**Task1**

prompt:

def is\_prime(n):

    if n <= 1:

        return False

    for i in range(2, n):

        if n % i == 0:

            return False

    return True

print is\_prime(7)

print is\_prime(10)

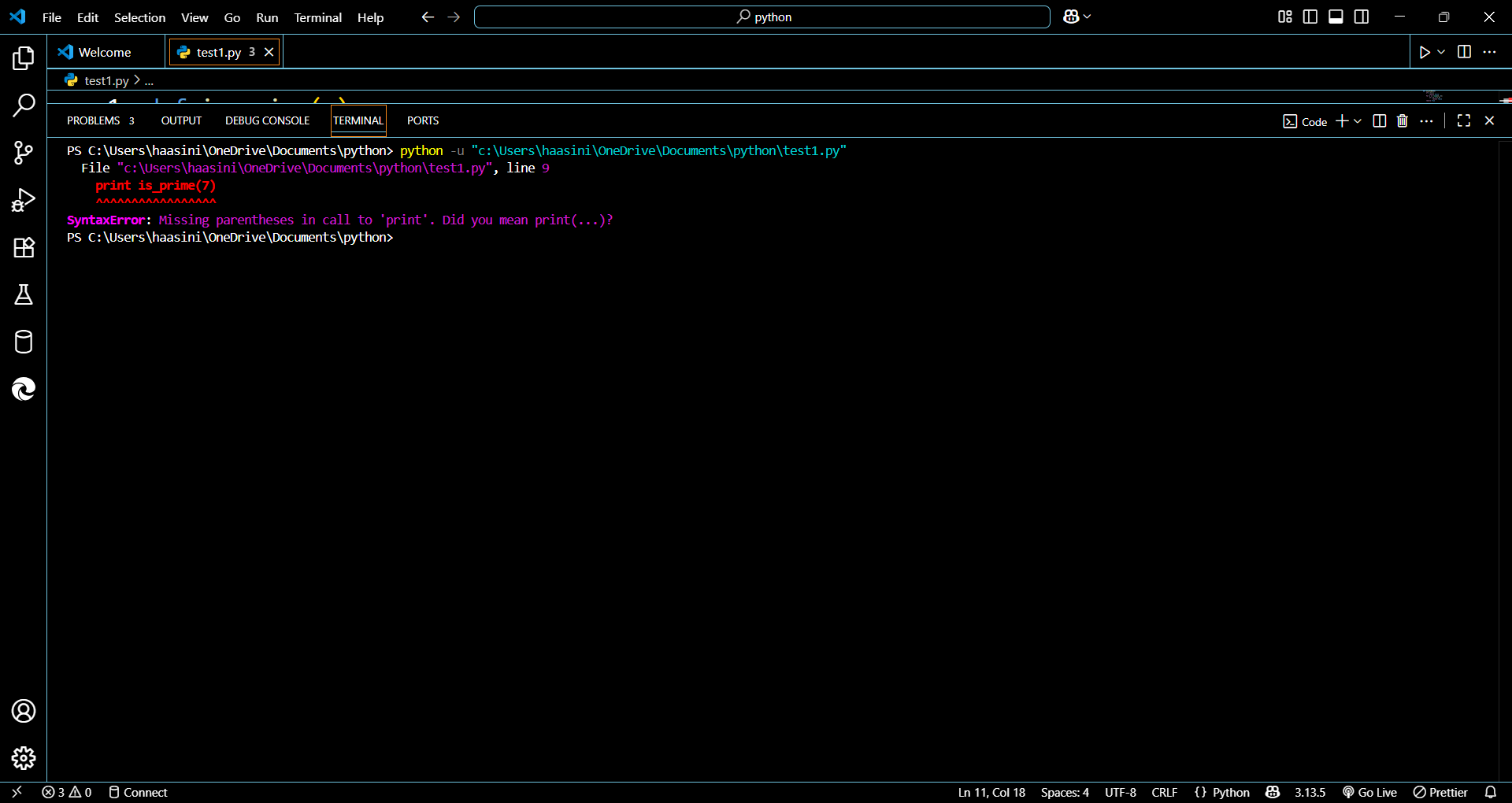
print is\_prime(1)

Find the syntax error in above code.

Code with error:



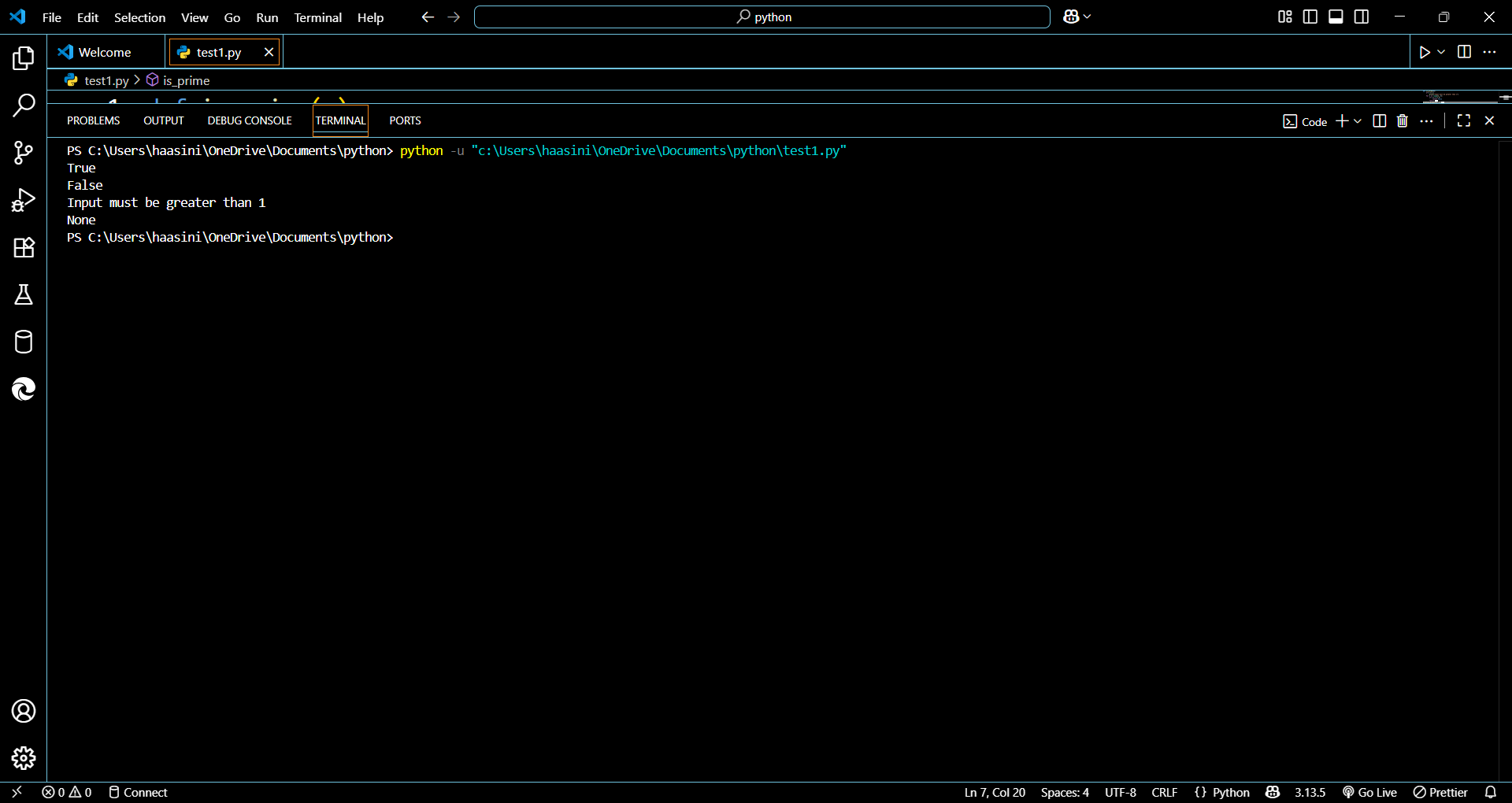
Error:



Output:

Fixed code:





**TASK 2:**

Prompt:

def checknum(n):

    if n%2=0:

        print("even")

    else:

        print("odd")

n=int(input("Enter a number: "))

checknum(n)

identity and explain why this causes bug.



Error:



Output:

The bug is caused due to using = instead of ==.



