1]
```

```
    if 32 <= xor_result <= 126:
```

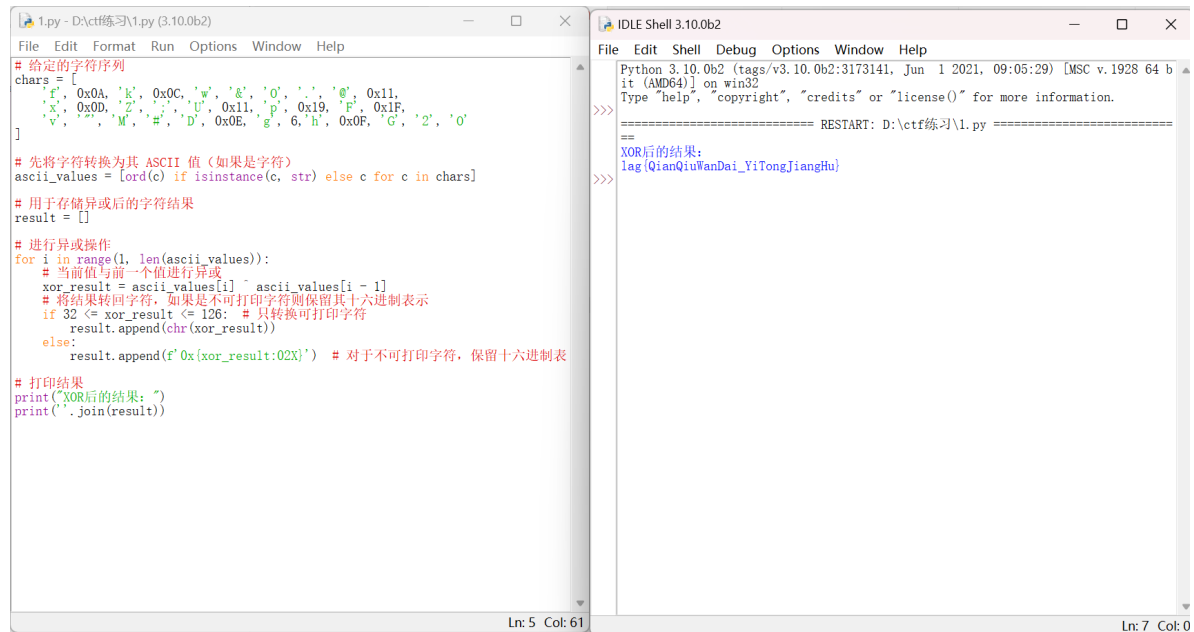
```
        result.append(chr(xor_result))
```

```

else:
    result.append(f'0x{xor_result:02X}')

print("XOR后的结果: ")
print(''.join(result))

