**广州大学学生实验报告**

**开课学院及实验室：**网络空间安全学院 **2023年2月22日**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学院** | 网络空间安全学院 | **年级/专业/班** | 网安211 | **姓名** |  | **学号** |  |
| **实验课程名称** | 现代密码学实验 | | | | | **成绩** |  |
| **实验项目名称** |  | | | | | **指导**  **老师** |  |

1. 实验目的
2. 选择一门语言（C、Python、Java等），熟悉DES和AES密码库的使用；
   1. 使用DES算法CBC模式，通过以下参数产生密文（m均不包含双引号，并验证密文是否为c = cffffa20a73c667b45fec867850a70a382dad87e90ef16a1。

m = ”today\_is\_tuesday”

key = 0x0011223344556677

iv = 0x7766554433221100

* 1. 使用AES算法CBC模式，通过以下参数产生密文（m均不包含双引号，并验证密文是否为:

c = 6d8e688ec4104ede9e925f49df4eaf79f1cfd4155cc5a15b48e73cda91adefa0。

m = ” today\_is\_tuesday\_helloworld”

key = 0x00112233445566770011223344556677

iv = 0x77665544332211007766554433221100

1. 验证以下哪些密钥是DES的弱密钥或半弱密钥？不考虑校验位：

0x0000000000000000 0x00FF00EFF0FF00FF

0x00FF00E100FF00FF 0xE1E1E1E1F0F0F0F0

1. 实验内容

（1）安装；

（2）熟悉Linux常用命令；

（3）安装gcc编程开发环境；

（4）练习使用gcc编译C程序并执行。

1. 实验过程及结果

1.

from Crypto.Cipher import DES

from Crypto.Util.Padding import pad

from binascii import unhexlify, hexlify

# Provided parameters

plaintext = b"today\_is\_tuesday"

key = unhexlify("0011223344556677")

iv = unhexlify("7766554433221100")

given\_ciphertext = "cffffa20a73c667b45fec867850a70a382dad87e90ef16a1"

# Padding the plaintext to a multiple of 8 bytes (64 bits)

padded\_plaintext = pad(plaintext, 8)

# Encrypt the plaintext using DES in CBC mode

cipher = DES.new(key, DES.MODE\_CBC, iv)

ciphertext = cipher.encrypt(padded\_plaintext)

# Convert the ciphertext to a hexadecimal string

ciphertext\_hex = hexlify(ciphertext).decode()

print("ciphertext\_hex:", ciphertext\_hex)

print("given\_ciphertext:", given\_ciphertext)

