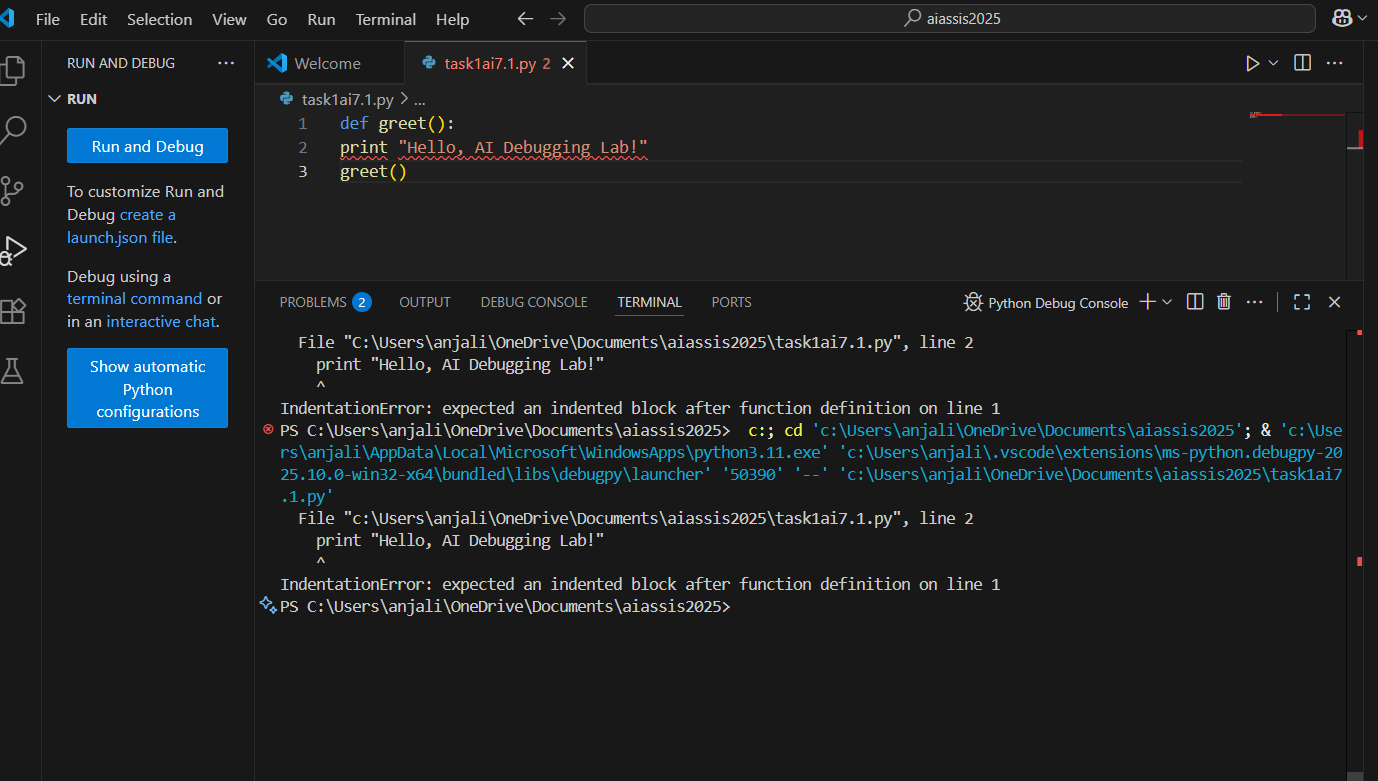
Assignment-7.1

TASK-1

Running given code:

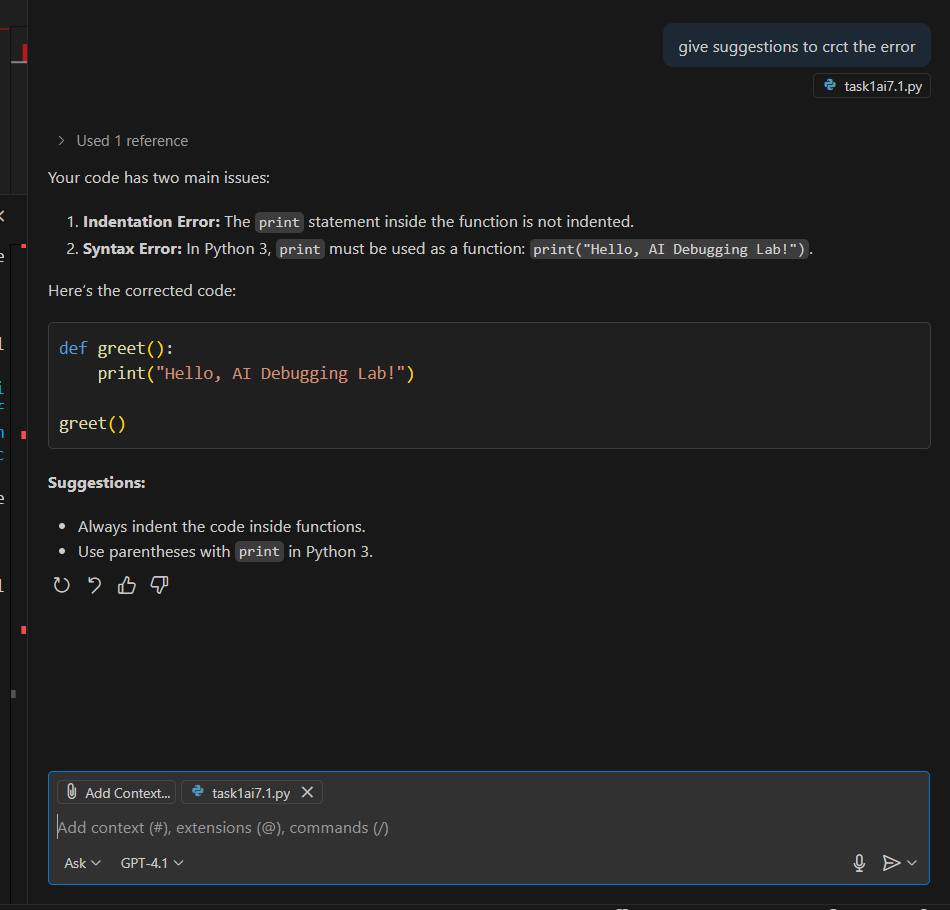


Errors:

1. **Indentation Error:** The print statement inside the function is not indented.
2. **Syntax Error:** In Python 3, print must be used as a function

**AI suggestions:**

* Always indent the code inside functions.
* Use parentheses with print in Python 3.



Corrected Syntax:

def greet():

    print ("Hello, AI Debugging Lab!")

greet()

A black screen with white text

AI-generated content may be incorrect.

OUTPUT:

Hello, AI Debugging Lab!

A black background with white text

AI-generated content may be incorrect.

Explanation of code:

* The function [greet()](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) prints the message "Hello, AI Debugging Lab!".
* The function is called, so the message is displayed.

Assert cases:

def greet():

    print ("Hello, AI Debugging Lab!")

greet()

assert "Hello" in "Hello, AI Debugging Lab!"

assert "Python" not in "Hello, AI Debugging Lab!"

assert len("Hello, AI Debugging Lab!") > 10

print("All tests passed ✅")

A screen shot of a computer

AI-generated content may be incorrect.

Output:

Hello, AI Debugging Lab!

All tests passed ✅

A screen shot of a computer

AI-generated content may be incorrect.

TASK-2

Logic Error – Incorrect Condition in an If Statement

Running given code:

A screenshot of a computer

AI-generated content may be incorrect.

Errors:

if n = 10:

^^^^^^

SyntaxError: invalid syntax. Maybe you meant '==' or ':=' instead of '='?

**SyntaxError** caused by using = instead of == in the if statement.

AI suggestions:

1. **Use == for comparison:**  
   Replace [if n = 10:](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) with [if n == 10:](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html).  
   = is for assignment, == is for comparison.

Corrected syntax:

def check\_number(n):

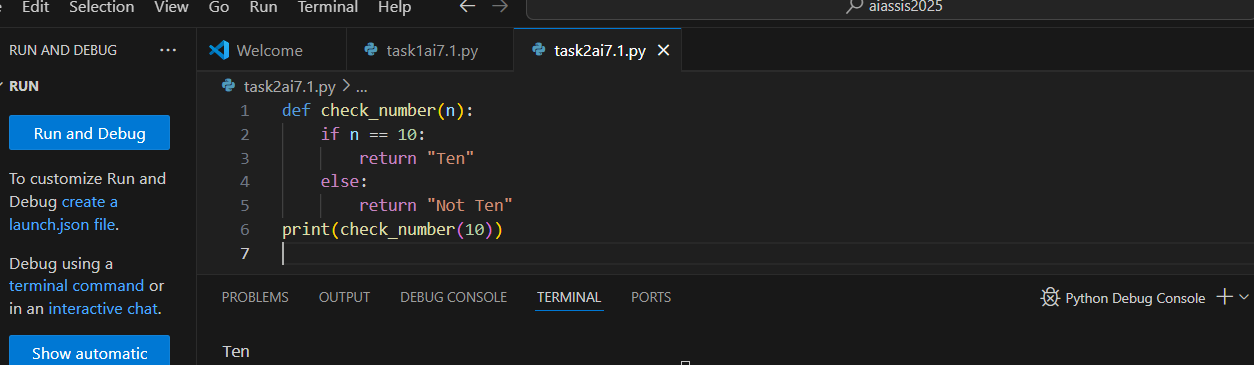
    if n == 10:

        return "Ten"

    else:

        return "Not Ten"

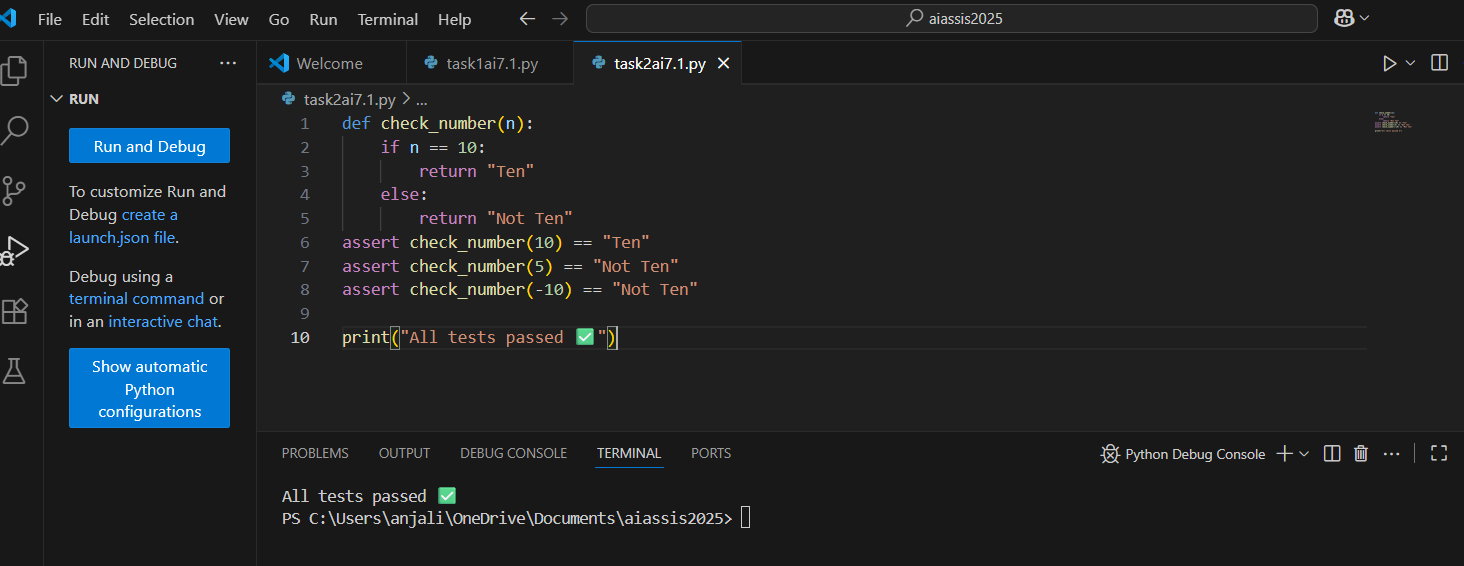
print(check\_number(10))