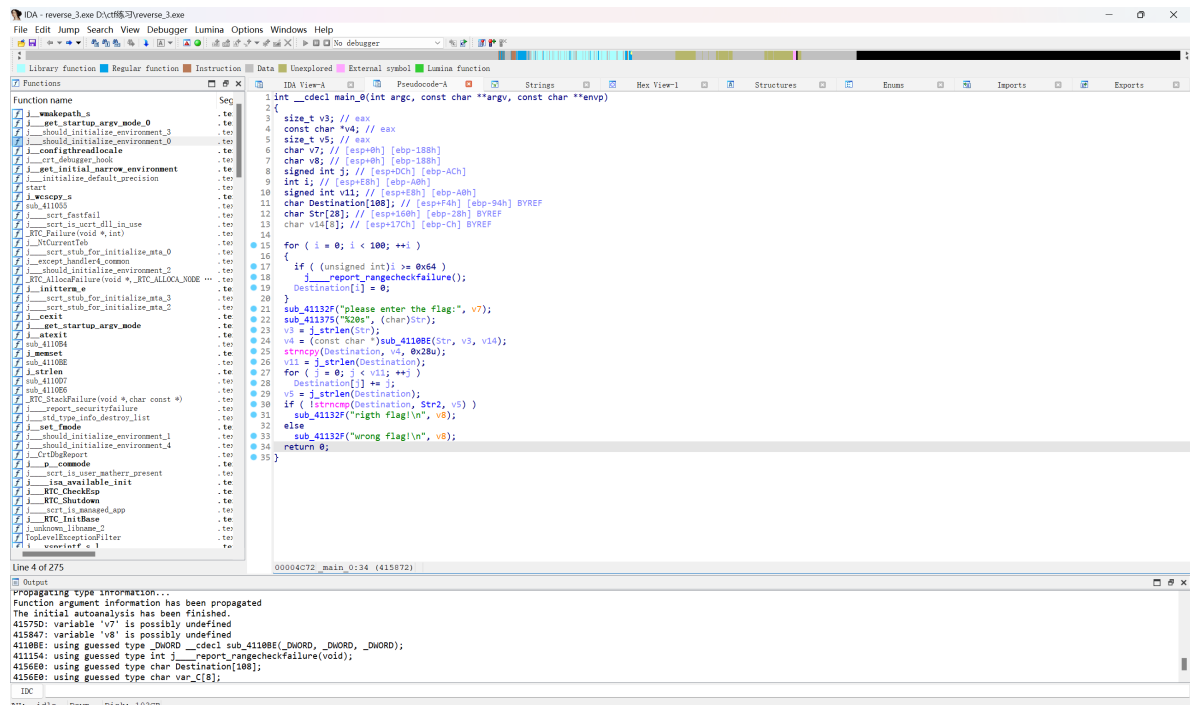
```



4.得到答案flag{QianQiuWanDai_YiTongJiangHu}

七 reserve3

1.查壳, 放入ida32, shift+f12, ctrl+x, f5调出主函数



2.读题得destination为e3niflH9b_C@n@dH, 解出原来的destination

destination = "e3niflH9b_C@n@dH"

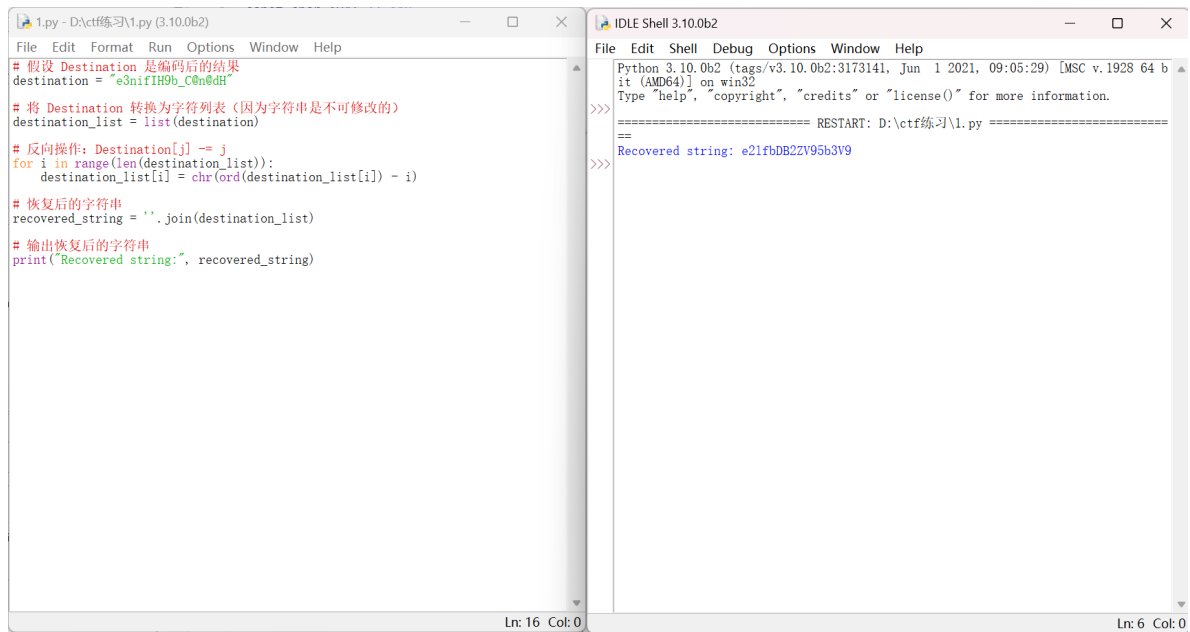
destination_list = list(destination)

for i in range(len(destination_list)):

destination_list[i] = chr(ord(destination_list[i]) - i)