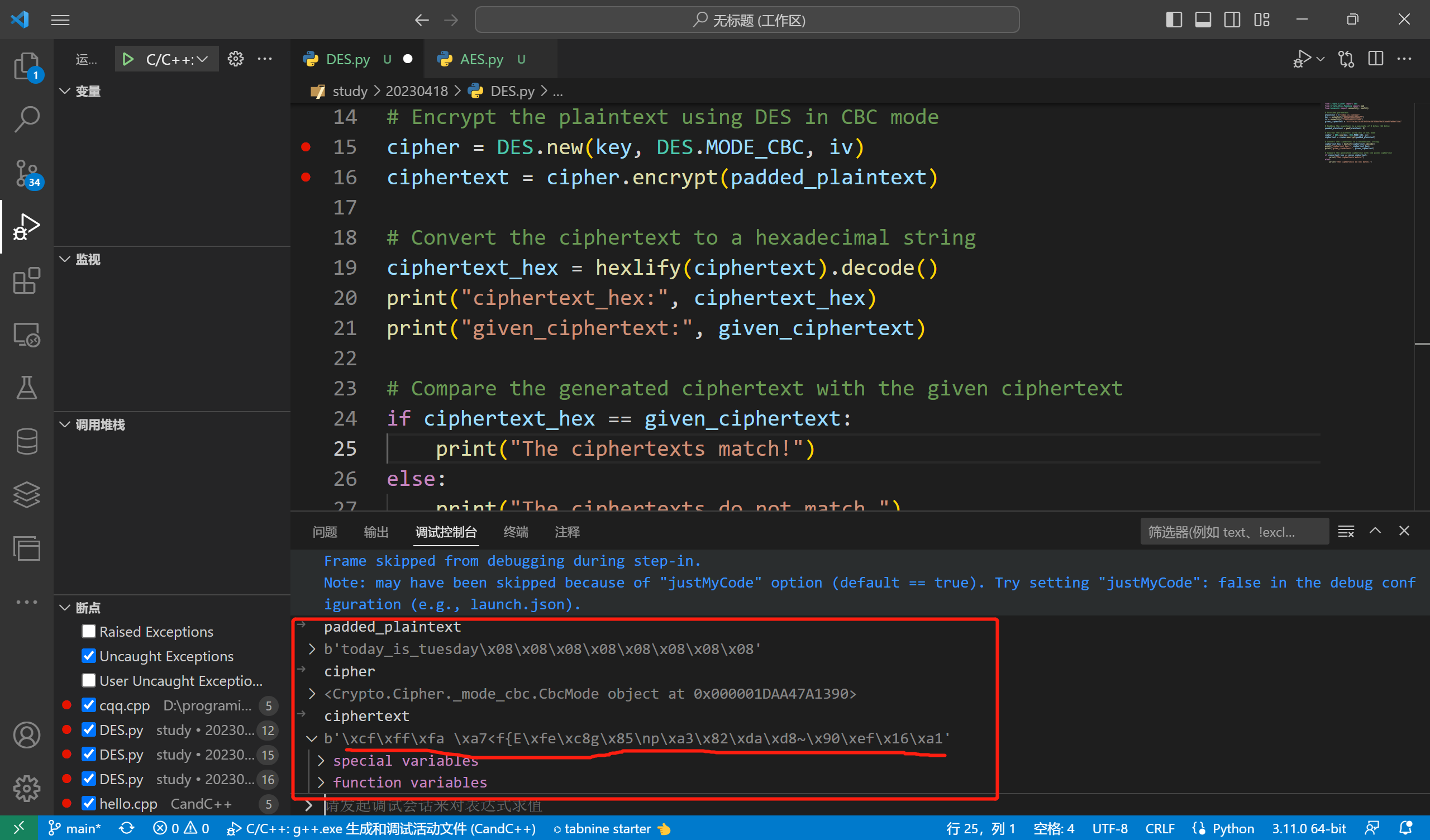
# Compare the generated ciphertext with the given ciphertext