Output:

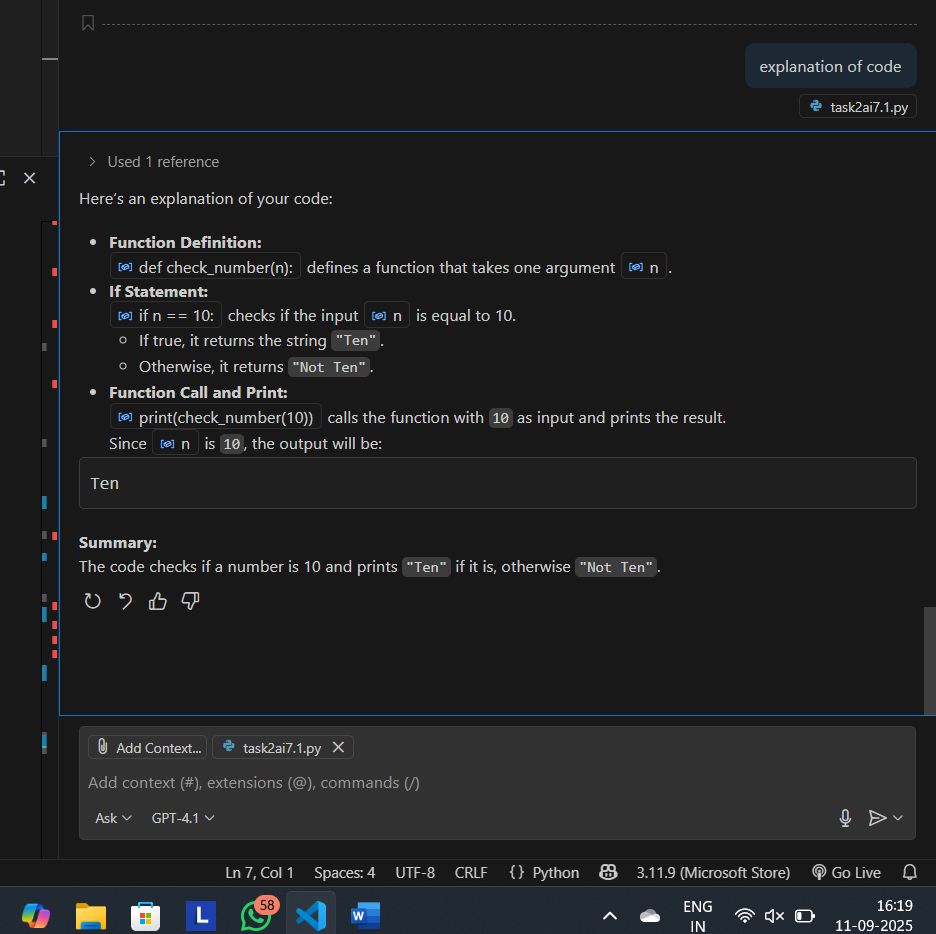
Ten

Assert cases:



AI explanation:

* **Function Definition:**  
  [def check\_number(n):](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) defines a function that takes one argument [n](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html).
* **If Statement:**  
  [if n == 10:](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) checks if the input [n](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) is equal to 10.
  + If true, it returns the string "Ten".
  + Otherwise, it returns "Not Ten".
* **Function Call and Print:**  
  [print(check\_number(10))](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) calls the function with 10 as input and prints the result.



TASK-3:

File Handling:

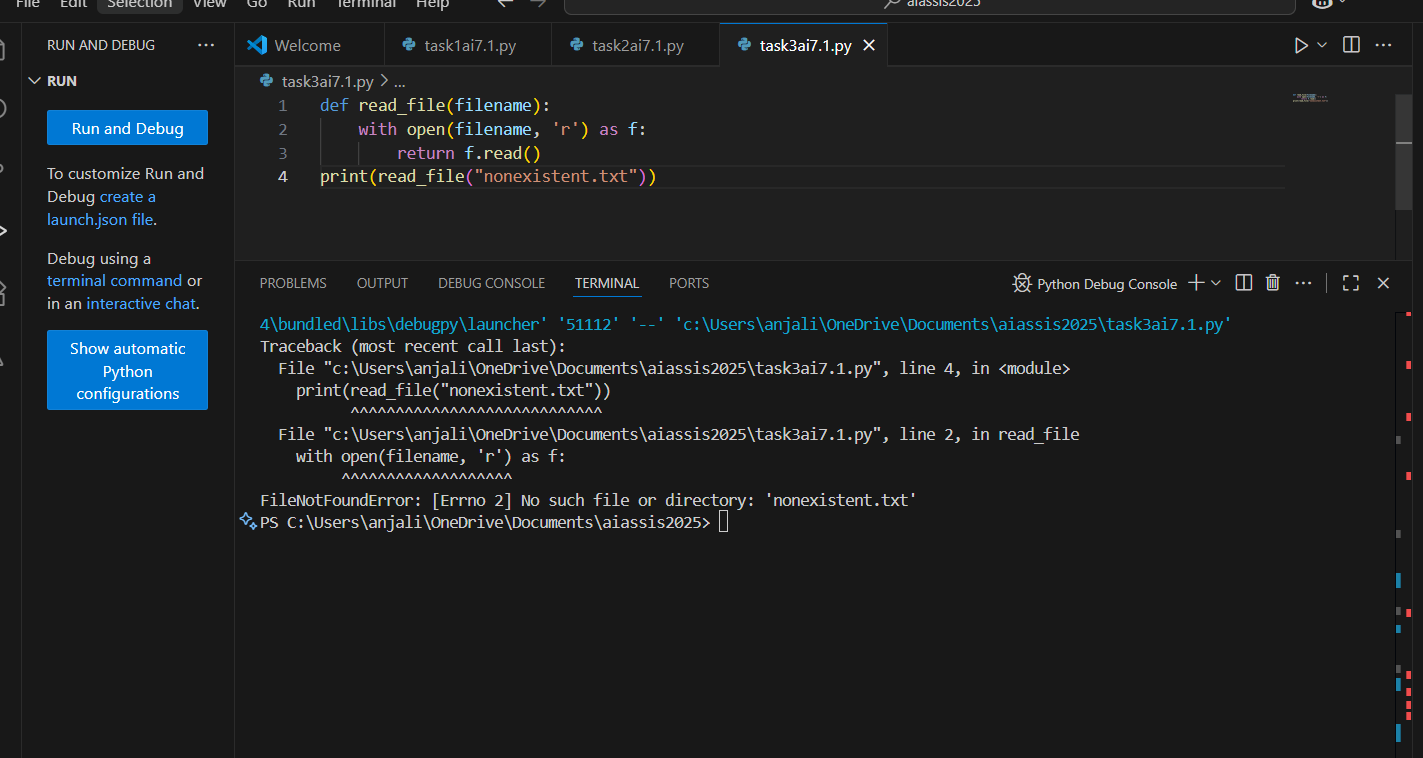
Running given code:

def read\_file(filename):

    with open(filename, 'r') as f:

        return f.read()

print(read\_file("nonexistent.txt"))



Errors:

* The function tries to open "nonexistent.txt" for reading.
* Since the file does **not exist**, Python raises a FileNotFoundError.
* There is **no try-except block** to handle this error.
* As a result, the program will crash and display an error message

Suggestions:

**Add a try-except block for user-friendly error handling**

Prompt used:

Implement a try-except block and Add a user-friendly error message on

def read\_file(filename):

    with open(filename, 'r') as f:

        return f.read()

print(read\_file("nonexistent.txt"))

Code given by AI:

def read\_file(filename):

try:

with open(filename, 'r') as f:

return f.read()

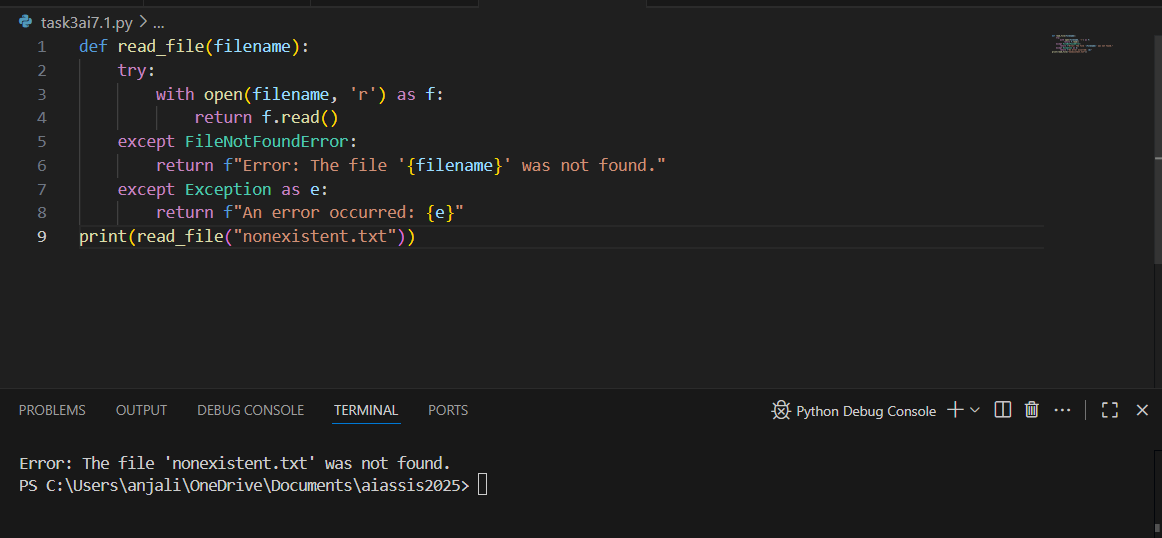
except FileNotFoundError:

return f"Error: The file '{filename}' was not found."

except Exception as e:

return f"An error occurred: {e}"