```
recovered_string = ''.join(destination_list)

print("Recovered string:", recovered_string)
```

The image shows two windows from the IDLE 3.10.0b2 environment. The left window displays a Python script named '1.py' that performs a XOR operation to recover a string. The script starts with a comment in Chinese, followed by the definition of a 'destination' string, its conversion to a list, a loop to perform XOR with a key 'j', and finally joining the list back into a string and printing it. The right window shows the execution output, which includes the Python version, a restart message, and the final output: 'Recovered string: e21fbDB2ZV95b3V9'.

```
1.py - D:\ctf练习\1.py (3.10.0b2)
File Edit Format Run Options Window Help
# 假设 Destination 是编码后的结果
destination = "e3nifIH9b_C@n@dh"
# 将 Destination 转换为字符列表 (因为字符串是不可修改的)
destination_list = list(destination)
# 反向操作: Destination[j] -= j
for i in range(len(destination_list)):
    destination_list[i] = chr(ord(destination_list[i]) - i)
# 恢复后的字符串
recovered_string = ''.join(destination_list)
# 输出恢复后的字符串
print("Recovered string:", recovered_string)

Ln: 16 Col: 0

IDLE Shell 3.10.0b2
File Edit Shell Debug Options Window Help
Python 3.10.0b2 (tags/v3.10.0b2:3173141, Jun 1 2021, 09:05:29) [MSC v.1928 64 b
it (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\ctf练习\1.py =====
>>>
Recovered string: e21fbDB2ZV95b3V9
>>>

Ln: 6 Col: 0
```

3.base64解密得到答案flag{i_love_you}

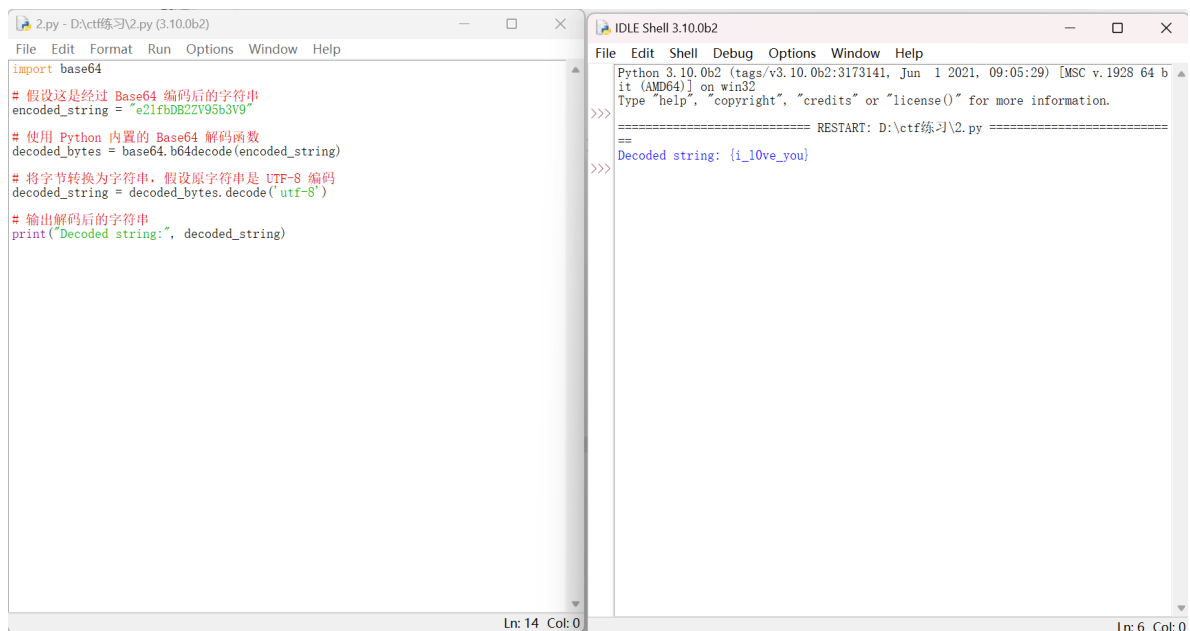
```
import base64