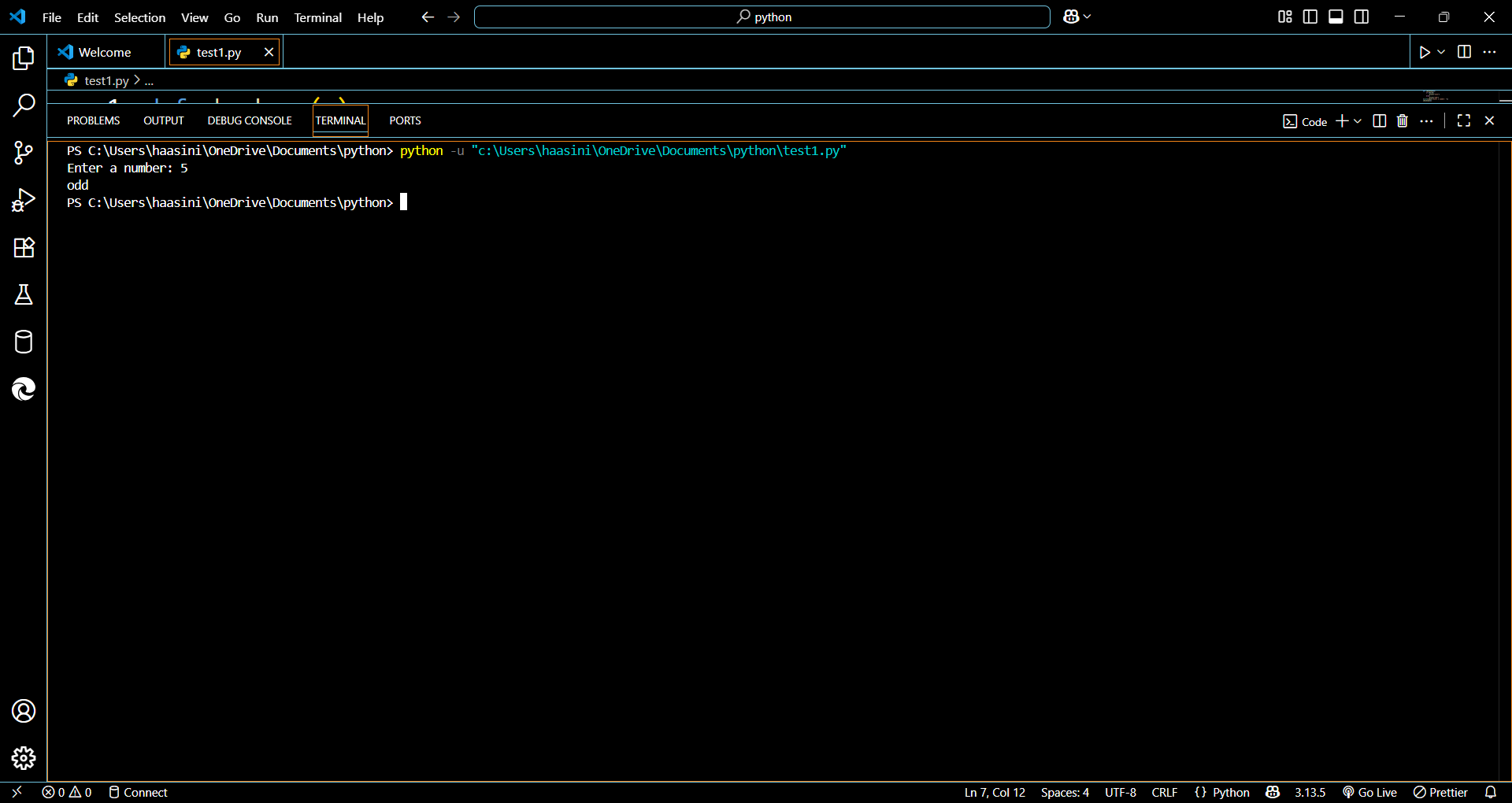


Test cases:

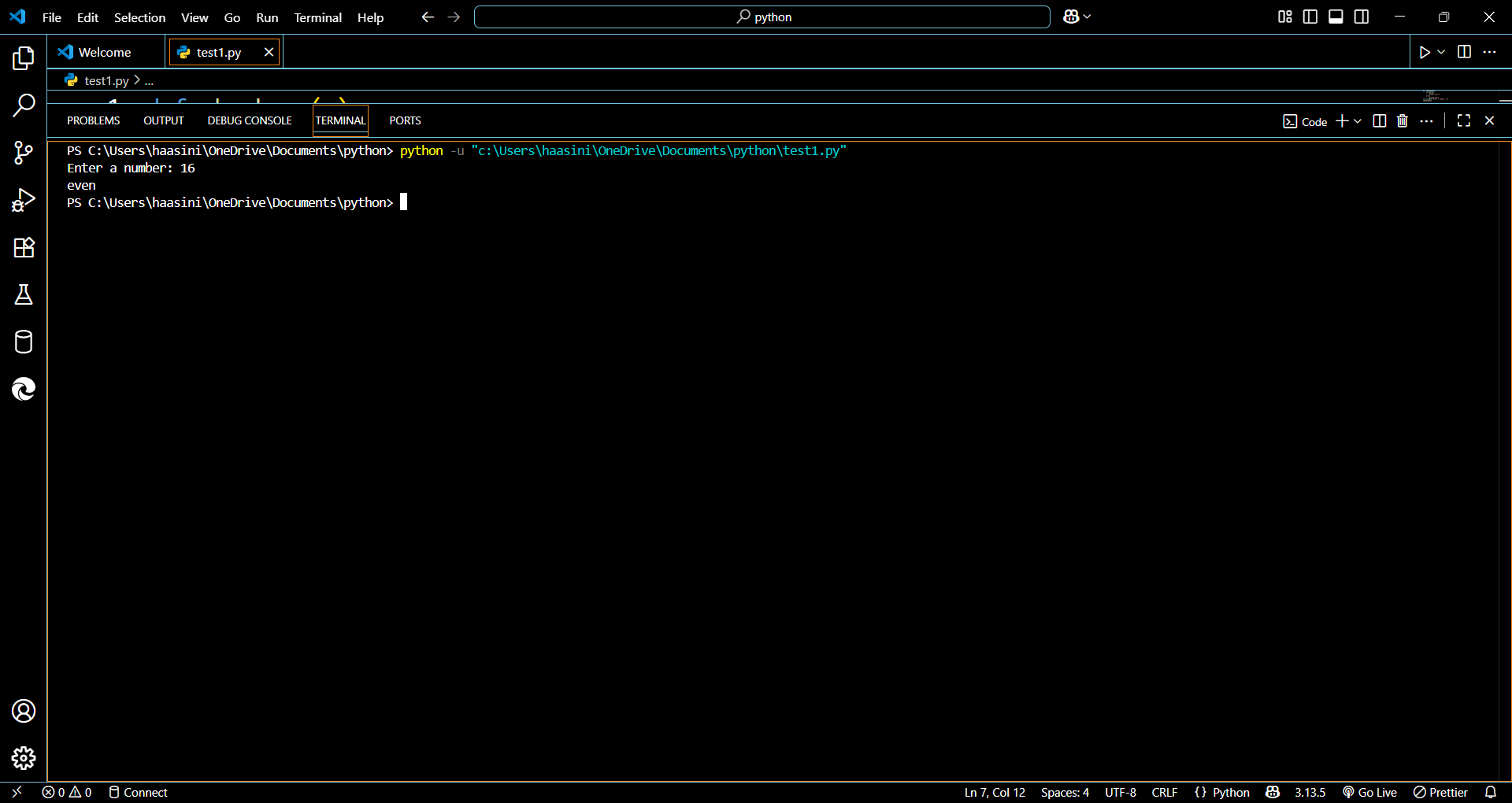
N=5

N=10

N=16







**TASK 3**

Prompt:

def read\_file(filename):

With open(filename, ‘r’) as f:

Retun f.read()

