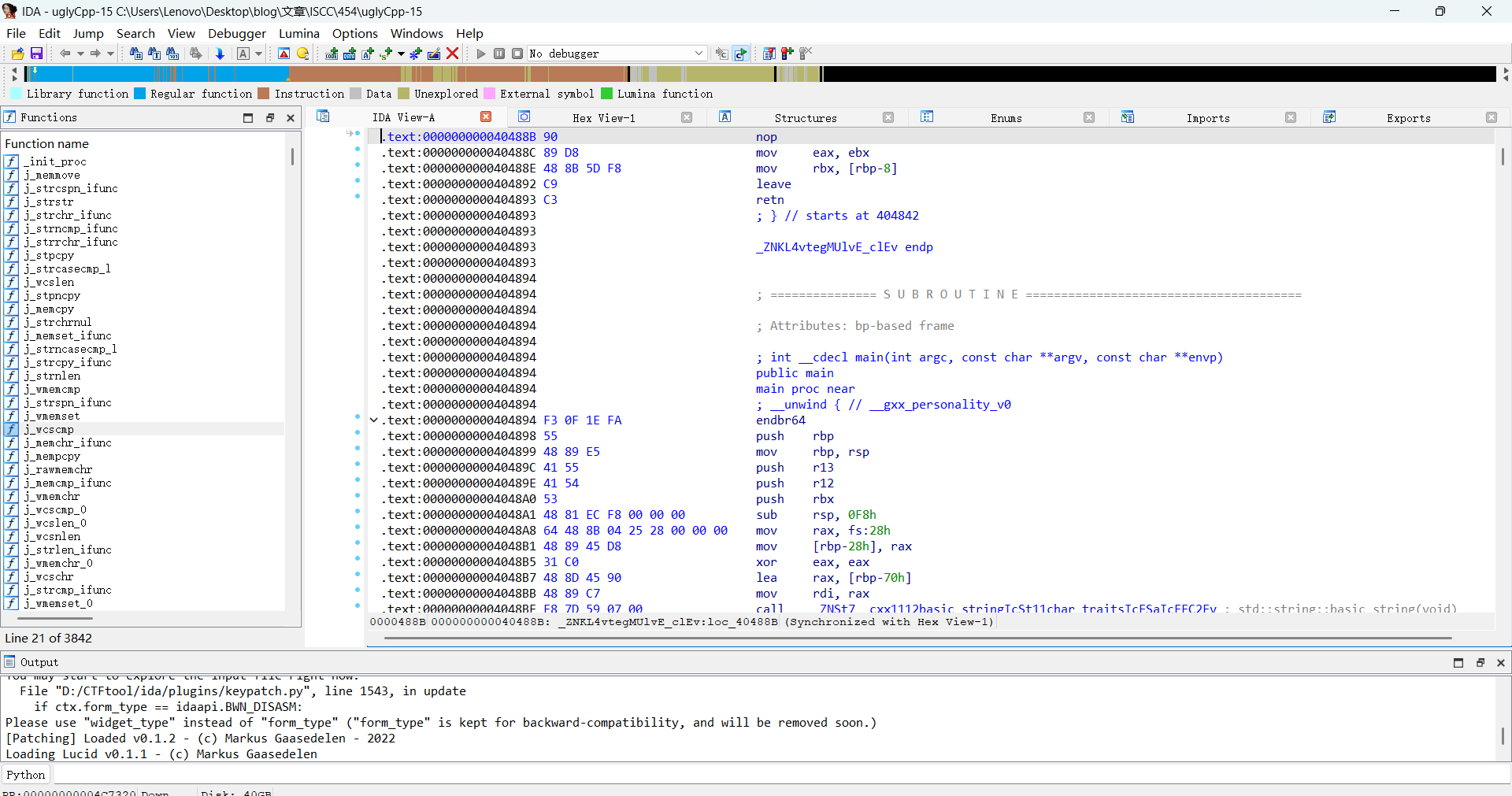
## QLNU-SEC-2403+于光泽+191897071@qq.com

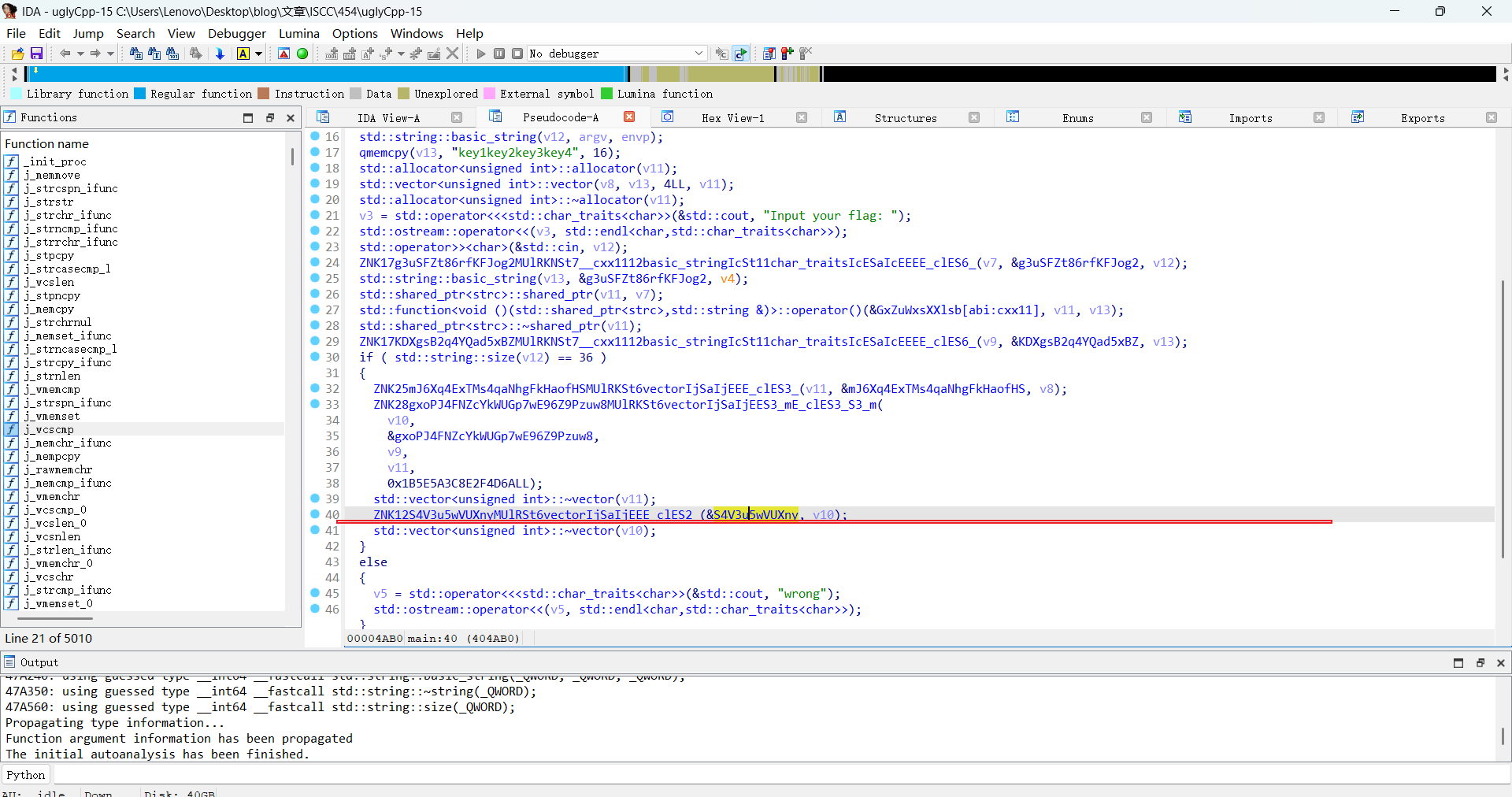
## REVERSE+ uglyCpp

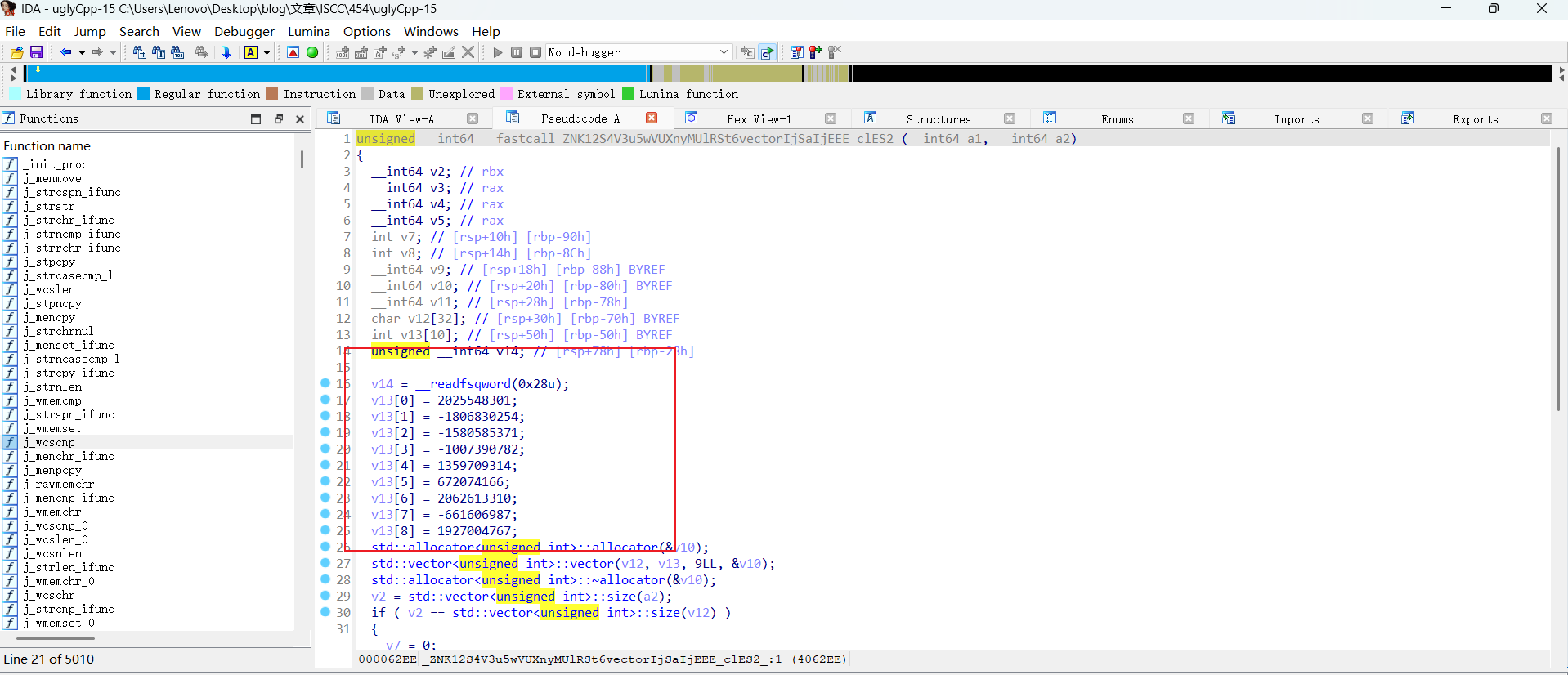
## 解题思路

1.放入ida中分析



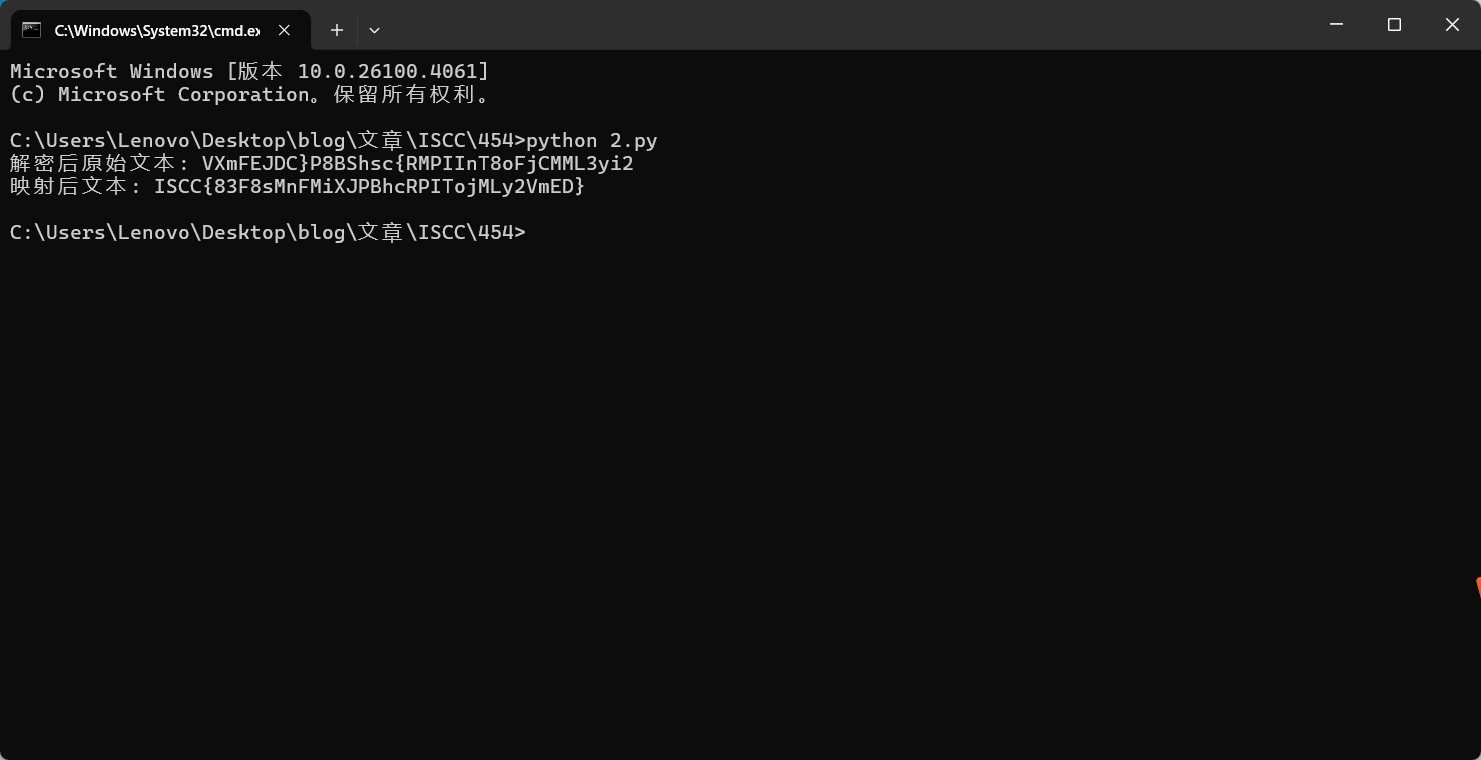
2.在j\_wcscmp函数中找到其中的密文





3.先异或解密，再字节解码，接着按照下面的规则映射

“5p6h7q8d9risbtjuevkwaxlyfzm0c1n2g3o4”“abcdefghijklmnopqrstuvwxyz0123456789”



## Exp:

def decrypt\_ciphertext(encrypted\_values, xor\_key\_list):

encrypted\_ints = [value & 0xffffffff for value in encrypted\_values]