encoded_string = "e2lfbDB2ZV95b3V9"

decoded_bytes = base64.b64decode(encoded_string)

decoded_string = decoded_bytes.decode('utf-8')

print("Decoded string:", decoded_string)
```

The image shows two windows from the IDLE 3.10.0b2 environment. The left window displays a Python script named '2.py' that imports the 'base64' module, decodes a Base64 encoded string, and prints the result. The right window shows the execution output, which includes the Python version, a restart message, and the final output: 'Decoded string: {i_love_you}'.

```
2.py - D:\ctf练习\2.py (3.10.0b2)
File Edit Format Run Options Window Help
import base64
# 假设这是经过 Base64 编码后的字符串
encoded_string = "e2lfbDB2ZV95b3V9"
# 使用 Python 内置的 Base64 解码函数
decoded_bytes = base64.b64decode(encoded_string)
# 将字节转换为字符串, 假设原字符串是 UTF-8 编码
decoded_string = decoded_bytes.decode('utf-8')
# 输出解码后的字符串
print("Decoded string:", decoded_string)

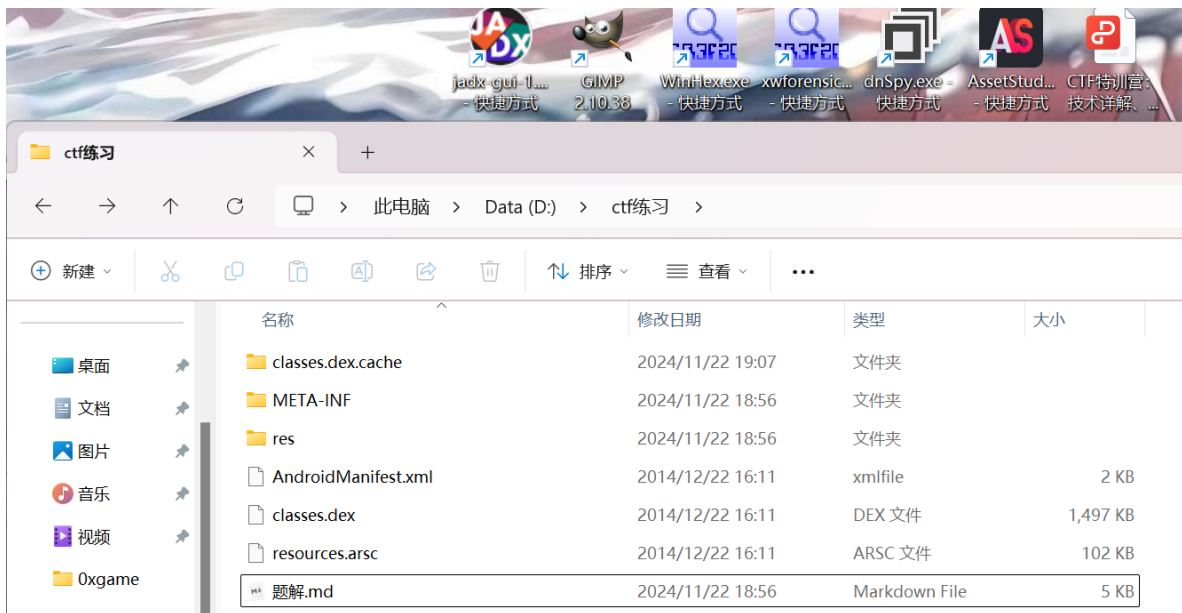
Ln: 14 Col: 0

IDLE Shell 3.10.0b2
File Edit Shell Debug Options Window Help
Python 3.10.0b2 (tags/v3.10.0b2:3173141, Jun 1 2021, 09:05:29) [MSC v.1928 64 b
it (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\ctf练习\2.py =====
>>>
Decoded string: {i_love_you}
>>>