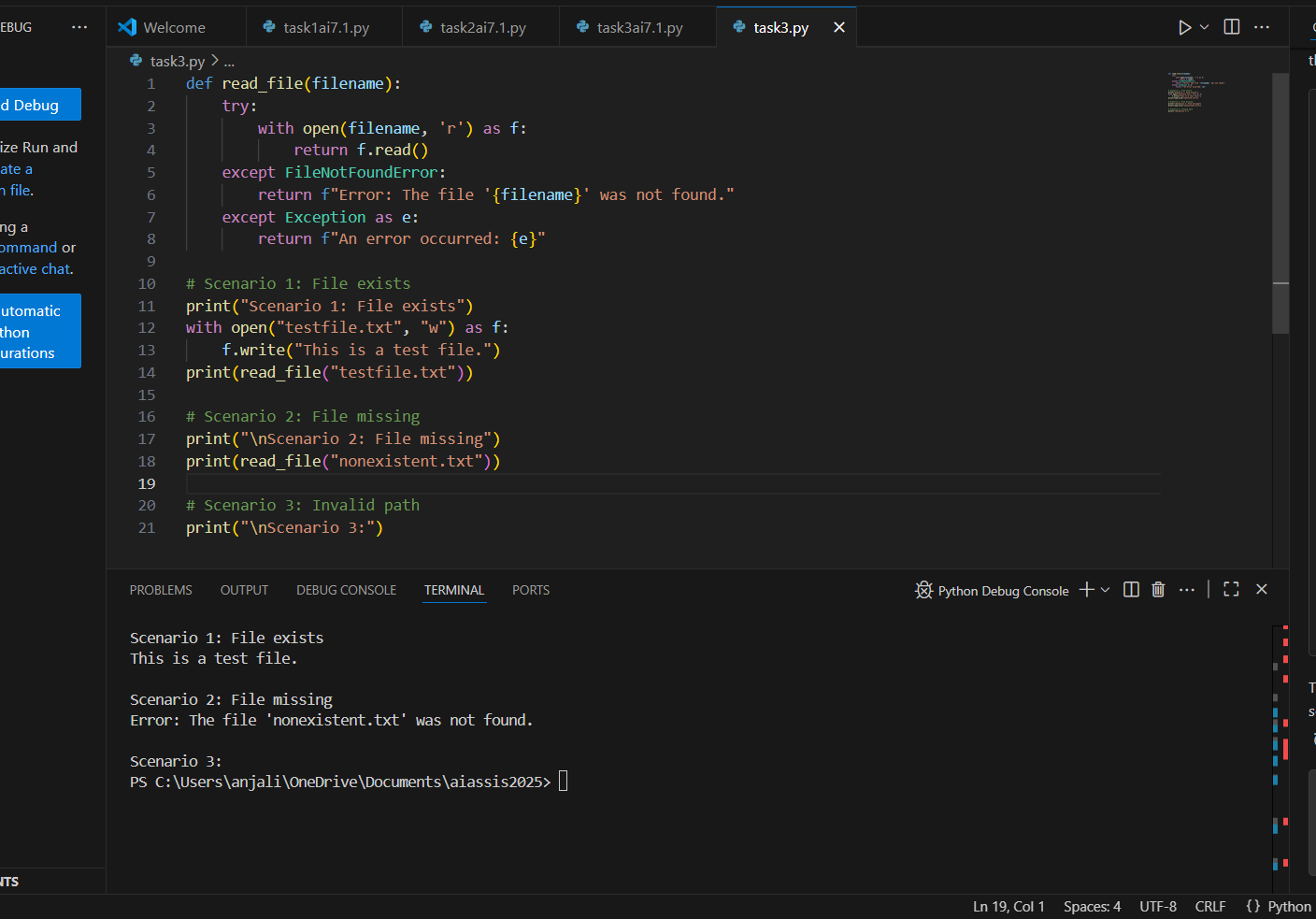
Execution:



Output:

Error: The file 'nonexistent.txt' was not found.

Scenarios:



TASK-4:

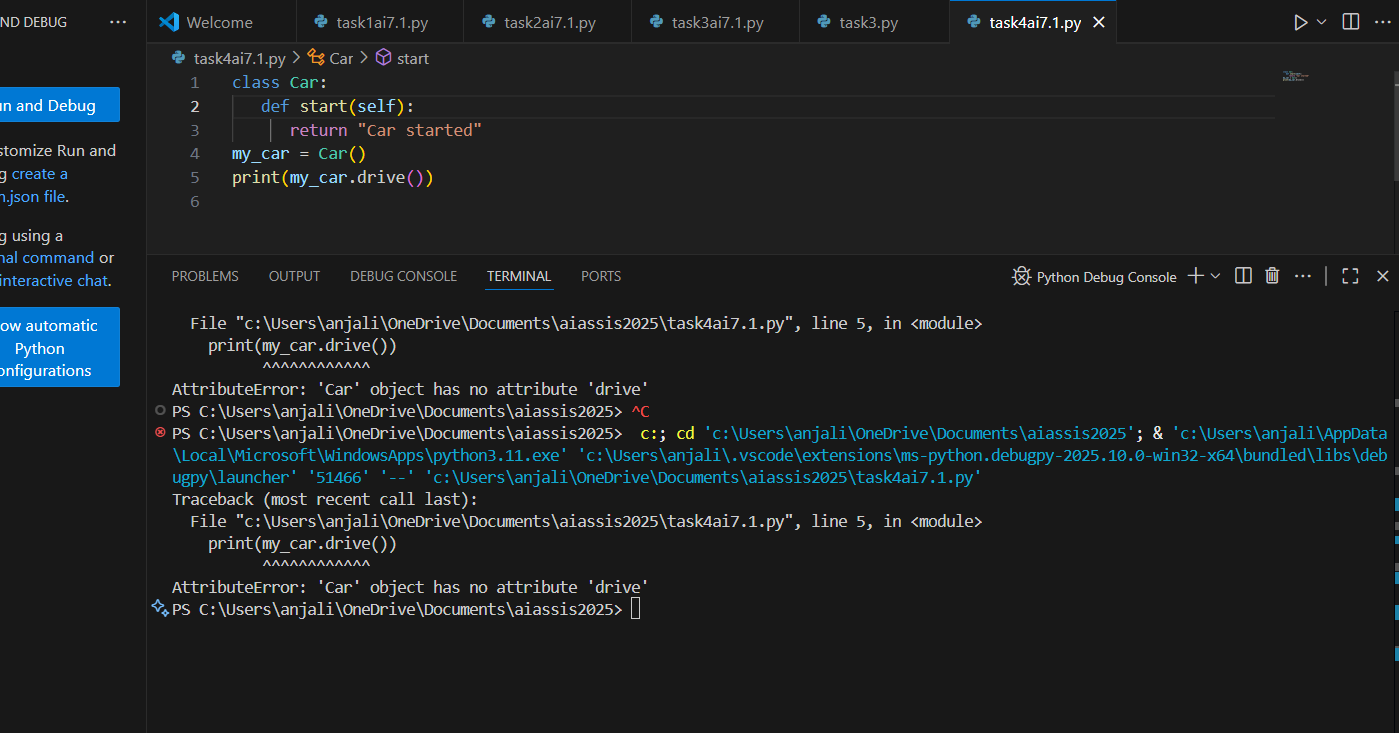
AttributeError – Calling a Non-Existent Method

Running given code:

class Car:

def start(self):  
 return "Car started"

my\_car = Car()  
print(my\_car.drive())



Error:

 AttributeError because the [Car](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) class does **not** have a drive method. You are trying to call [my\_car.drive](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html), but only [start](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) is defined

Analyzation:

It is better to **correct the method call** to match the methods defined in your class.

Prompt used:

debug and fix.# Bug: Calling an undefined method  
class Car:  
def start(self):  
return "Car started"  
my\_car = Car()  
print(my\_car.drive())

Corrected code:

class Car:

   def start(self):

      return "Car started"

my\_car = Car()

result =my\_car.start()

print(result)

A screenshot of a computer

AI-generated content may be incorrect.

Output: Car started

**Explanation:**  
This code defines a [Car](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) class with a [start](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) method that returns the string "Car started".  
An instance of [Car](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) is created and the [start](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) method is called, printing the result.

Assert tests:

class Car:

   def start(self):

      return "Car started"

my\_car = Car()

result =my\_car.start()

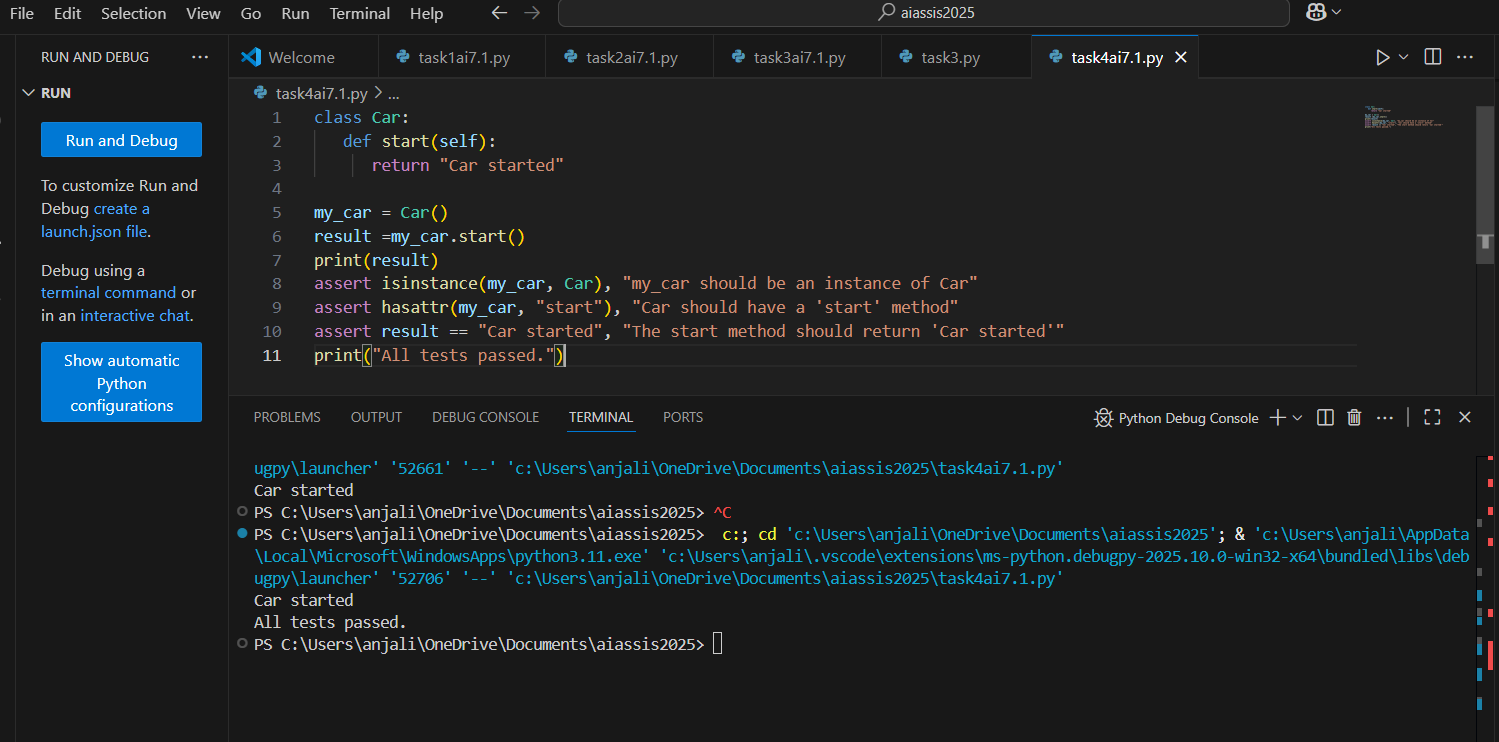
print(result)

assert isinstance(my\_car, Car), "my\_car should be an instance of Car"

assert hasattr(my\_car, "start"), "Car should have a 'start' method"

assert result == "Car started", "The start method should return 'Car started'"

print("All tests passed.")



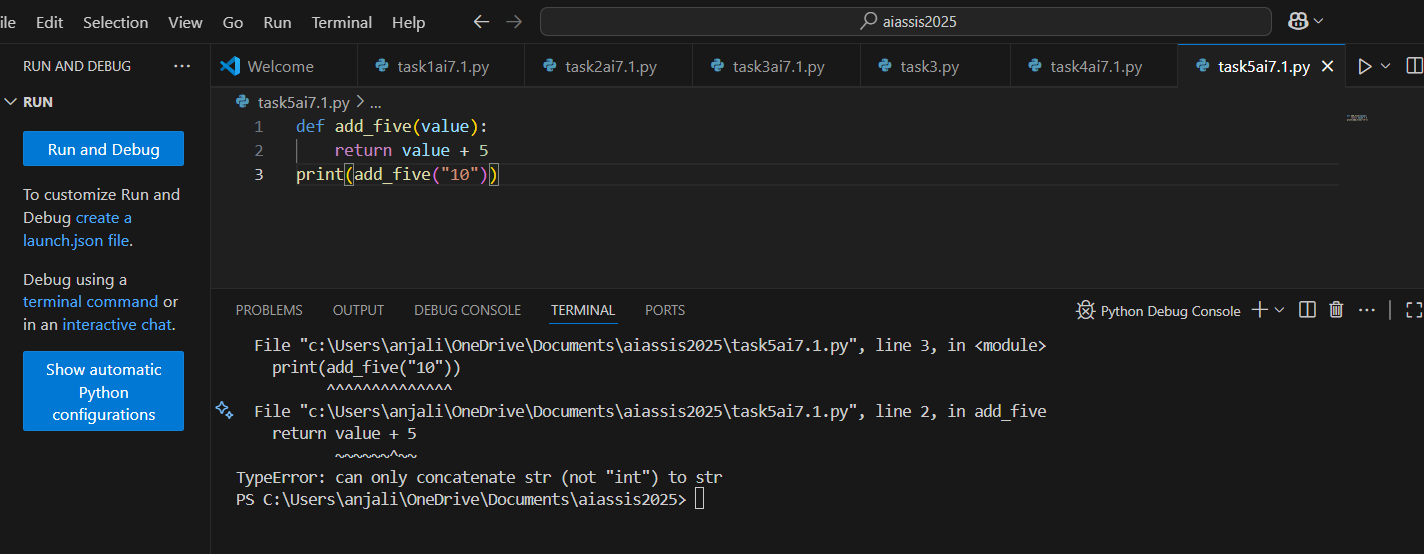
TASK-5

TypeError – Mixing Strings and Integers in Addition

Prompt used:

Resolve the given code and suggest the two correct codes using typecasting and string concatenation that takes multiple inputs in both cases.

def add\_five(value):  
return value + 5  
print(add\_five("10"))

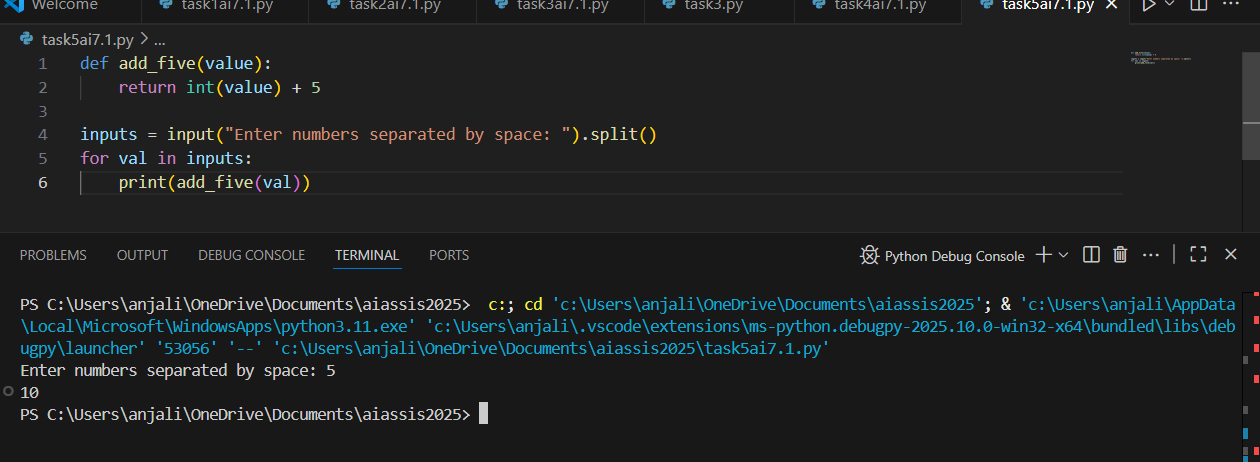


Error:

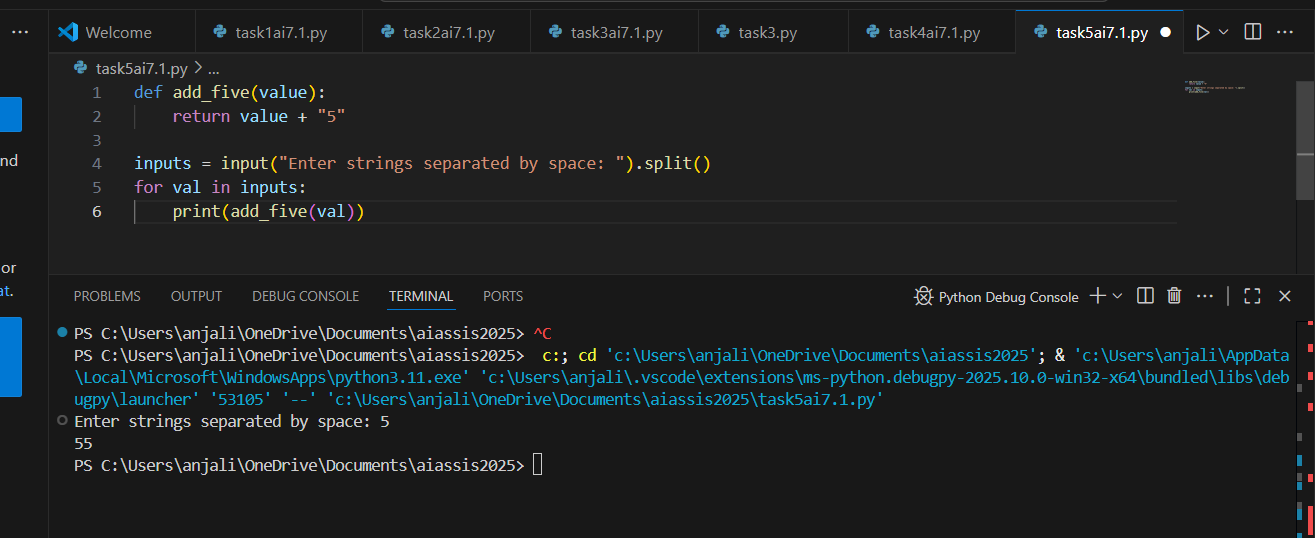
TypeError: Trying to add an integer (5) to a string ("10")

Correction of code:

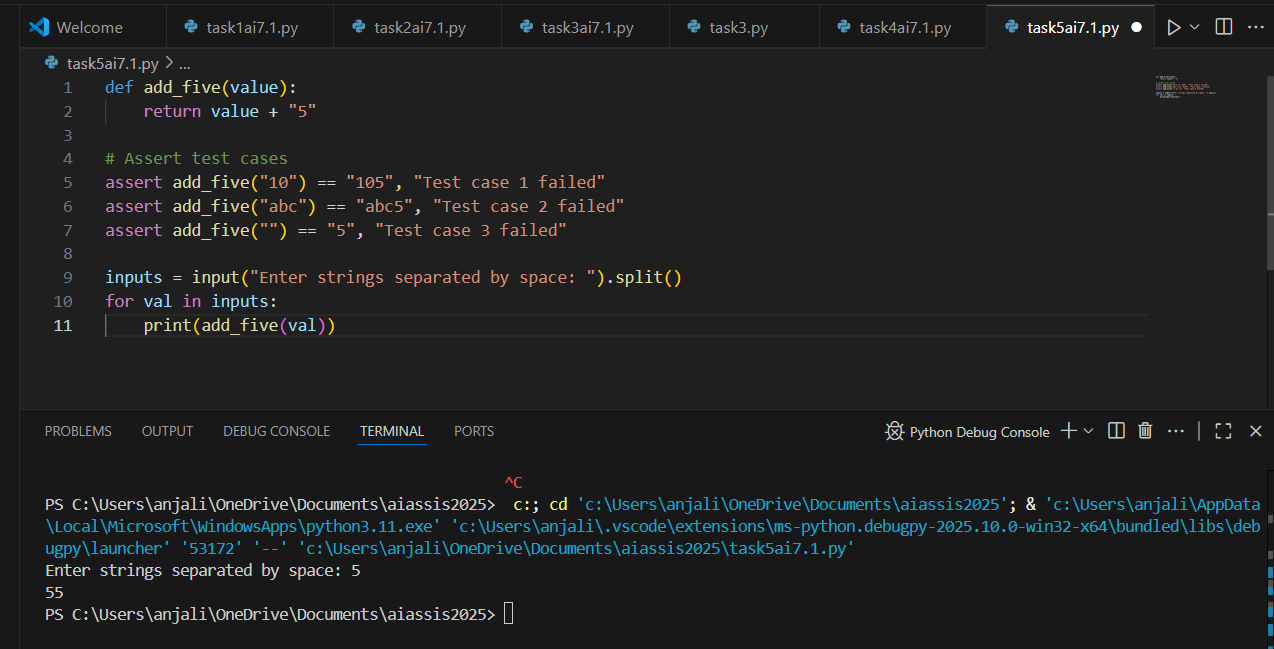
**1.Using typecasting (convert input to int and add 5):**



**2.Using string concatenation (append "5" to each input string):**



Assert Cases:



AI explanation :

* The function [add\_five](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) takes one argument, [value](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html).
* Inside the function, [value](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) is converted from a string to an integer using [int(value)](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html).
* It then adds 5 to this integer and returns the result.
* [print(add\_five("10"))](vscode-file://vscode-app/c:/Users/anjali/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) calls the function with the string "10", so it becomes 10 + 5, which is 15.