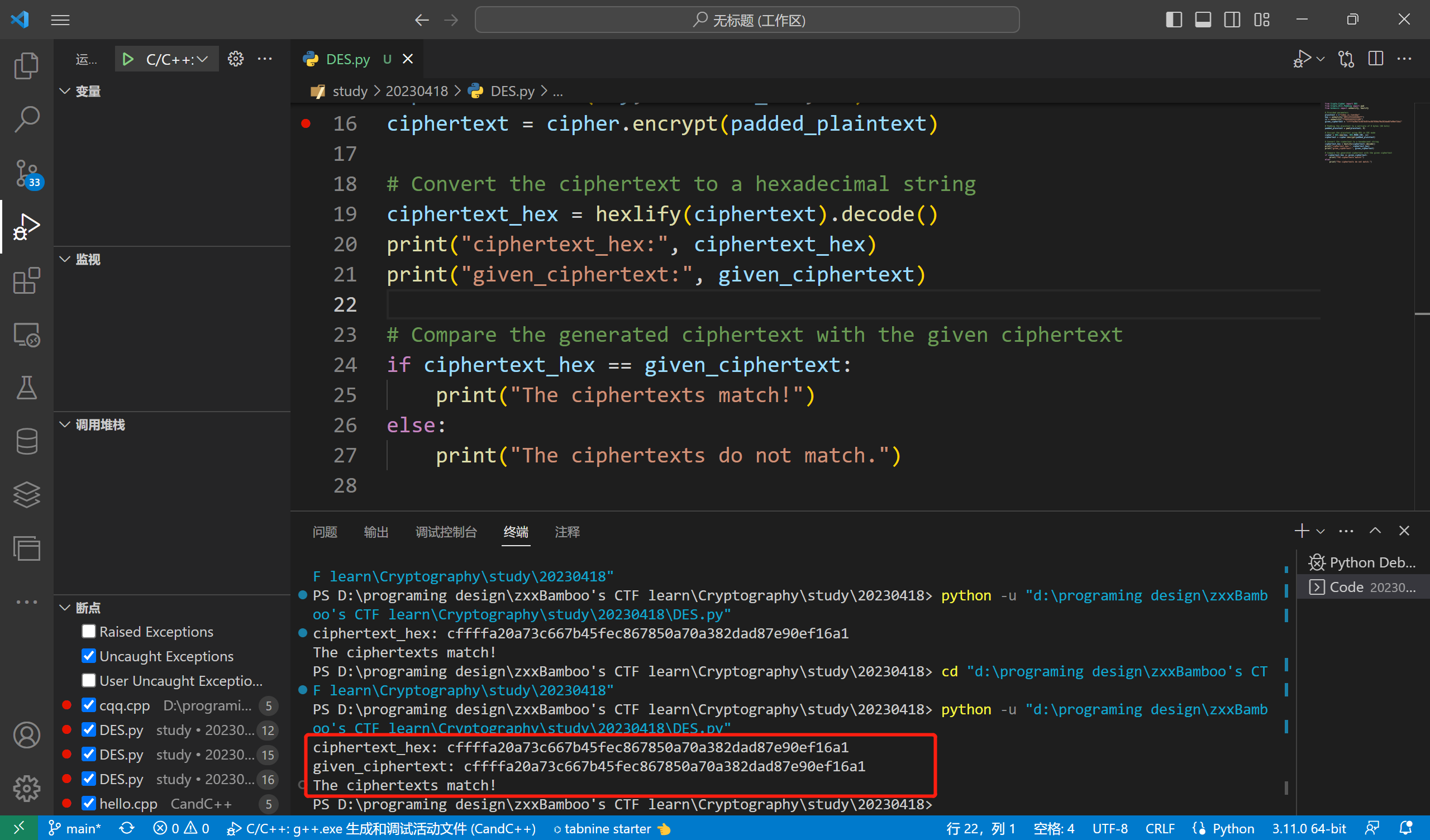
if ciphertext\_hex == given\_ciphertext:

    print("The ciphertexts match!")

else:

    print("The ciphertexts do not match.")





2.

from Crypto.Cipher import AES

from Crypto.Util.Padding import pad

from binascii import unhexlify, hexlify

# Provided parameters

plaintext = b"today\_is\_tuesday\_helloworld"

key = unhexlify("00112233445566770011223344556677")

iv = unhexlify("77665544332211007766554433221100")

given\_ciphertext = "6d8e688ec4104ede9e925f49df4eaf79f1cfd4155cc5a15b48e73cda91adefa0"

# Padding the plaintext to a multiple of 16 bytes (128 bits)

padded\_plaintext = pad(plaintext, 16)

# Encrypt the plaintext using AES in CBC mode

cipher = AES.new(key, AES.MODE\_CBC, iv)

ciphertext = cipher.encrypt(padded\_plaintext)

# Convert the ciphertext to a hexadecimal string

ciphertext\_hex = hexlify(ciphertext).decode()

print("ciphertext\_hex:", ciphertext\_hex)

print("given\_ciphertext:", given\_ciphertext)