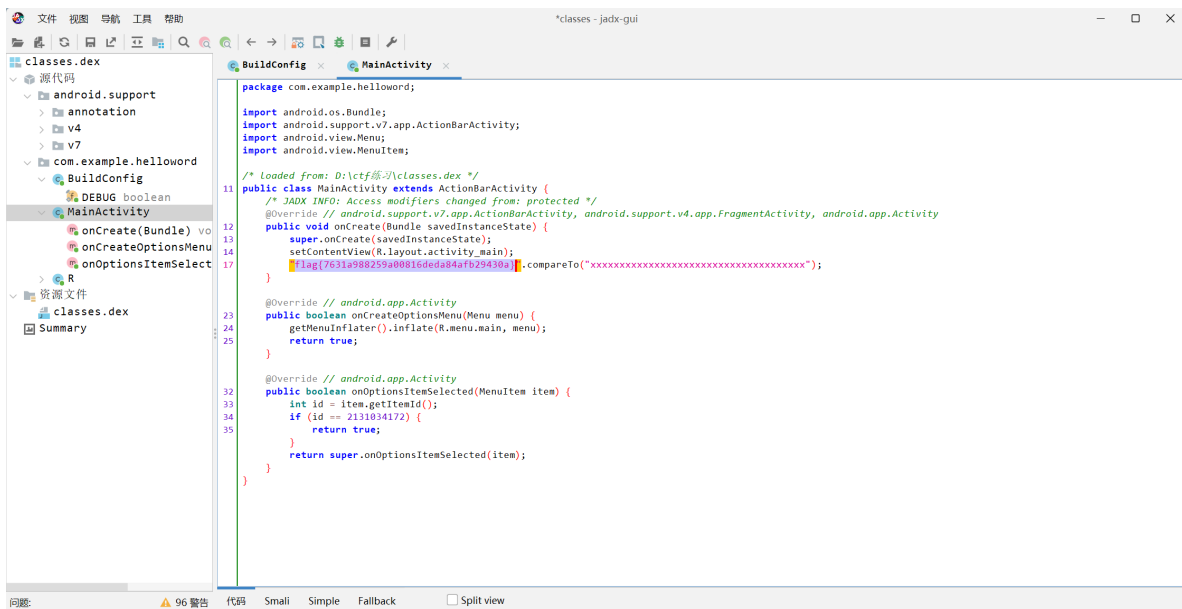
Ln: 6 Col: 0
```

八 helloworld

1.看到classes.dex文件, 放到jadx中

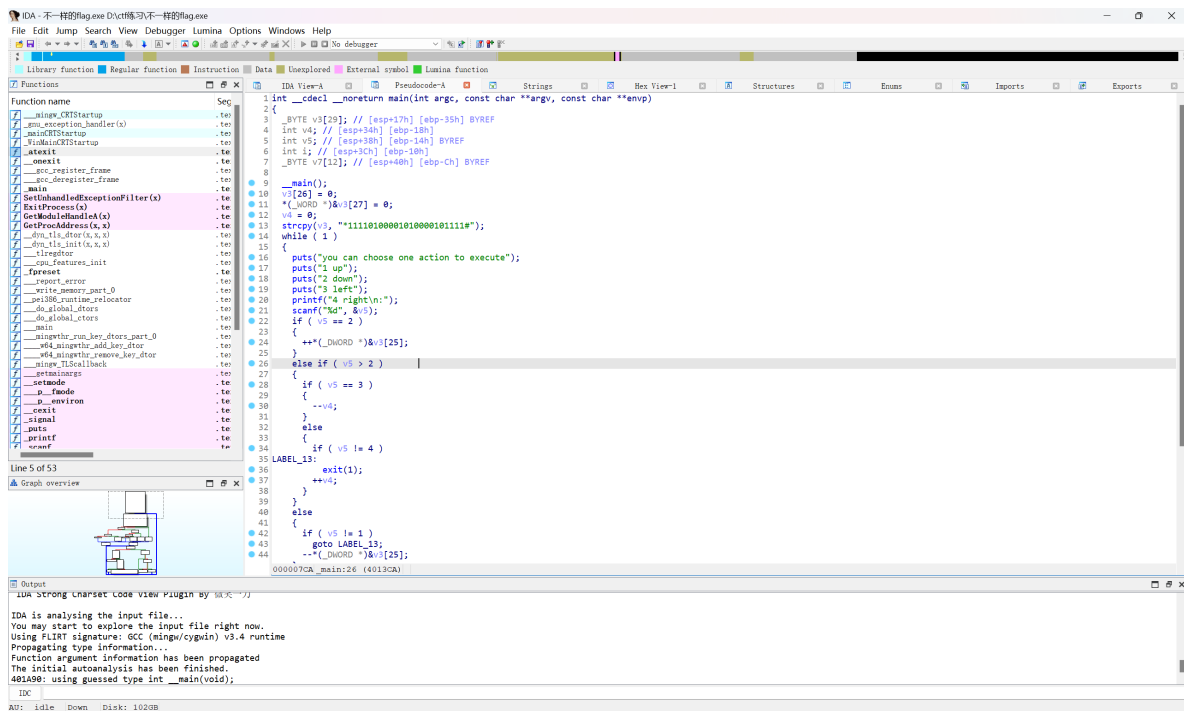


2.题目提到helloworld，那就打开helloworld这个文件，找到答案