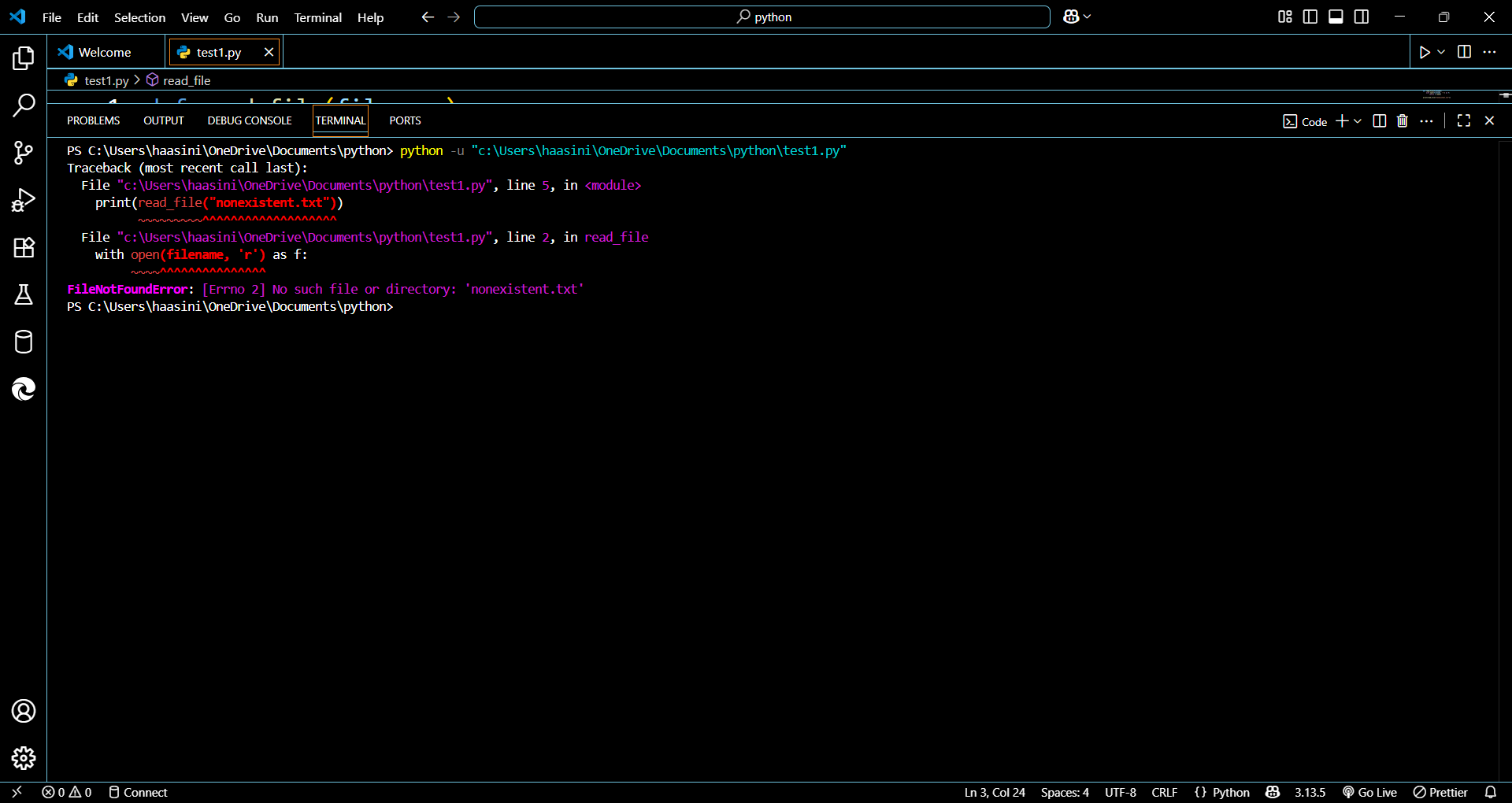
print(read\_file(“nonexistent.txt”)

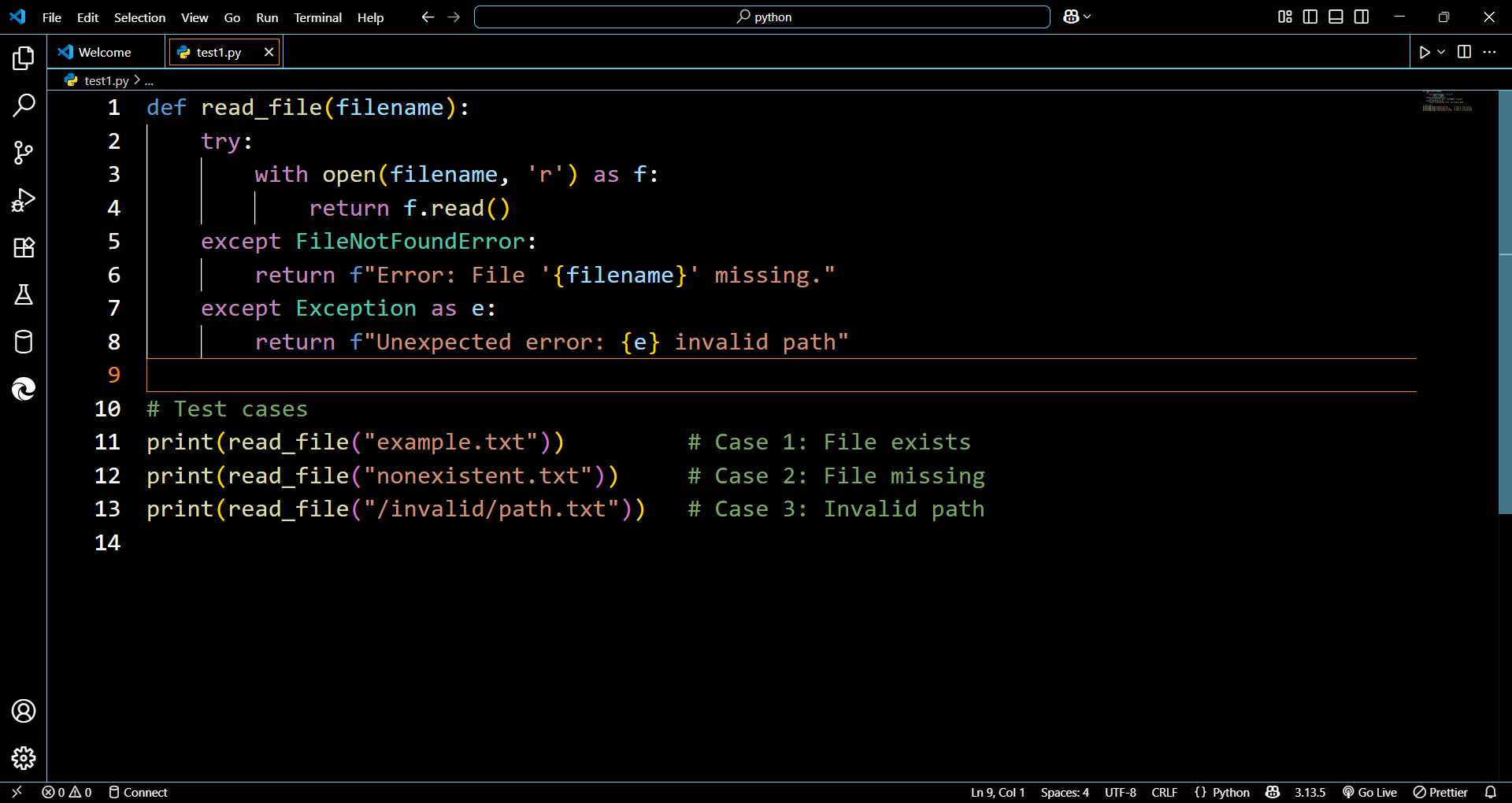
fix the error with proper file handling like reading the lines from text file , printing invalid path for wrong past and missing file if file path is not existing.

Code:

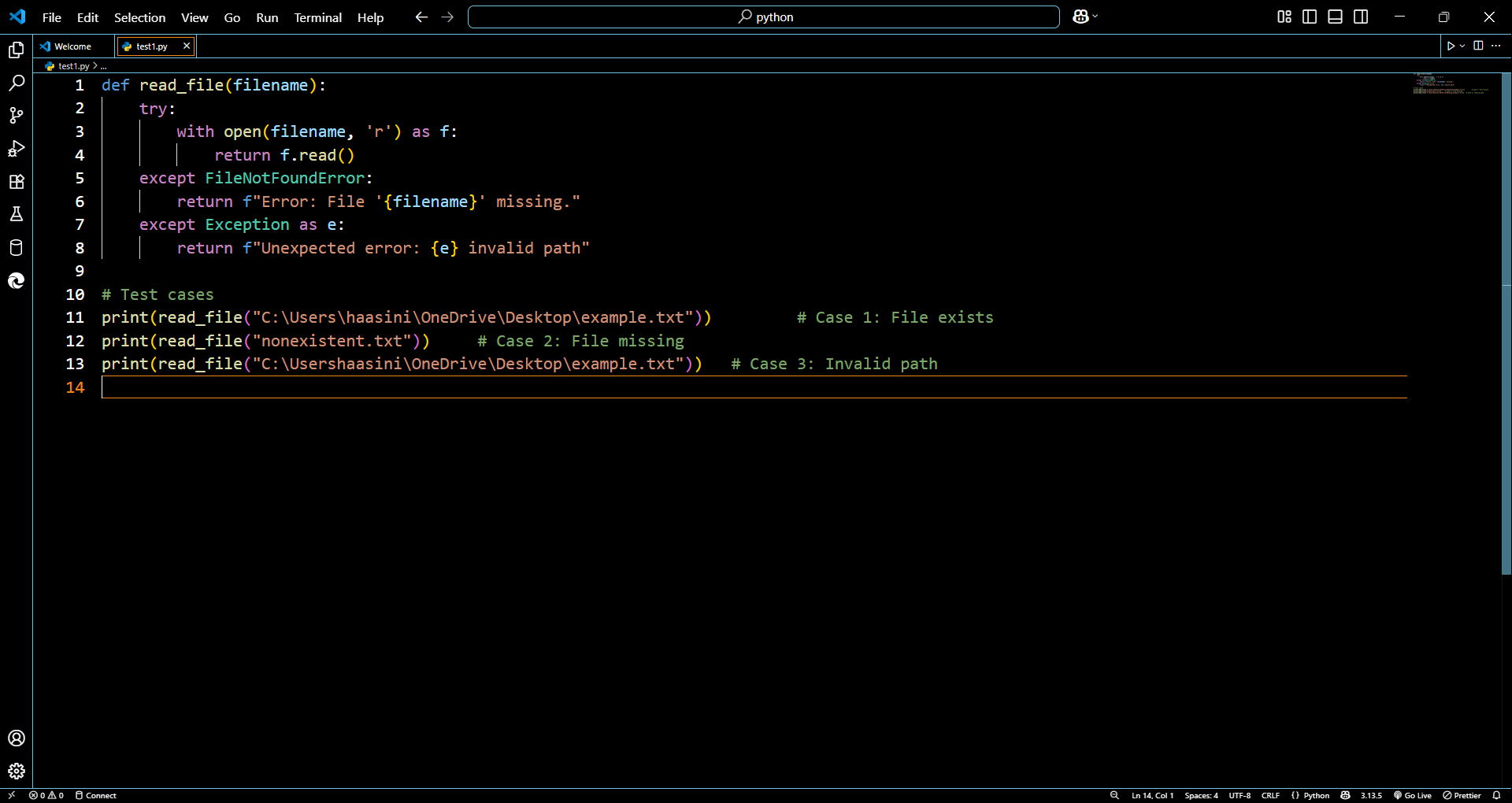


Error:



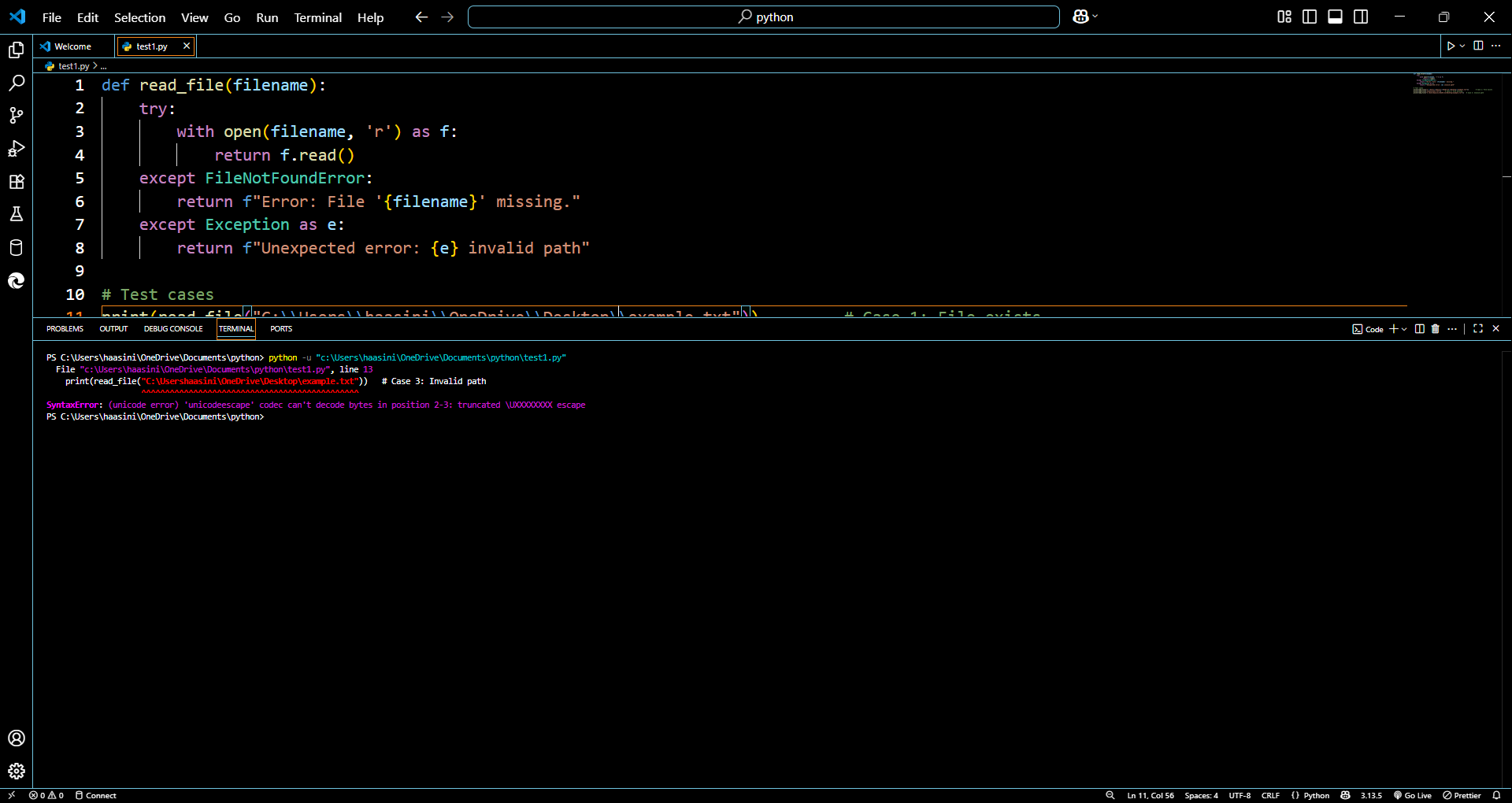
Fixed code: 

Output:



Test case 1 invalid path

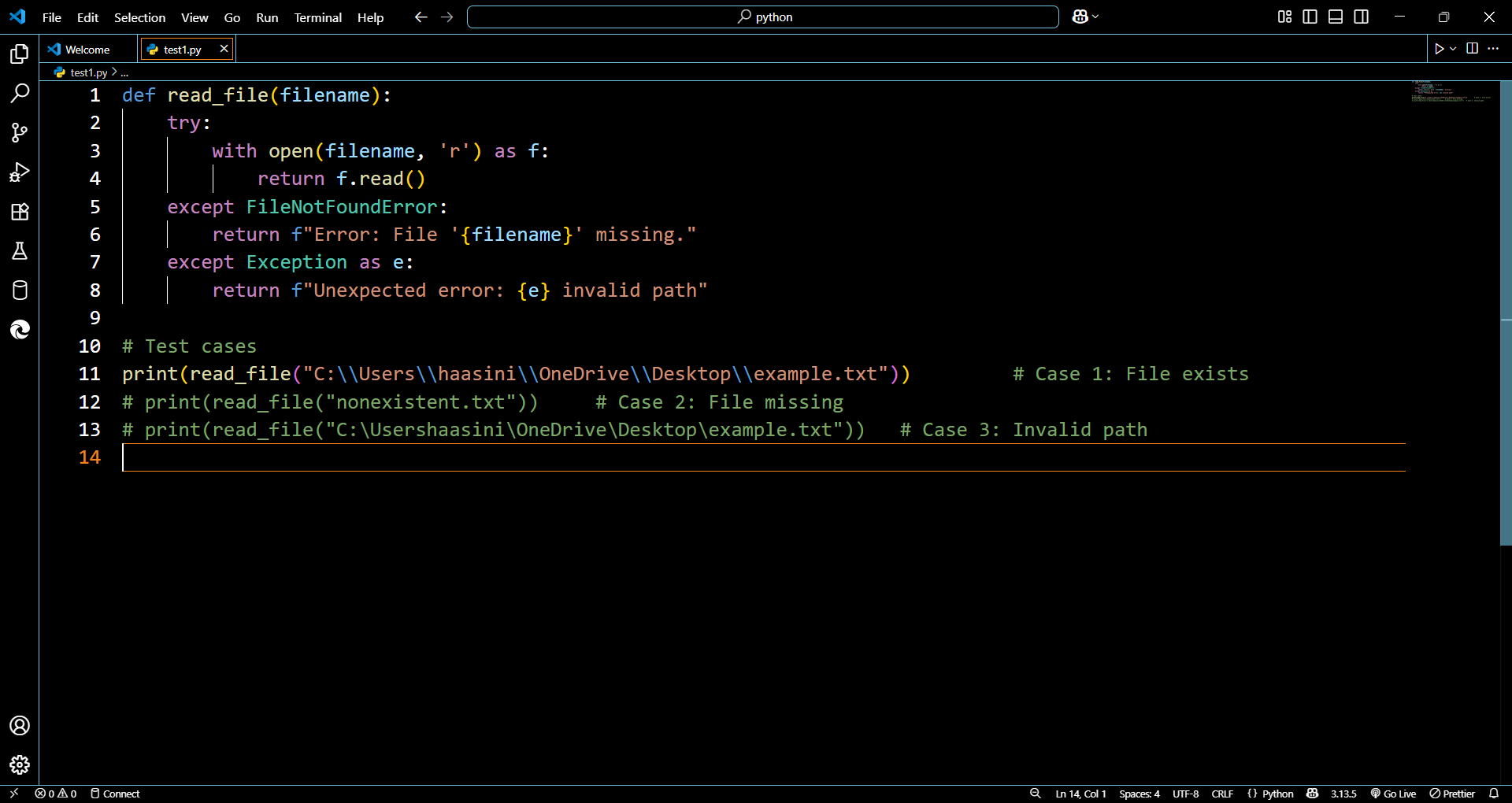
Code: 