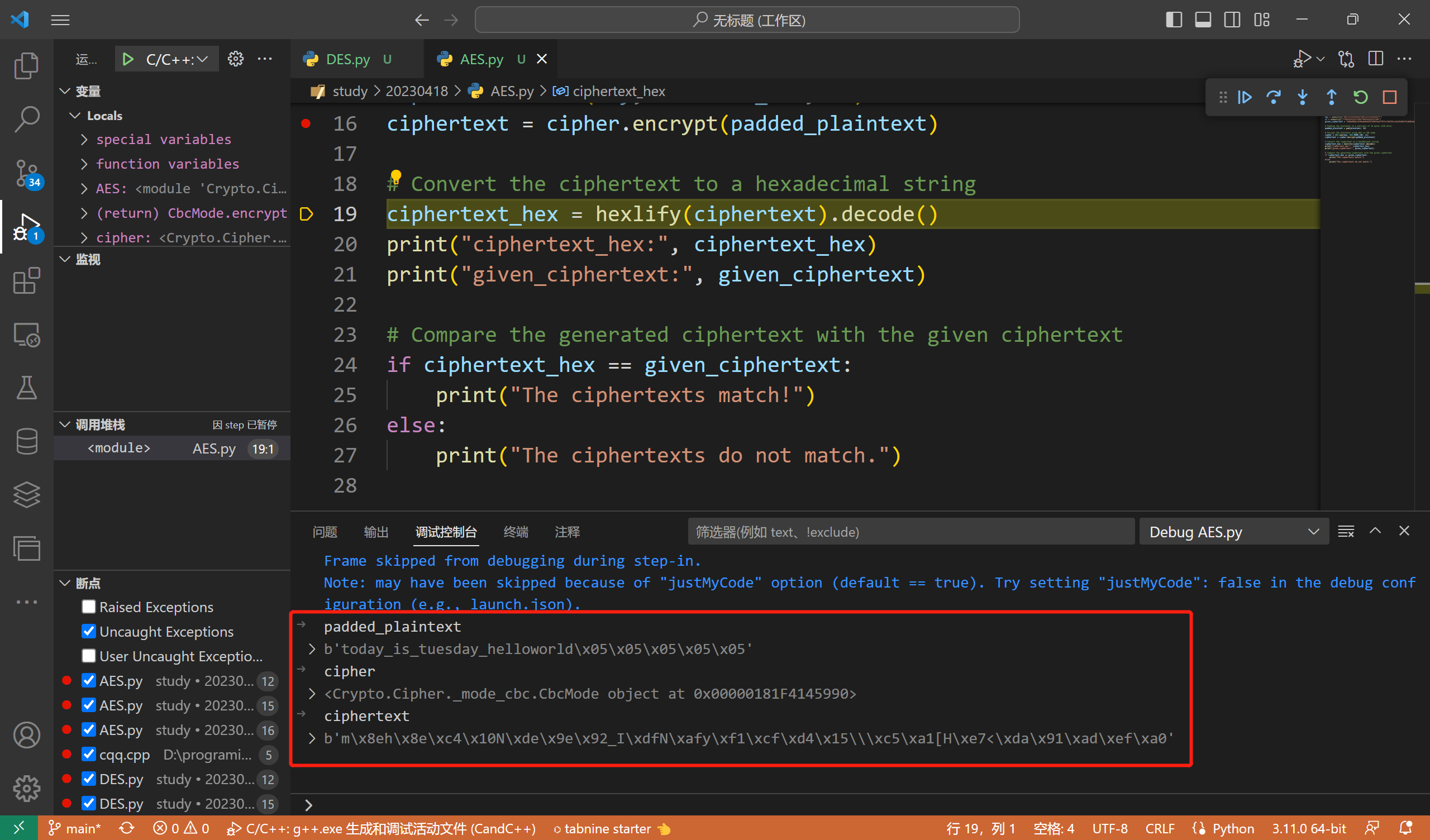
# Compare the generated ciphertext with the given ciphertext