flag{7631a988259a00816deda84afb29430a}



九 不一样的flag

1.查壳，放入ida32，shift+f12，ctrl+x，f5调出主函数



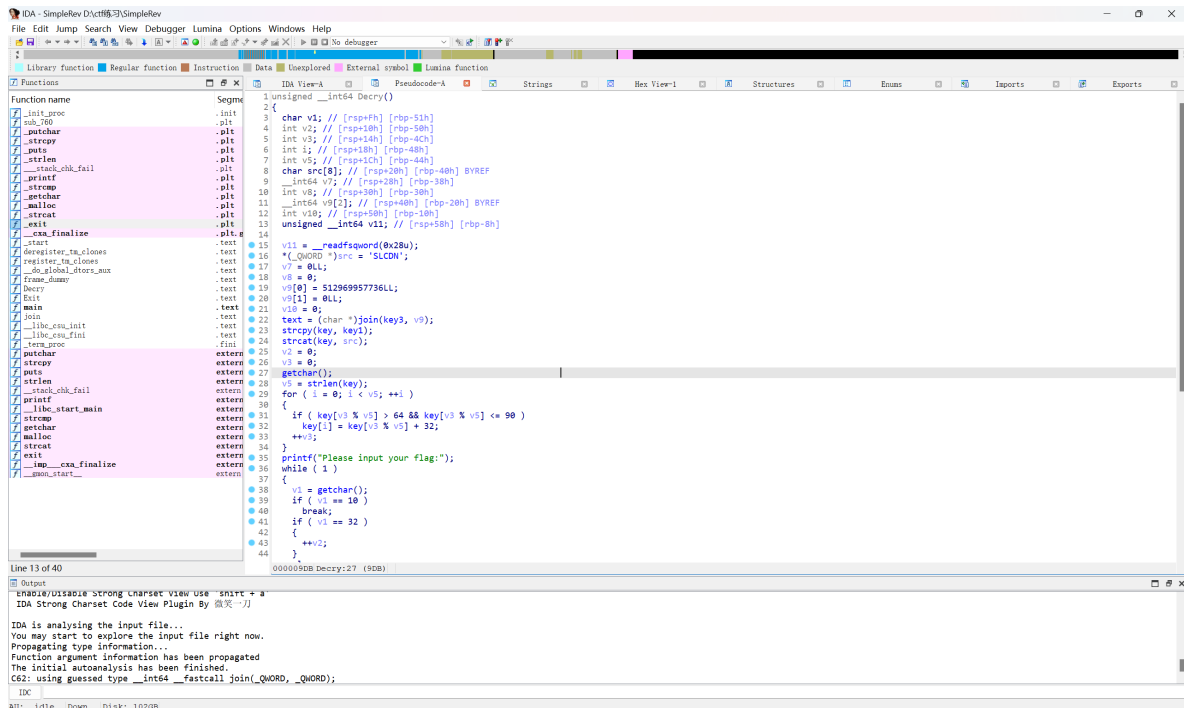
2.读代码，这是一个迷宫，1上 2下 3左 4右，迷宫为