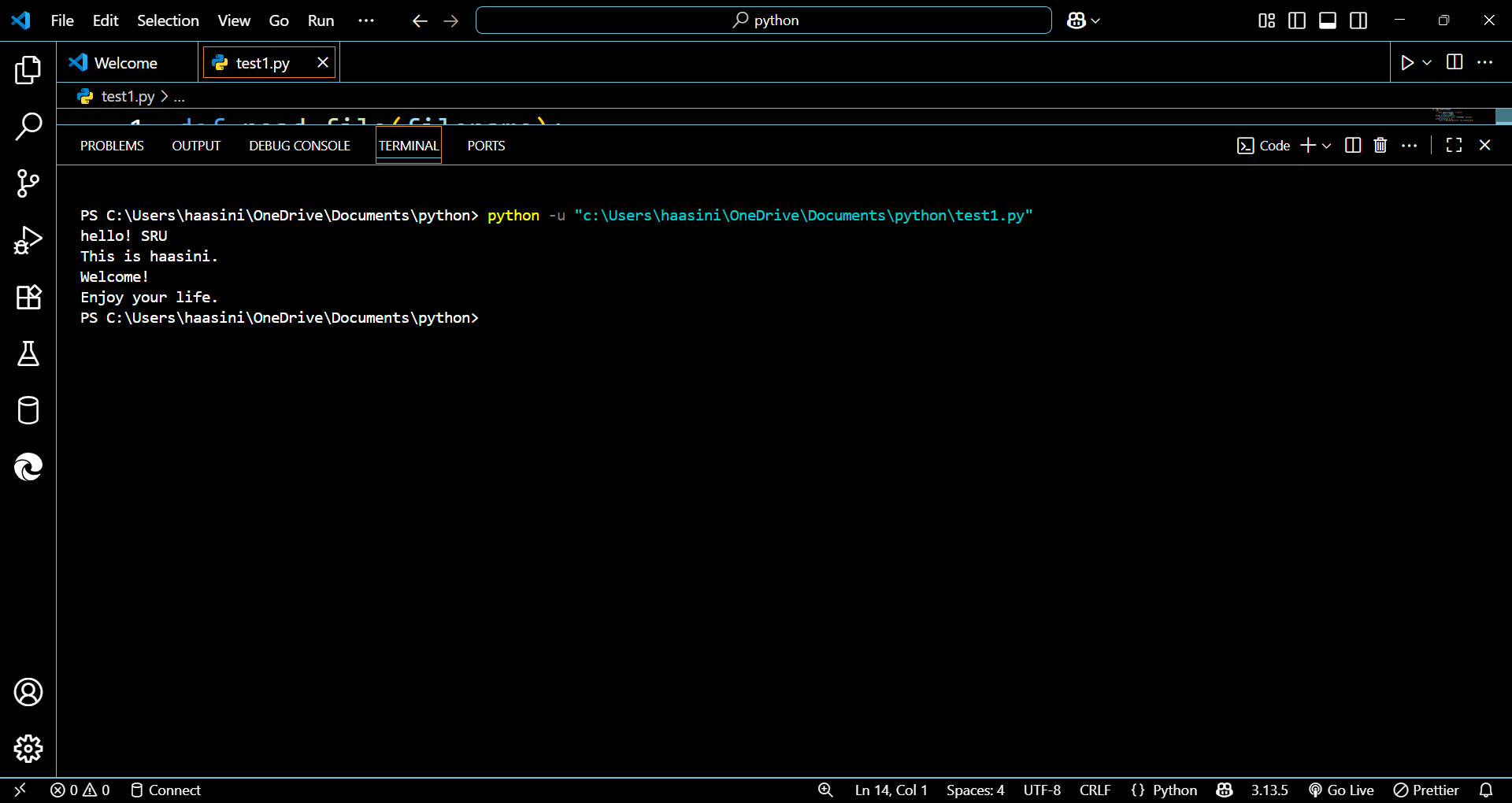


Test case 2

Reading the lines from the text

Code:





Text case 3 : file missing

Code:





**TASK 4:**

class Car:

    def start(self):

        return "Car started"

my\_car = Car()

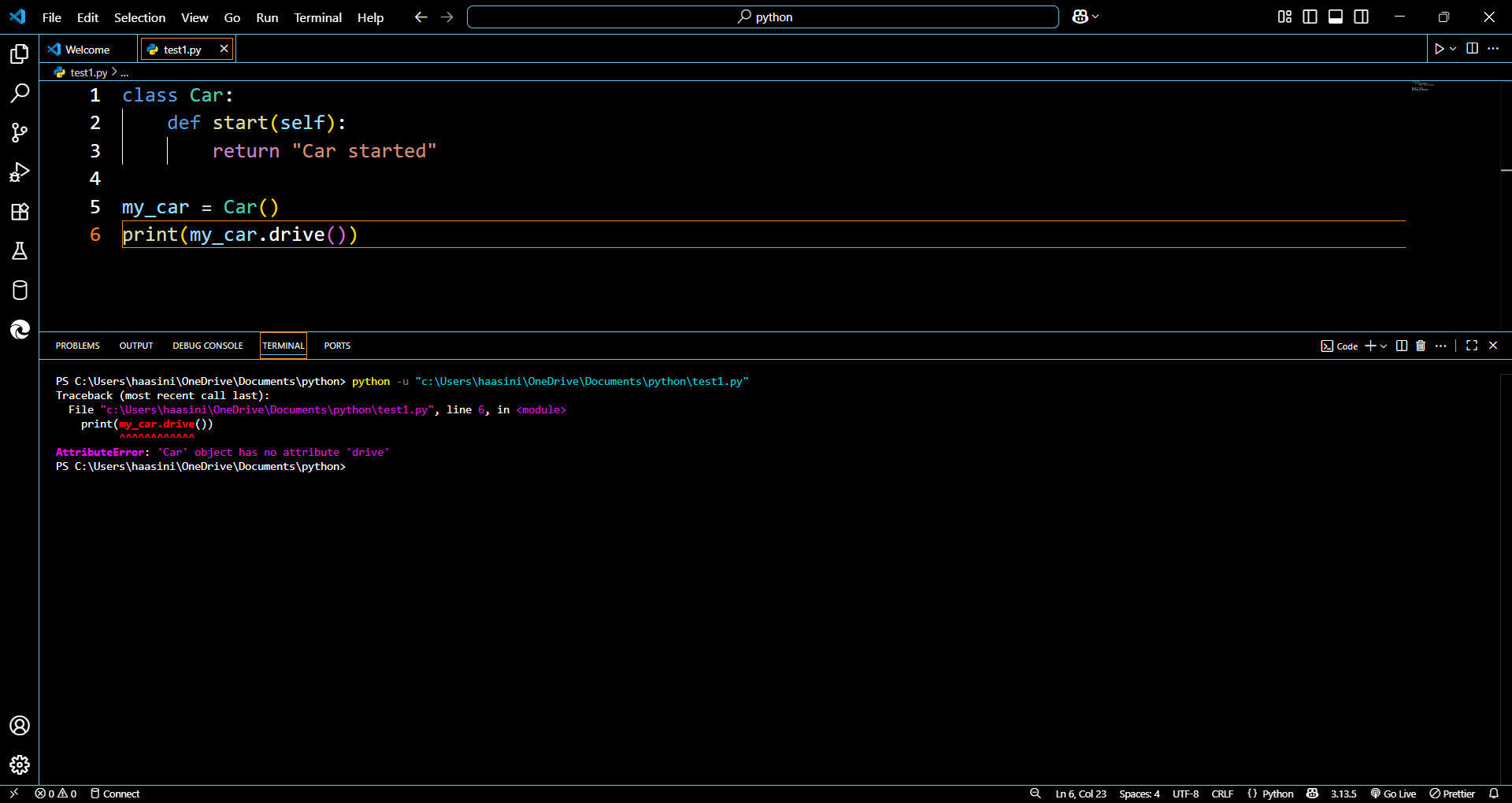
print(my\_car.drive())