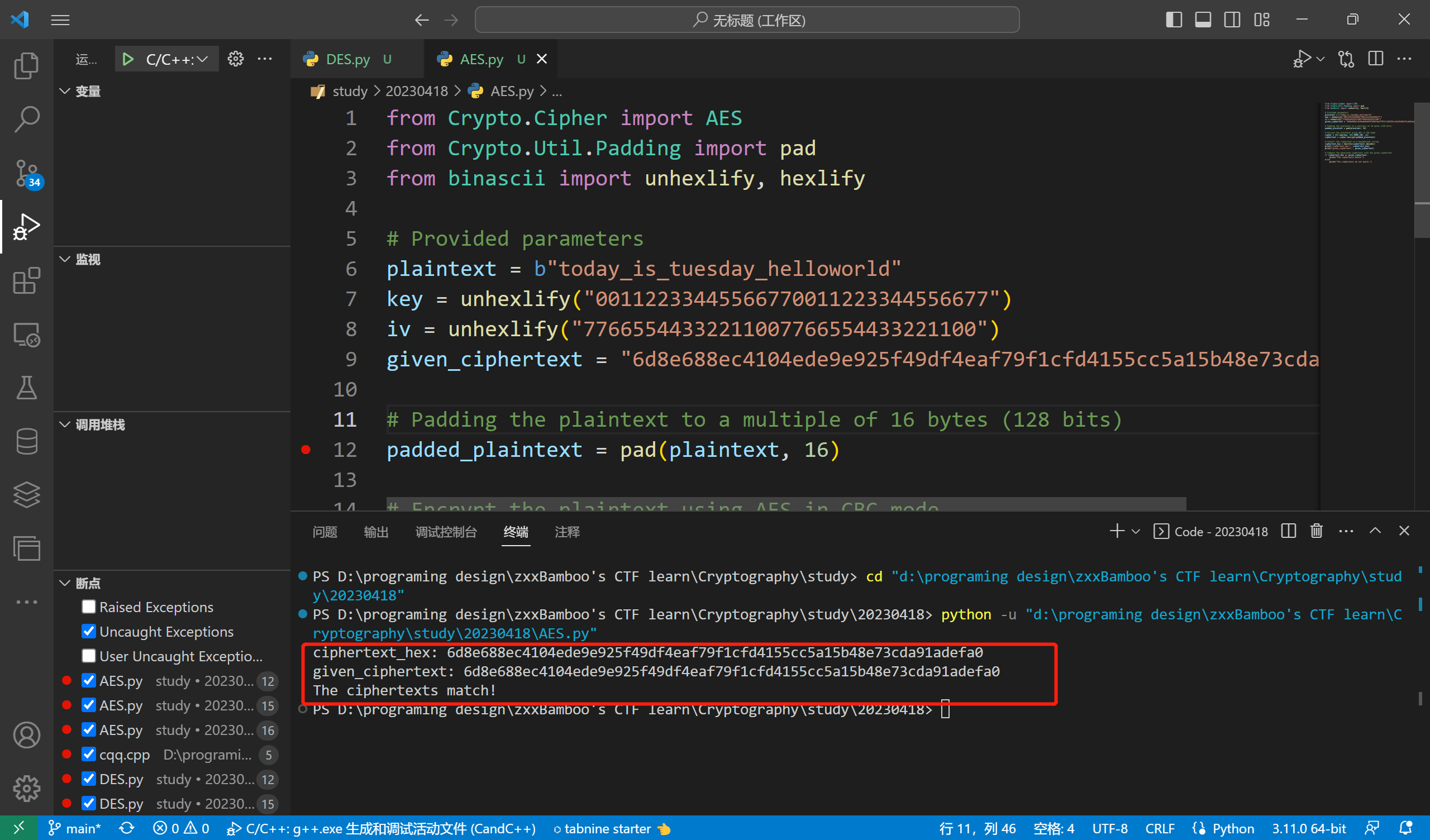
if ciphertext\_hex == given\_ciphertext:

    print("The ciphertexts match!")

else:

    print("The ciphertexts do not match.")





3.

from Crypto.Cipher import DES

from Crypto.Util.Padding import pad

from binascii import unhexlify, hexlify

def generate\_subkeys(key):

    des = DES.new(key, DES.MODE\_ECB)

    c = des.encrypt(pad(b'0' \* 8, 8))

    d = des.decrypt(c)

    return (c, d)

def check\_weak\_key(key):

    c, d = generate\_subkeys(key)

    return c == d

def check\_semi\_weak\_keys(key1, key2):

    c1, d1 = generate\_subkeys(key1)

    c2, d2 = generate\_subkeys(key2)

    return c1 == d2 and c2 == d1

# Provided keys

keys = [

    unhexlify("0000000000000000"),

    unhexlify("00FF00EFF0FF00FF"),

    unhexlify("00FF00E100FF00FF"),

    unhexlify("E1E1E1E1F0F0F0F0")

]

for key in keys:

    if check\_weak\_key(key):