```
x 1 1 1 1
0 1 0 0 0
0 1 0 1 0
0 0 0 1 0
1 1 1 1 #
```

3.答案为输入顺序，即flag{222441144222}

+ simplerev

1.查壳，放入ida64，shift+f12，ctrl+x，f5



2.读代码得text为killshadow key为ADSFKNDCLS，解题得到答案flag{KLDQCUDFZO}

```

text = "killshadow"
key = "ADSFKNDCLS"

key = list(key)
for i in range(len(key)):
    if ord(key[i]) >= 65 and ord(key[i]) <= 90:
        key[i] = chr(ord(key[i]) + 32)

key = ''.join(key)

encrypted_flag = ""

v3 = 0

for i in range(len(text)):
    for j in range(15):

        tmp = chr(ord(text[i]) - 97 + 26 * j - 97 + ord(key[v3 % len(key)]) + 39)

        if ord(tmp) >= 65 and ord(tmp) <= 90:

            encrypted_flag += tmp

            break

    v3+=1

print(f"Encrypted flag: {encrypted_flag}")

```

```

1.py - D:\ctf练习\1.py (3.10.0b2)
File Edit Format Run Options Window Help
# 已知的明文 (text) 和密钥 (key)
text = "killshadow" # 明文 (大写)
key = "ADSFKNDCLS" # 密钥 (大写)

# 将密钥转换为小写
key = list(key) # 将密钥转为列表
for i in range(len(key)):
    if ord(key[i]) >= 65 and ord(key[i]) <= 90: # 判断是否为大写字母
        key[i] = chr(ord(key[i]) + 32) # 转换为小写字母

# 重新生成小写的 key
key = ''.join(key)

# 初始化加密后的结果
encrypted_flag = ""
v3 = 0 # 密钥索引

# 加密过程
for i in range(len(text)):
    for j in range(15): # 尝试不同的偏移量
        # 计算加密后的字符
        tmp = chr(ord(text[i]) - 97 + 26 * j - 97 + ord(key[v3 % len(key)]) + 39)

        # 如果结果是大写字母 (A-Z), 则保存该字符并跳出循环
        if ord(tmp) >= 65 and ord(tmp) <= 90:
            encrypted_flag += tmp
            break

    # 更新密钥索引
    v3 += 1

# 输出加密后的 flag
print(f"Encrypted flag: {encrypted_flag}")

Ln: 8 Col: 26

IDLE Shell 3.10.0b2
File Edit Shell Debug Options Window Help
Python 3.10.0b2 (tags/v3.10.0b2:3173141, Jun 1 2021, 09:05:29) [MSC v.1928 64 b
it (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\ctf练习\1.py =====
>>>
Encrypted flag: KLDQCUDF20
>>>

Ln: 6 Col: 0

```