debug and fix this code with 3 assert cases

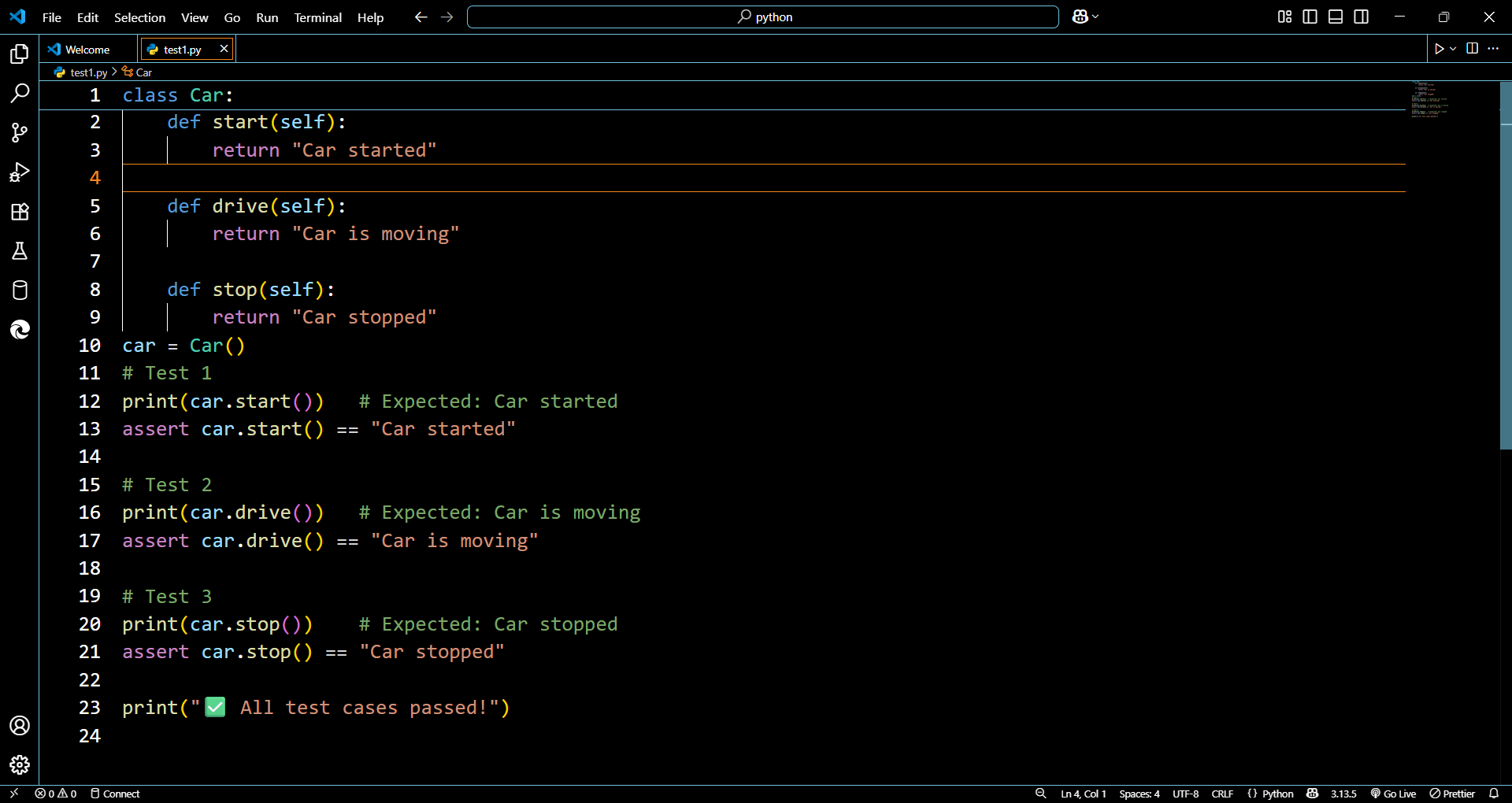
code:



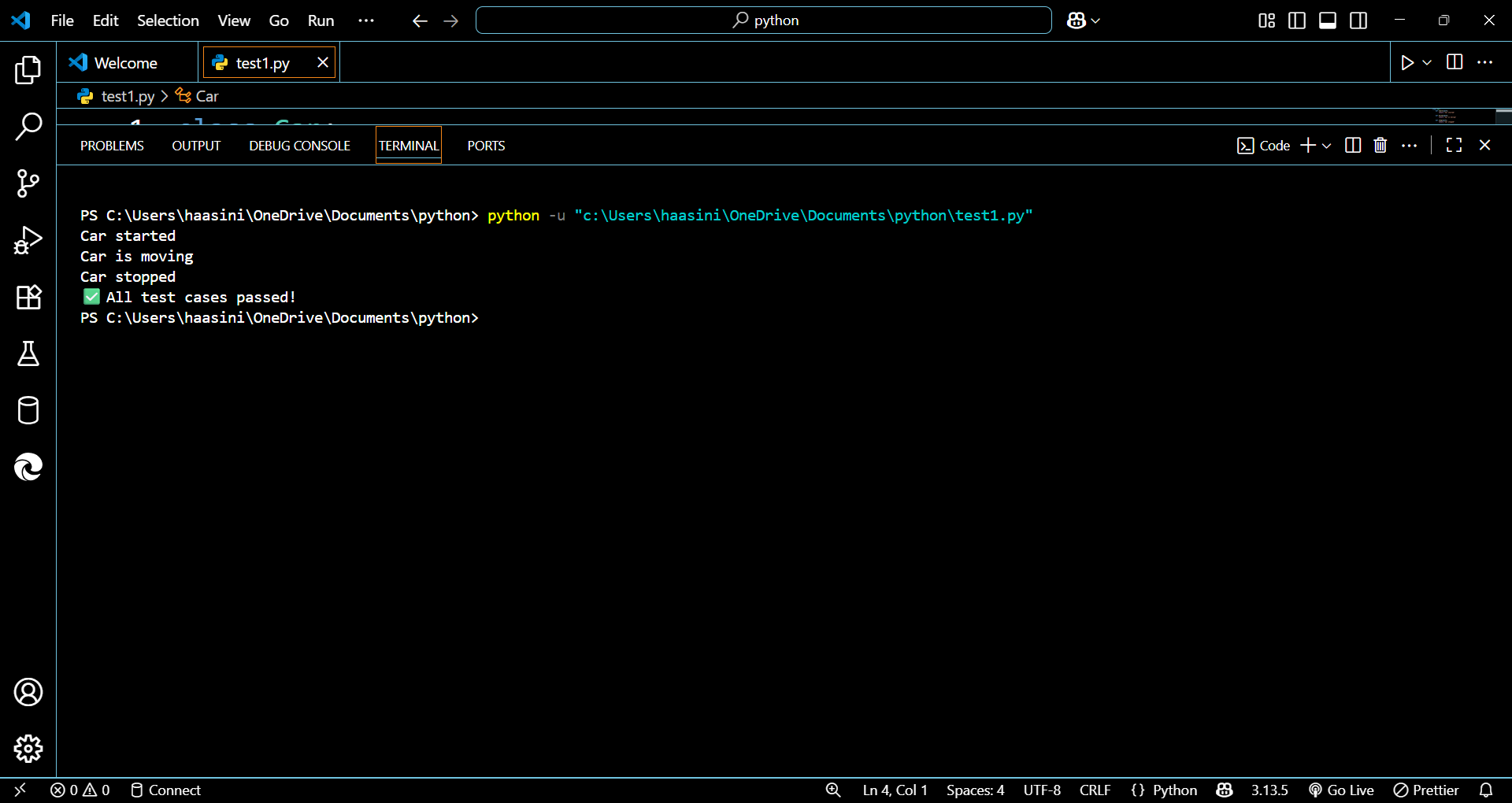
Error:



Fixed code:



Output:



**TASK 5:**

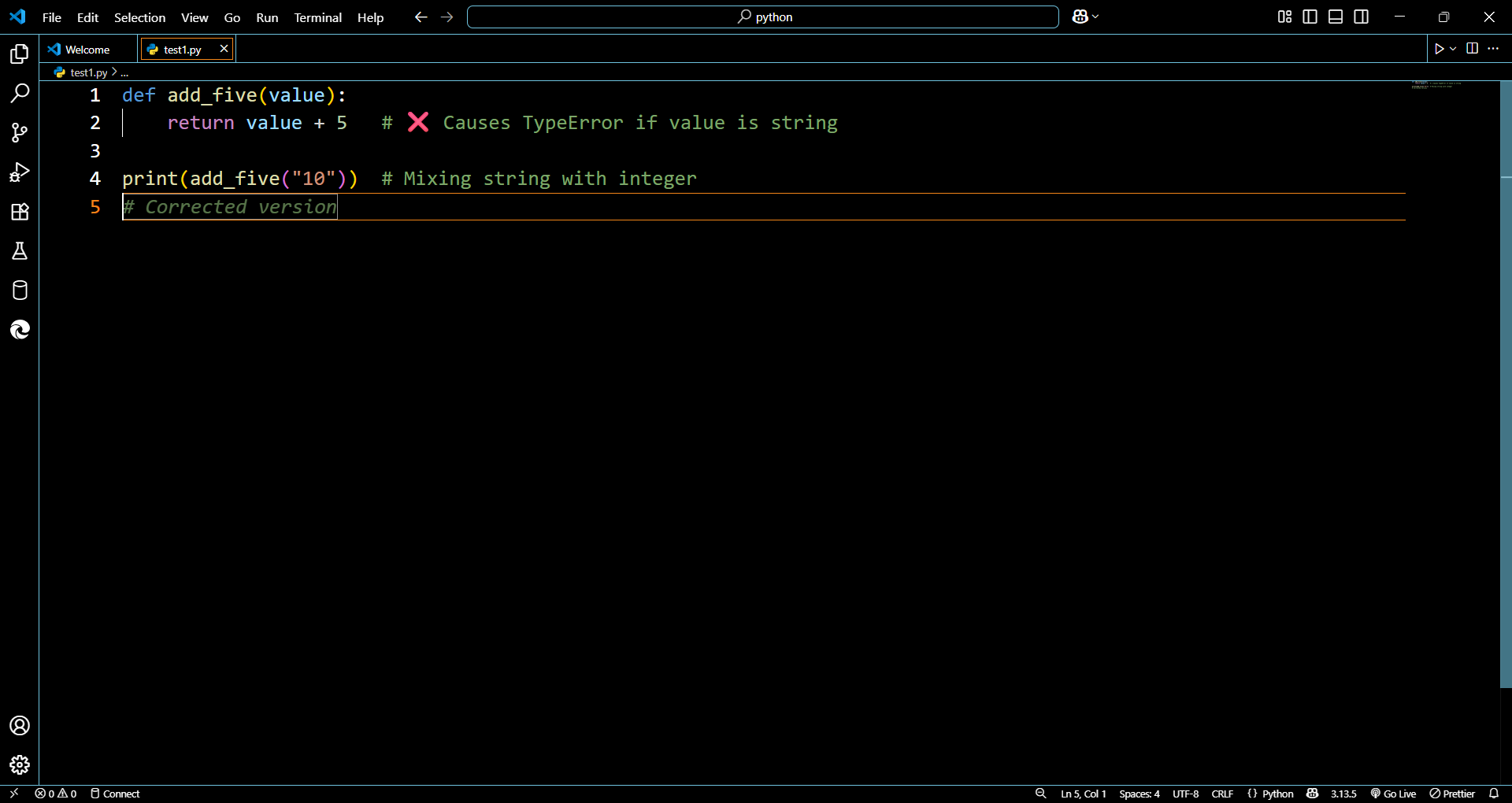
Prompt:

def add\_five(value):

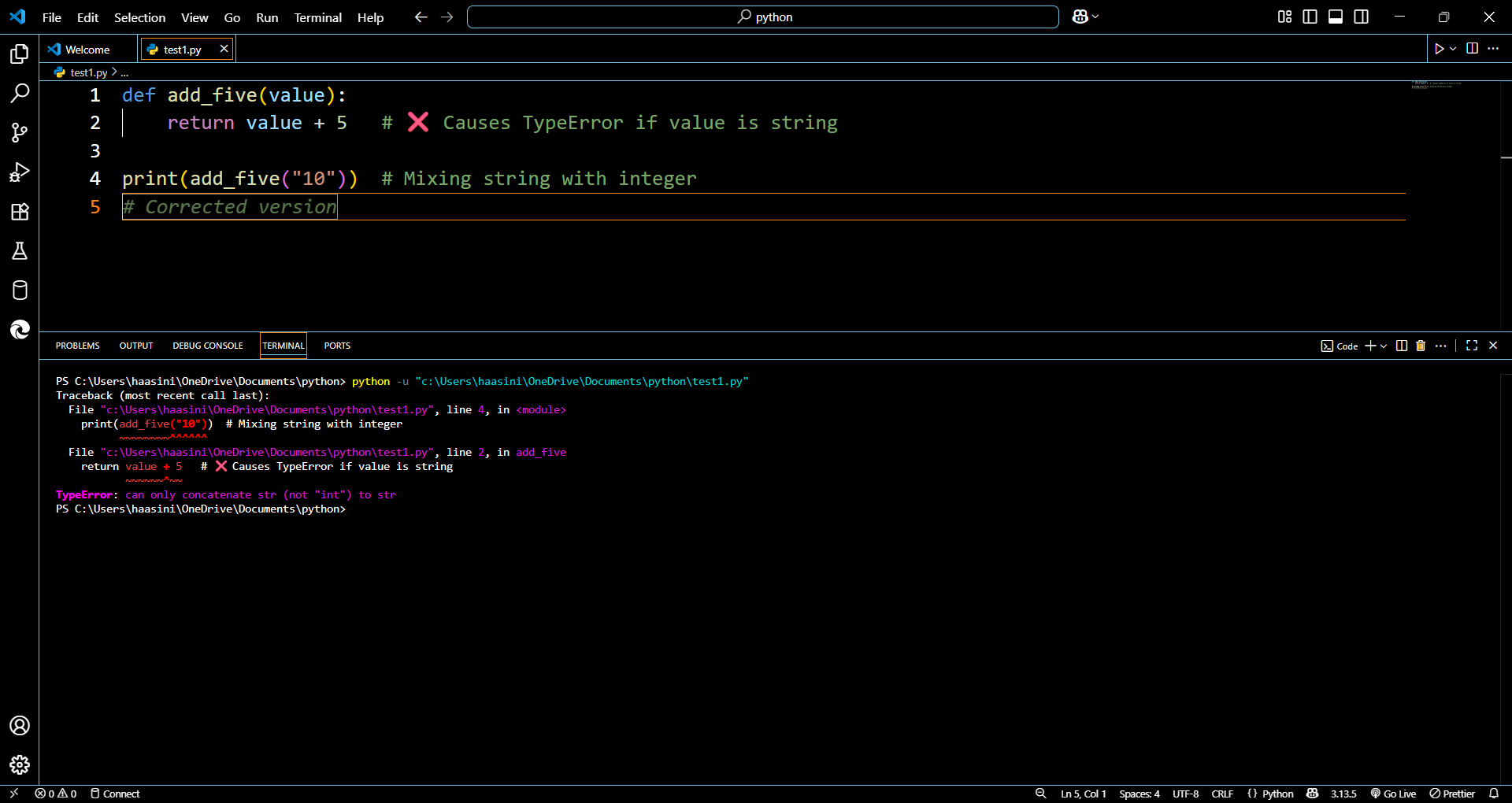
    return value + 5   # ❌ Causes TypeError if value is string

print(add\_five("10"))  # Mixing string with integer

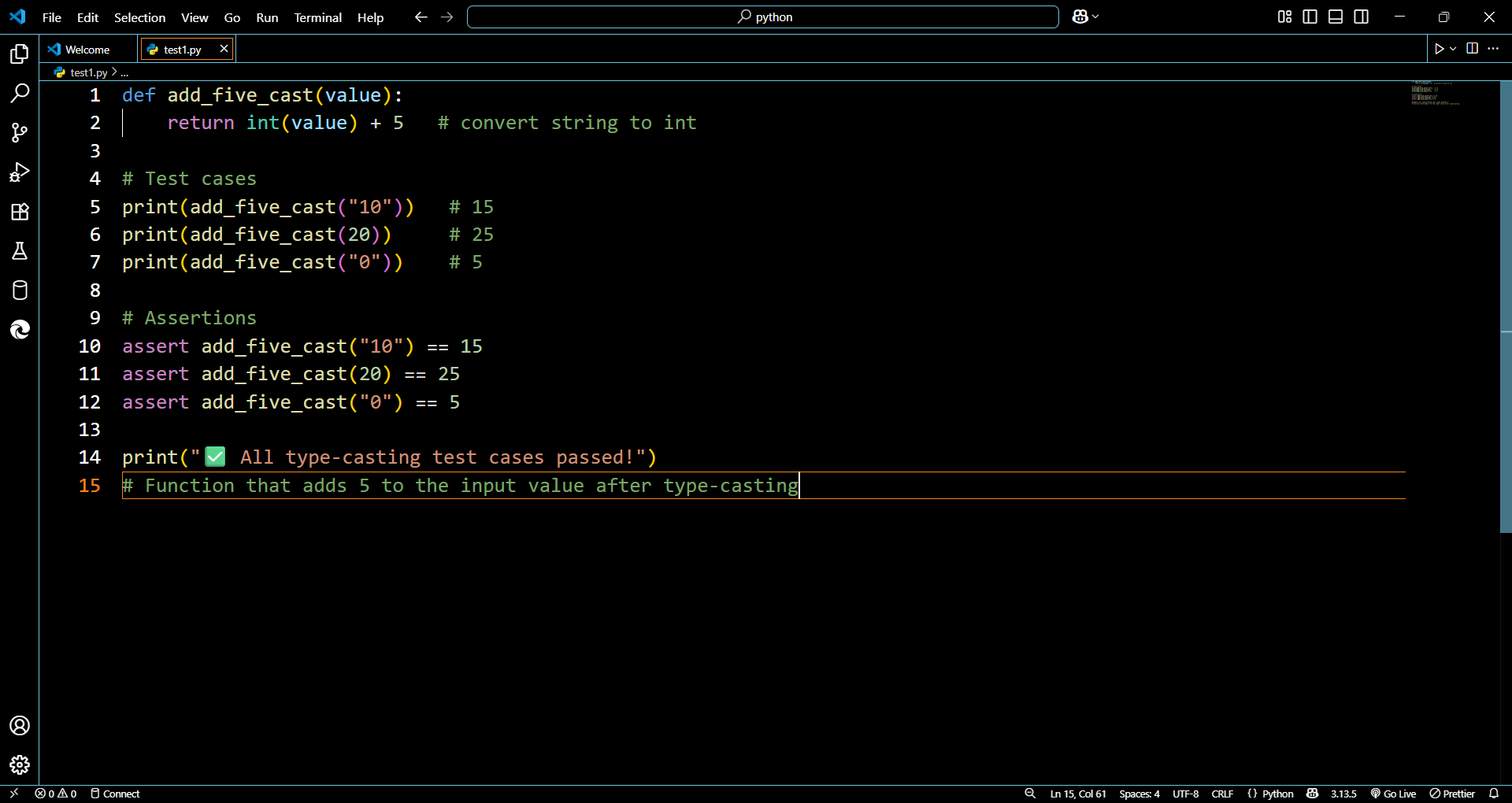
debug the erro and give the code with type casting and string concatenation.