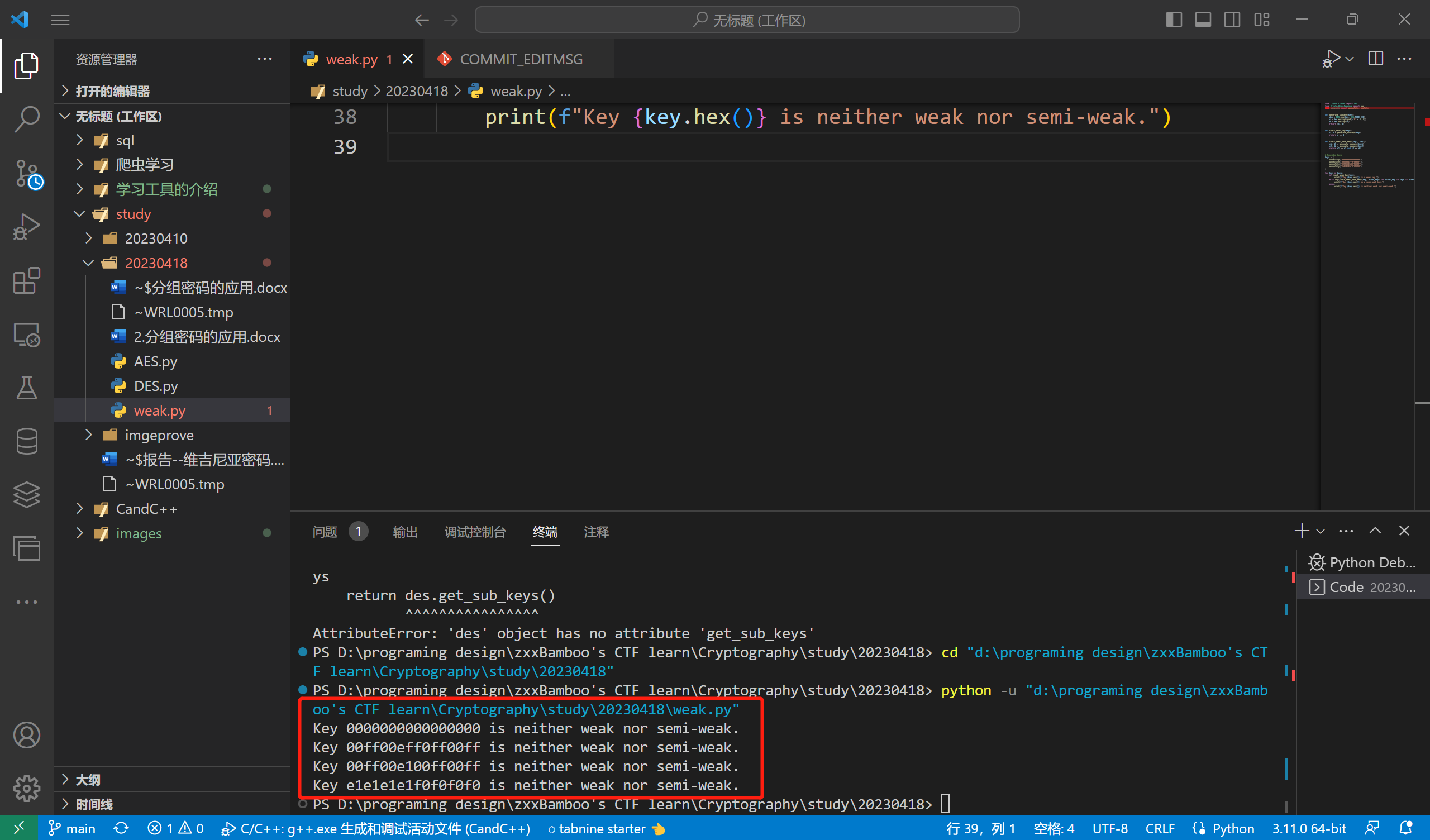
        print(f"Key {key.hex()} is a weak key.")

    elif any(check\_semi\_weak\_keys(key, other\_key) for other\_key in keys if other\_key != key):

        print(f"Key {key.hex()} is a semi-weak key.")

    else:

        print(f"Key {key.hex()} is neither weak nor semi-weak.")



1. 实验总结