decrypted\_ints = [encrypted\_ints[i] ^ xor\_key\_list[i] for i in range(len(encrypted\_ints))]

return decrypted\_ints

def convert\_ints\_to\_string(integer\_list):

try:

return "".join([num.to\_bytes(length=4, byteorder="little").decode() for num in integer\_list])

except UnicodeDecodeError:

print("解码失败，可能密钥错误或数据损坏")

return ""

def apply\_custom\_character\_mapping(input\_text, source\_character\_table, target\_character\_table):

result = []

for char in target\_character\_table:

if char in source\_character\_table:

position = source\_character\_table.index(char)

if position < len(input\_text):

result.append(input\_text[position])

return ''.join(result)

def main():

encrypted\_data = [

2025548301, -1806830254, -1580585371, -1007390782,

1359709314, 672074166, 2062613310, -661606987, 1927004767

]

decryption\_key = [

0x3ED6325B, 0xD709BF17, 0xE3F27E18, 0xA0870791,

0x0146D6F9, 0x7C6140FF, 0x10B69406, 0x94DDE0F6, 0x40B2BB6C

]

source\_mapping\_table = "5p6h7q8d9risbtjuevkwaxlyfzm0c1n2g3o4"

target\_mapping\_table = "abcdefghijklmnopqrstuvwxyz0123456789"

decrypted\_values = decrypt\_ciphertext(encrypted\_data, decryption\_key)

original\_text = convert\_ints\_to\_string(decrypted\_values)

print("解密后原始文本:", original\_text)

transformed\_text = apply\_custom\_character\_mapping(original\_text, source\_mapping\_table, target\_mapping\_table)

print("映射后文本:", transformed\_text)

if \_\_name\_\_ == "\_\_main\_\_":

main()