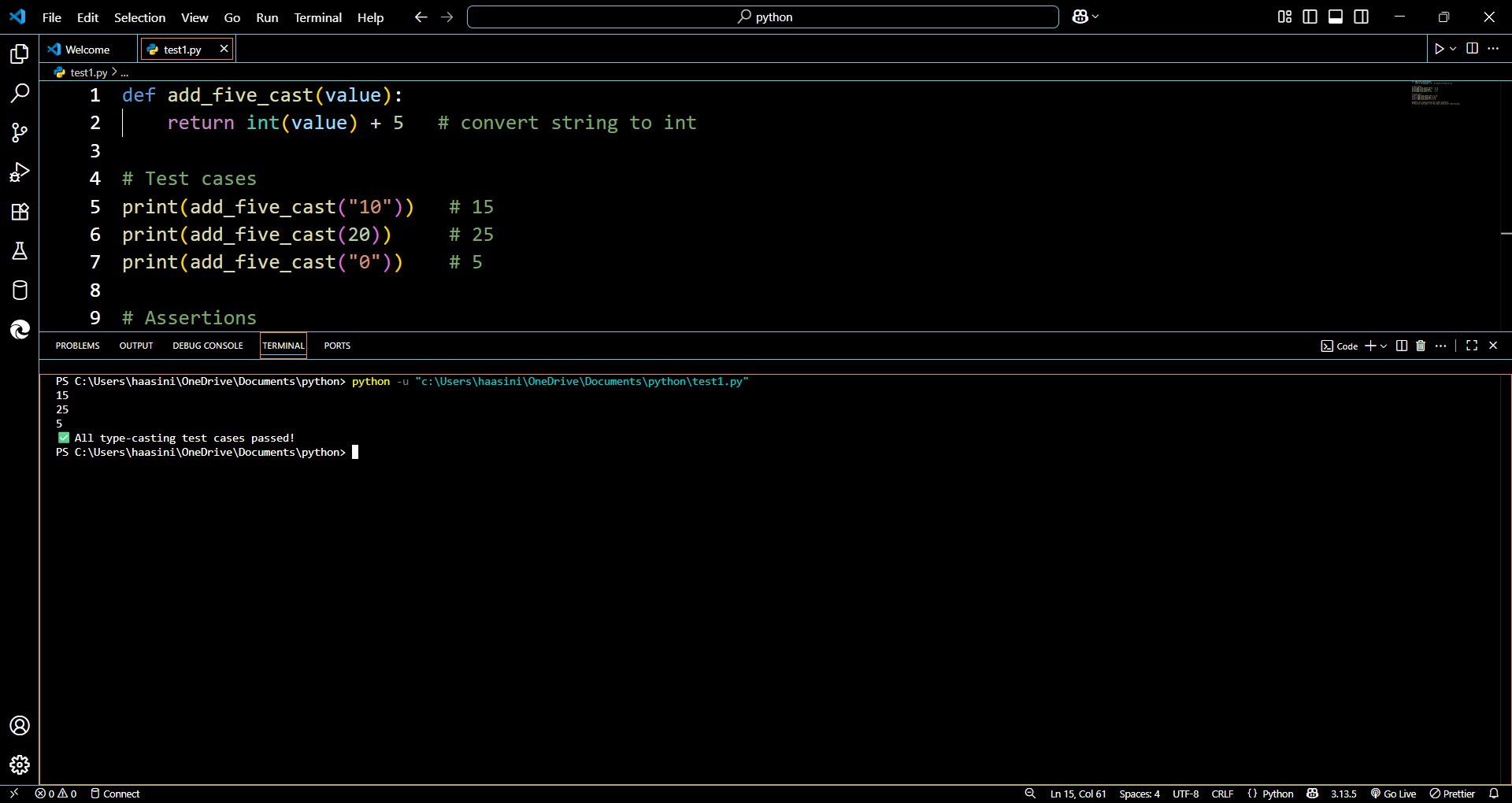
Error:



Fixed code type casting:

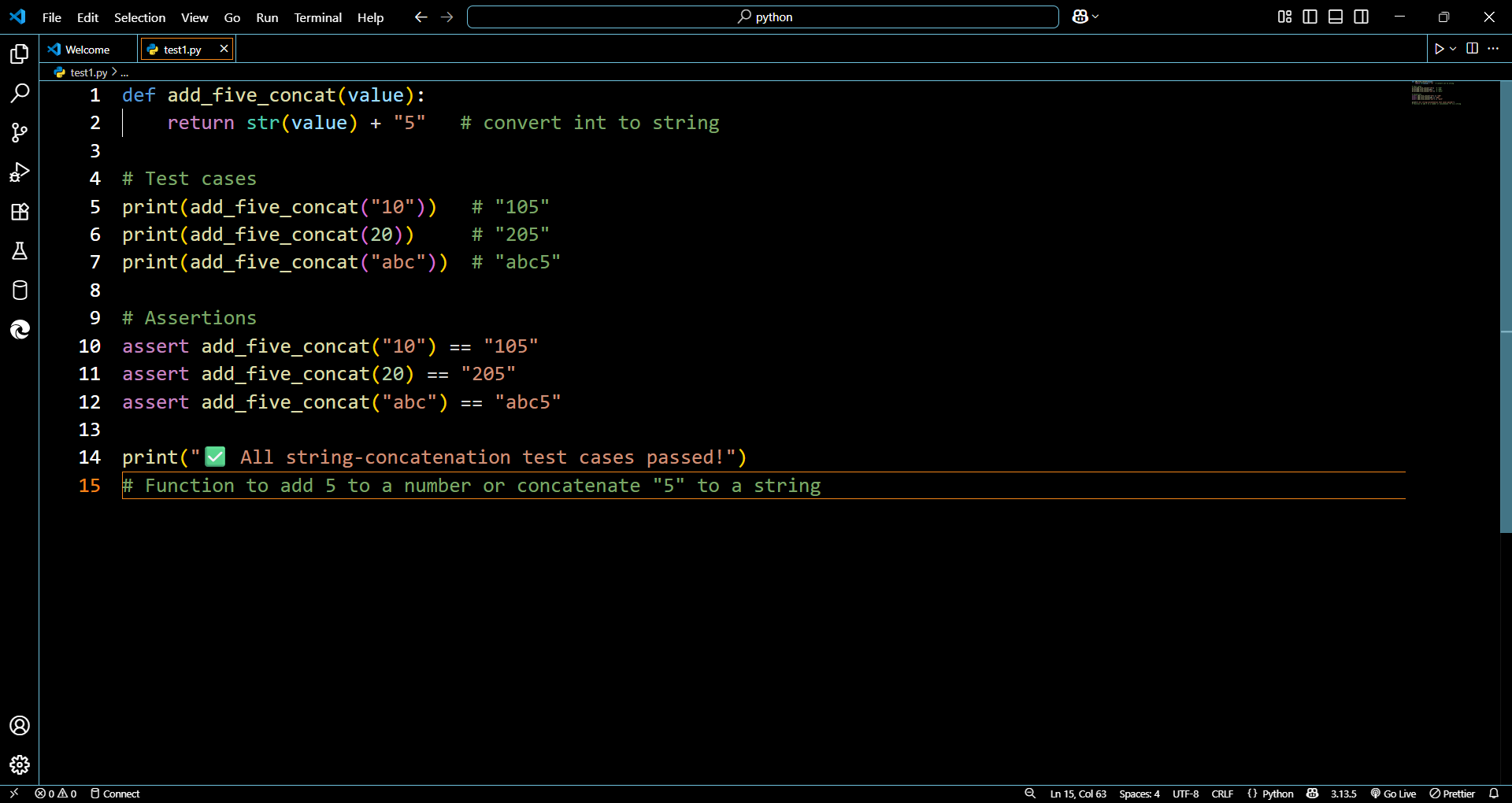


Output:



Fixed code 2 String concatenation

Code:



Output:

