**ASSIGNMENT 02**

**SOFTWARE UNIT TESTING REPORT**

**SOFTWARE ENGINEERING:   
PROCESS AND TOOLS**

**S223 PRT582**

**MY NGOC DOAN – S370857**

Table of Contents

[INTRODUCTION 1](#_Toc144027797)

[PROCESS 3](#_Toc144027798)

[CONCLUSION 9](#_Toc144027799)

# INTRODUCTION

The requirement is building a program called “Guess the Number game” in Python following the game rules: The player needs to guess a four-digit randomly generated number. The program will then give some clues about the number. Once the player guesses the number correctly, the program will show the number of attempts taken.

The process to build this game has steps:

1. Create the number that the player must guess (we can call it the “correct number”).
2. The player types the answer directly from the terminal.
3. Whenever receiving any input from a user, the program must check if the input is valid or not. In this case, the input is a four-digit number.
4. After comparing the number of player guessing, if the answer is not correct, the computer will show a hint for the player to continue to play.
5. The hint is based on the rule: if any digit in is correct and is in the right position, it will be marked as a circle symbol (“O”), if that digit is correct but not in the right position, it will be marked as a cross symbol (“X”). So, if any digit is not correct, it won’t be shown again.
6. Each time the player makes a guess, the program will count it. The total number of attempts will be shown.
7. The player can continue to play the game or quit it anytime by inputting “0” (number zero)

The working process of this game is illustrated in this diagram:

Show attempt times

Show hint

Show attempt times

correct

incorrect

Quit

Input number

Create 4-digit number

Check valid input

Compare the result

Give hint

Couting attemp

4-digit number

“0”

In this case, I use the Test Driven Development as an automated unit testing tool to create this program.

The Test Driven Development (TDD) is a testing tool that needs the test to be written before writing the actual code. The TDD starts with every small function of any program. Every problem must be checked separately. Using TDD means we must think about all the cases related to every function that can happen, so we write the test first to have a clear idea about the problem we are solving so we will have a clue to write the code about it. We continue using the test to determine whether the code is correct or not. If there are mistakes found, we must go back and fix the code until all the cases of the test can be solved successfully. All those steps repeat until all the function codes are written and all the test cases are checked carefully.

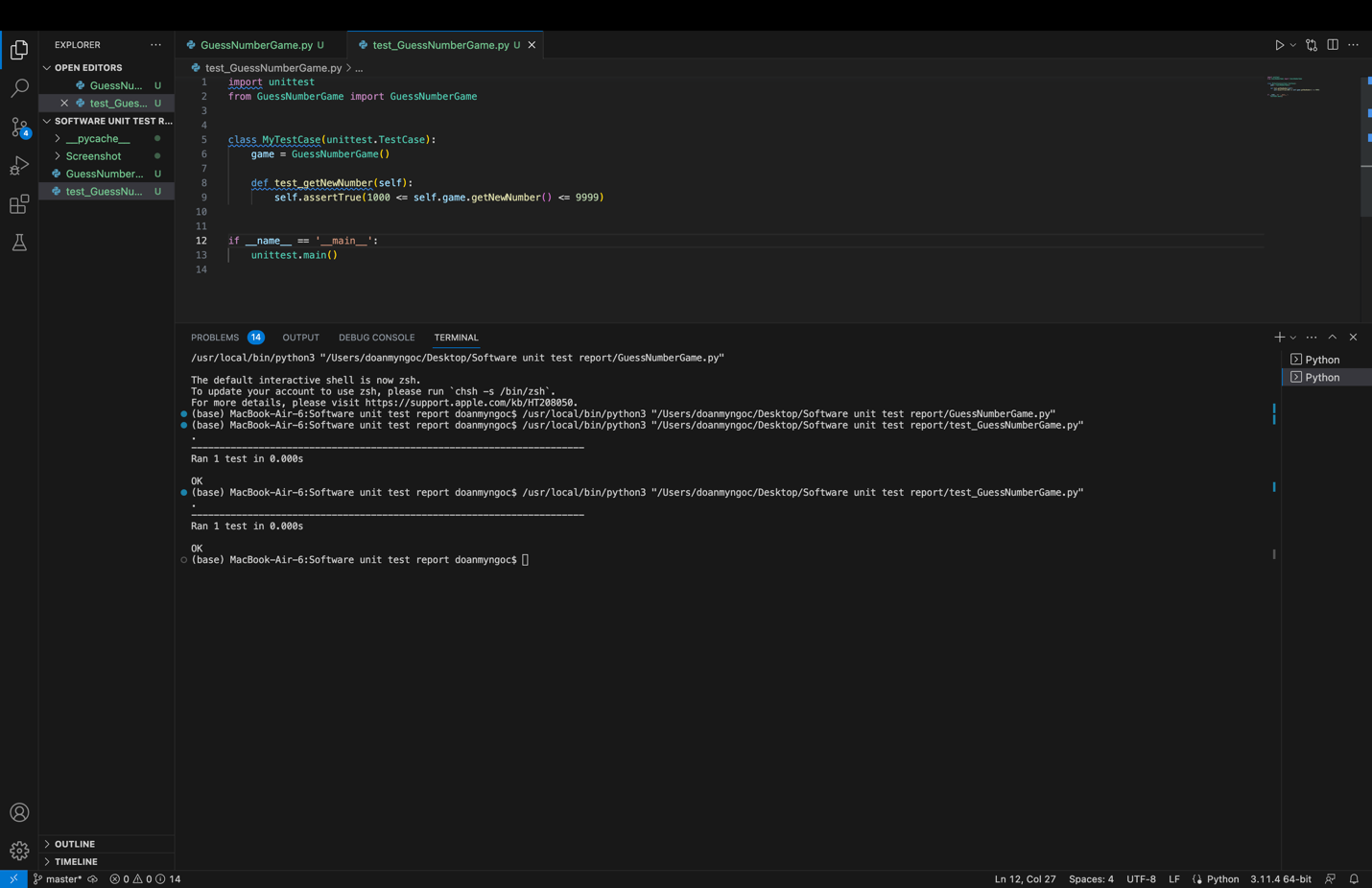
Of course, there are many benefits when using TDD, the most important is that it helps improve the code quality. Writing the test case helps the writer understand more about the requirements of the program and have a clear view of how the program should run, and how each function should behave. So the quality of the code will be higher. Besides, when we check every single function while we are writing it, that prevents wasting too much time to find the problem when the program gets the result which is not as same as we expect. If we don’t use TDD, we write all the code for the program from the beginning to the end and after that, we start to test it. If it doesn’t work as expected, we will waste too much time to find where the bug is. The more complicated the program is, the more time we need to find bugs. So TDD can prevent this from happening.

# PROCESS

Using TDD to create the Guess the Number game is dividing the program into small functions, each function has to be tested and fixed if the code doesn’t run as expected.

**Step 1:** Create a random four-digit number.

The number has to meet the requirement of being from 1000 to 9999.

****

**Step 2:** Checking valid input from the user:

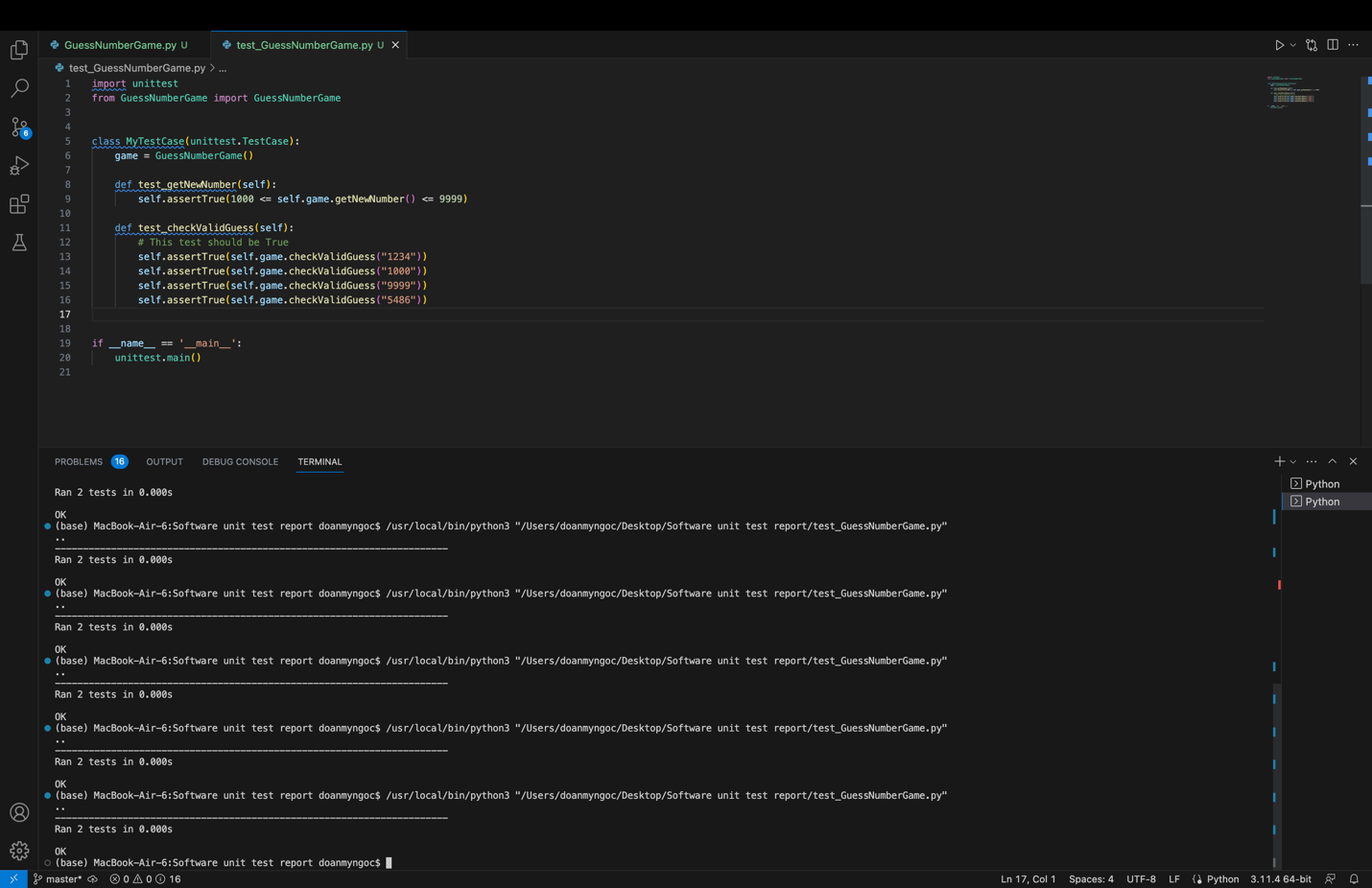
In this step, the user can input the number directly, but the wrong input can cause a problem with a program.

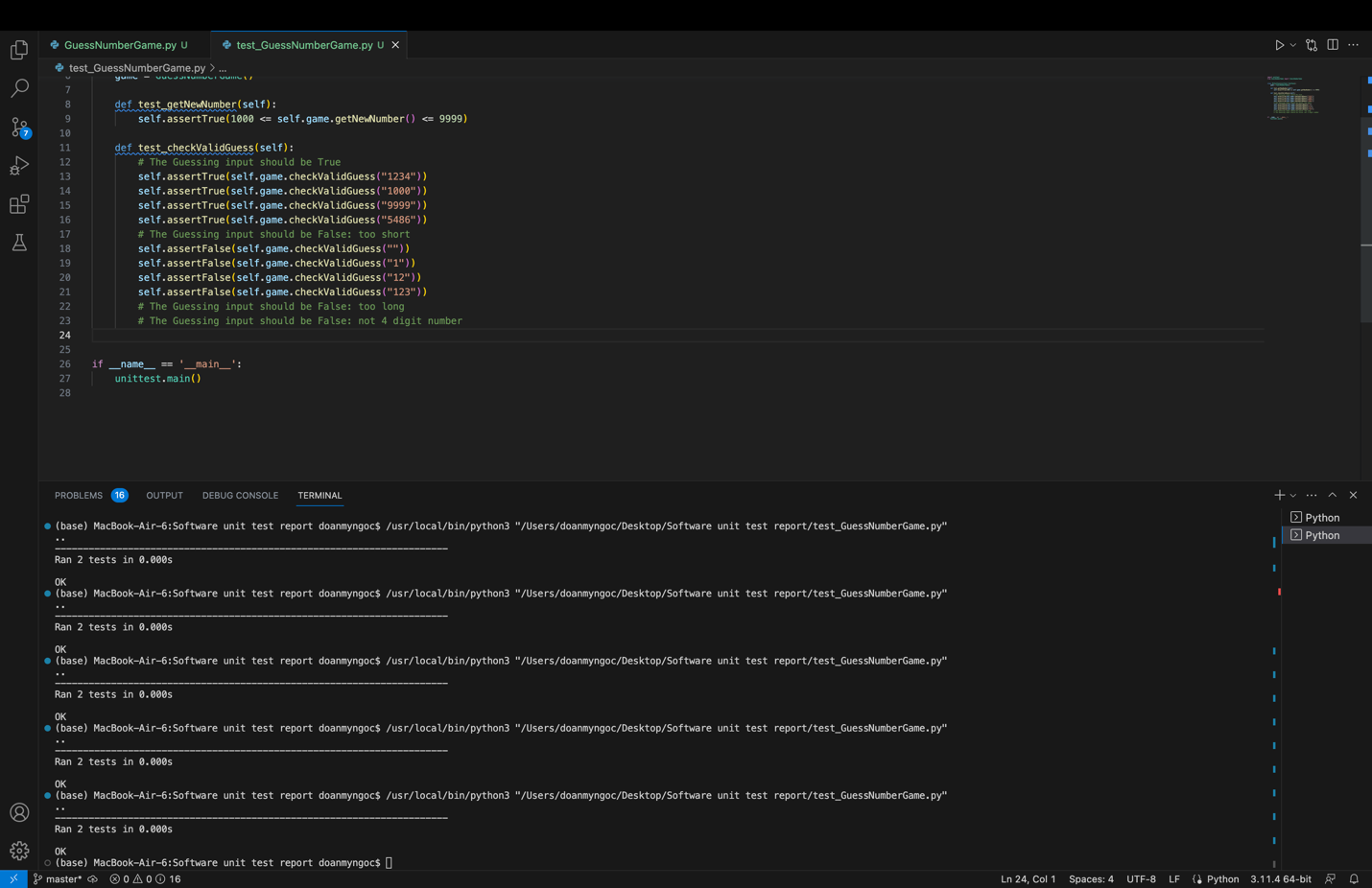
In this game, the input must be a four-digit number.

Whenever the user uses any character, it will show an alert.

The test case for this function has 4 scenarios:

* Input is 4 digit number: it is True
* Input is shorter than 4 digit: it is False
* Input is longer than 4 digit: it is False
* Input has any character: it is False

****



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Step 3:** Giving hint:

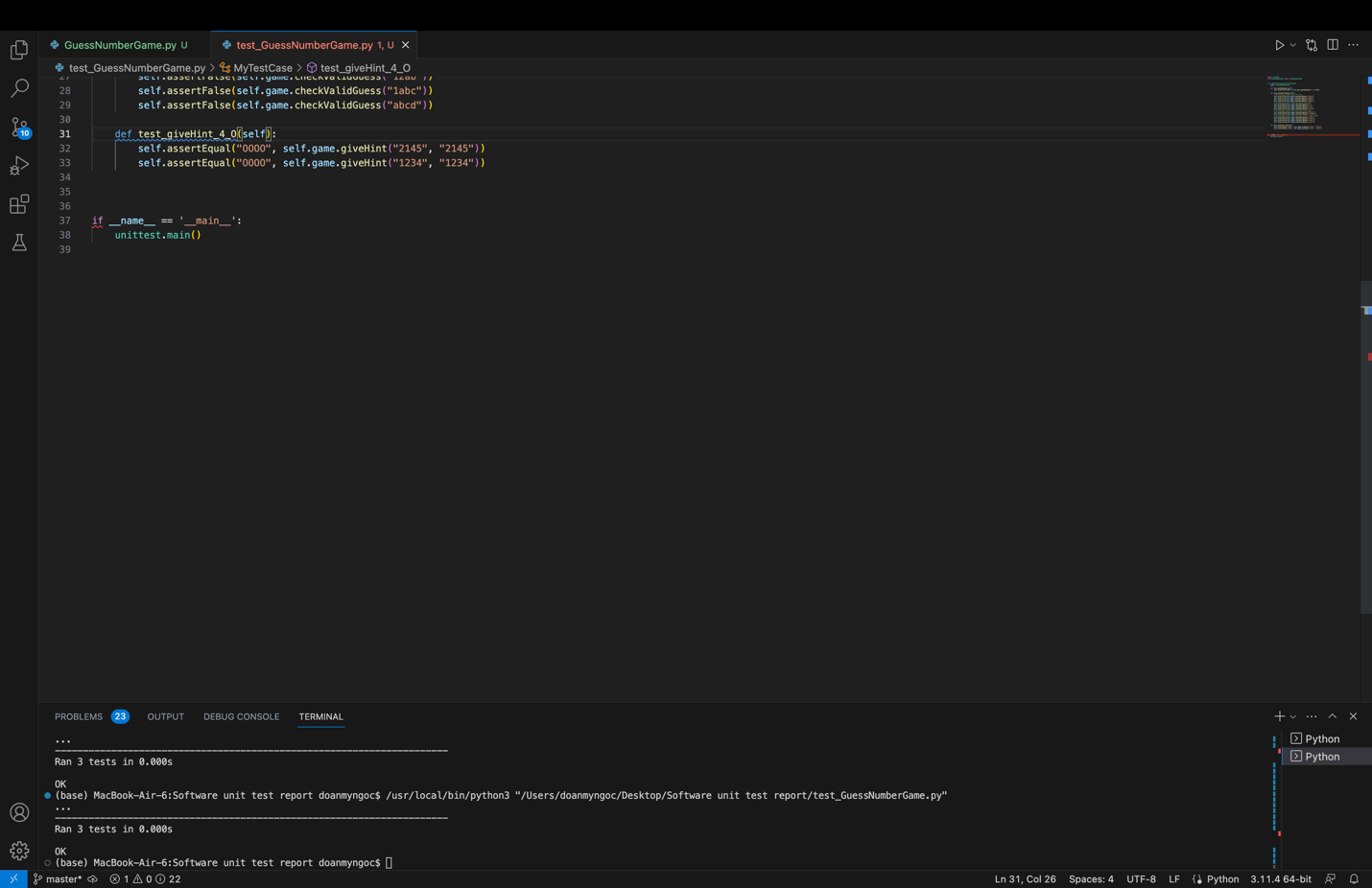
The hint is given when there is a digit is correct, it will be shown as “O” when it in the right position and “X” when it is in the wrong position.

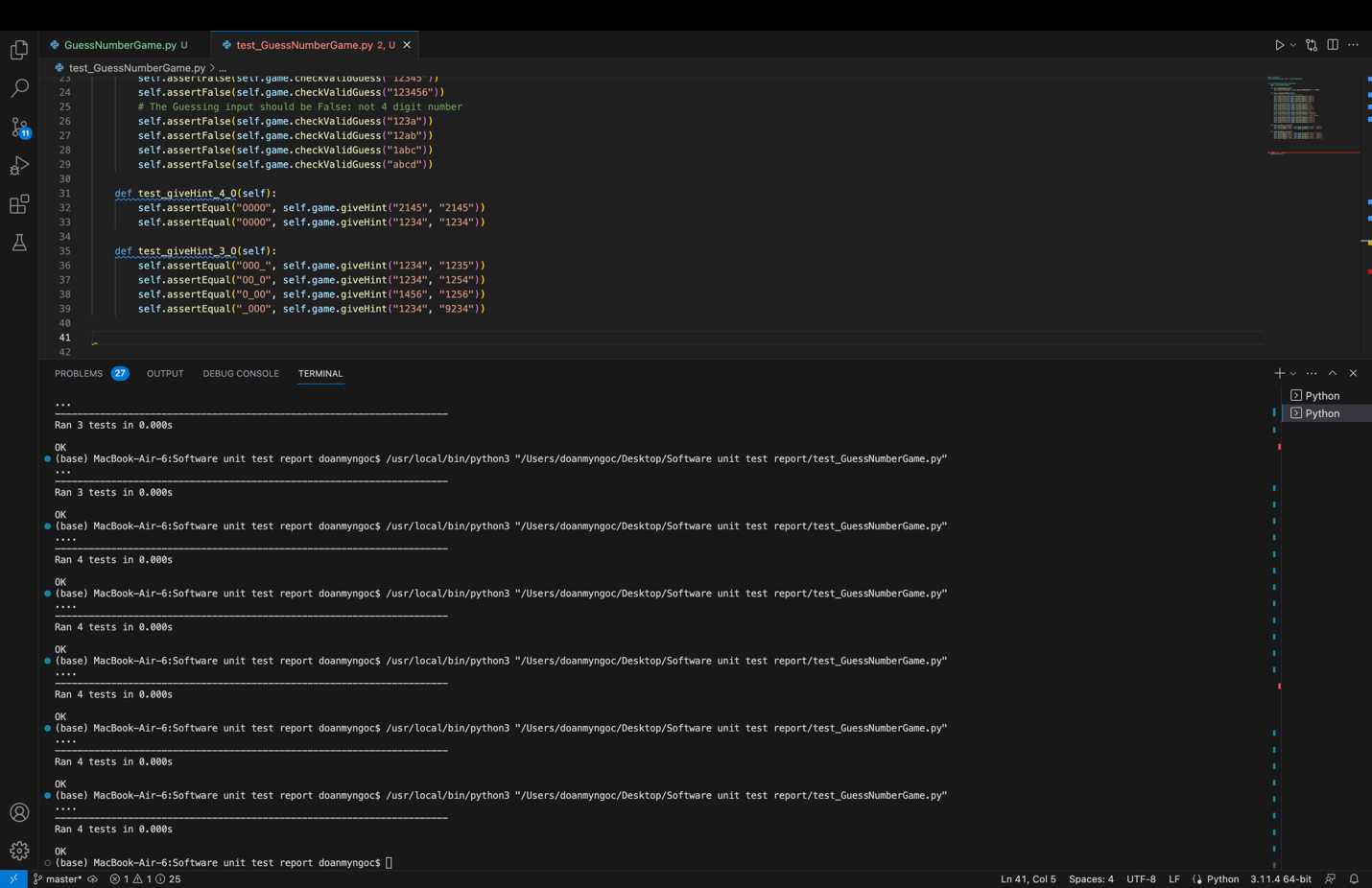
There are so many scenarios to test for this function. Such as:

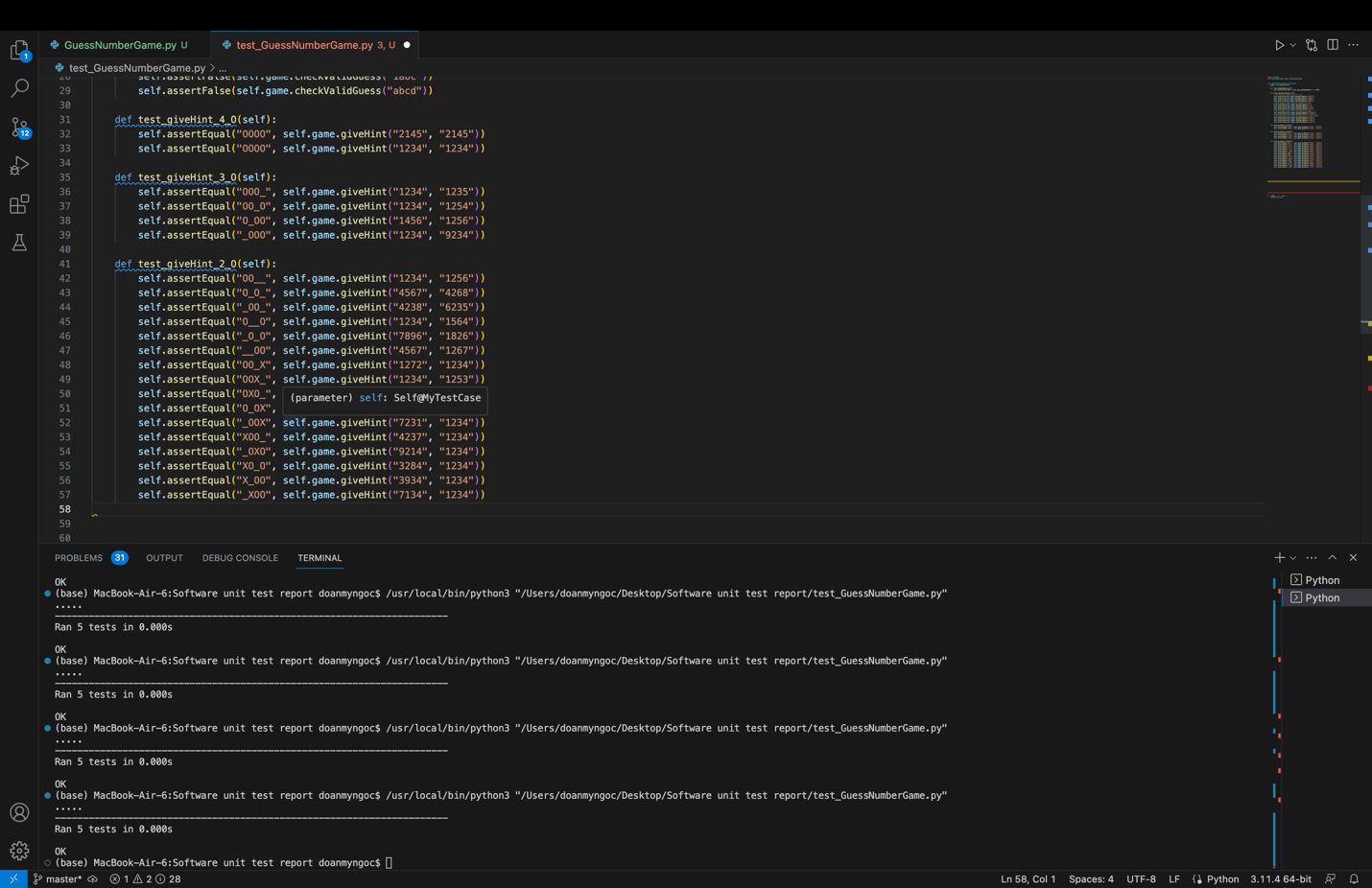
* All four digits are correct, it will be shown as “OOOO”.
* There are 3 digits that are correct, in the right position and 1 is not: it will be shown with 3 “O” and 1 “\_”
* There are 2 digits that are correct, in the right position and 2 are not, so it can be shown in 2 forms: 2 “O” and 2 “\_” or 2 “O”, 1 “X” and 1 “\_” because 2 other digit can be correct but in the wrong position or completely incorrect.

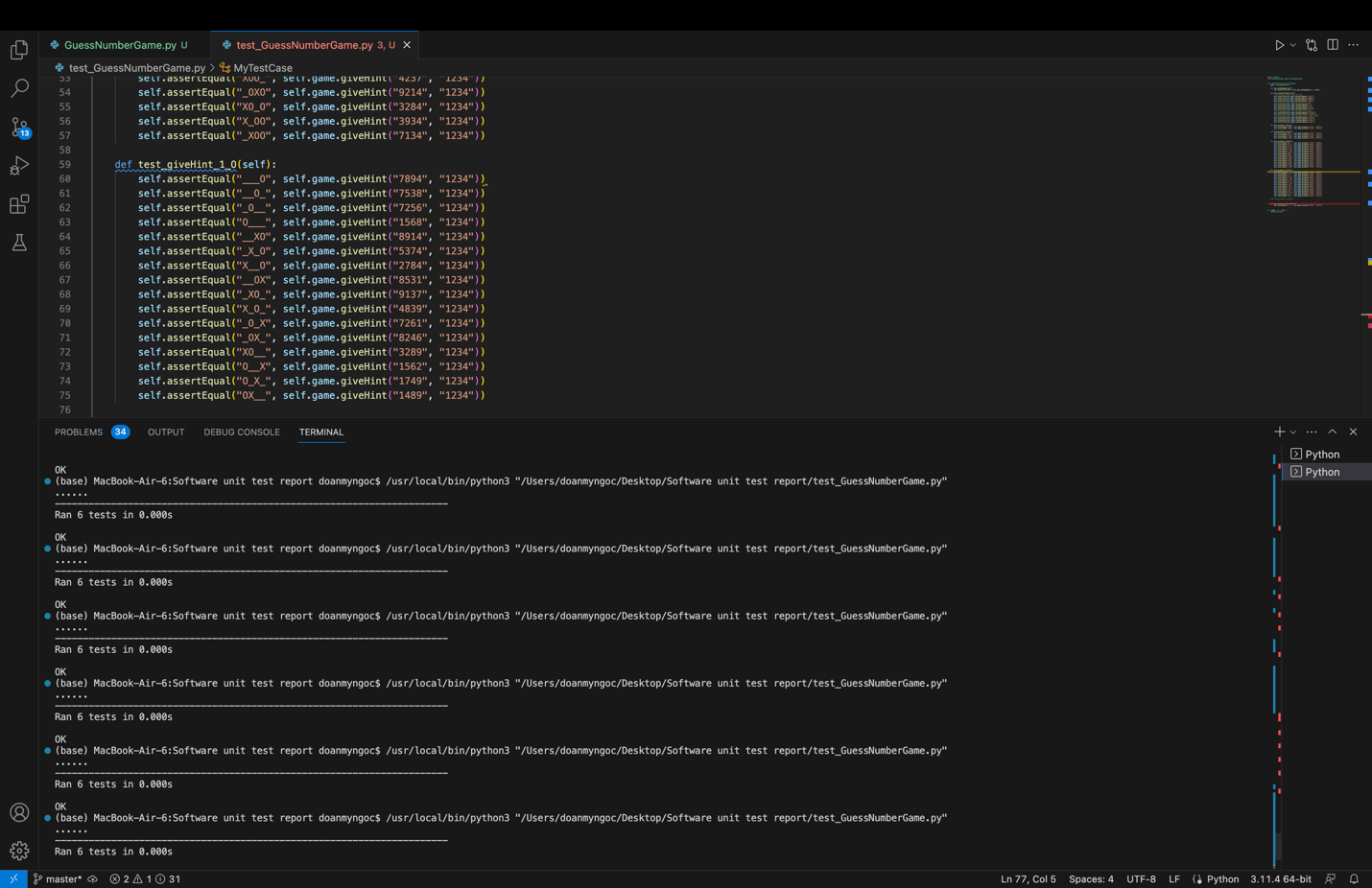
Similar to that, we will have more cases:

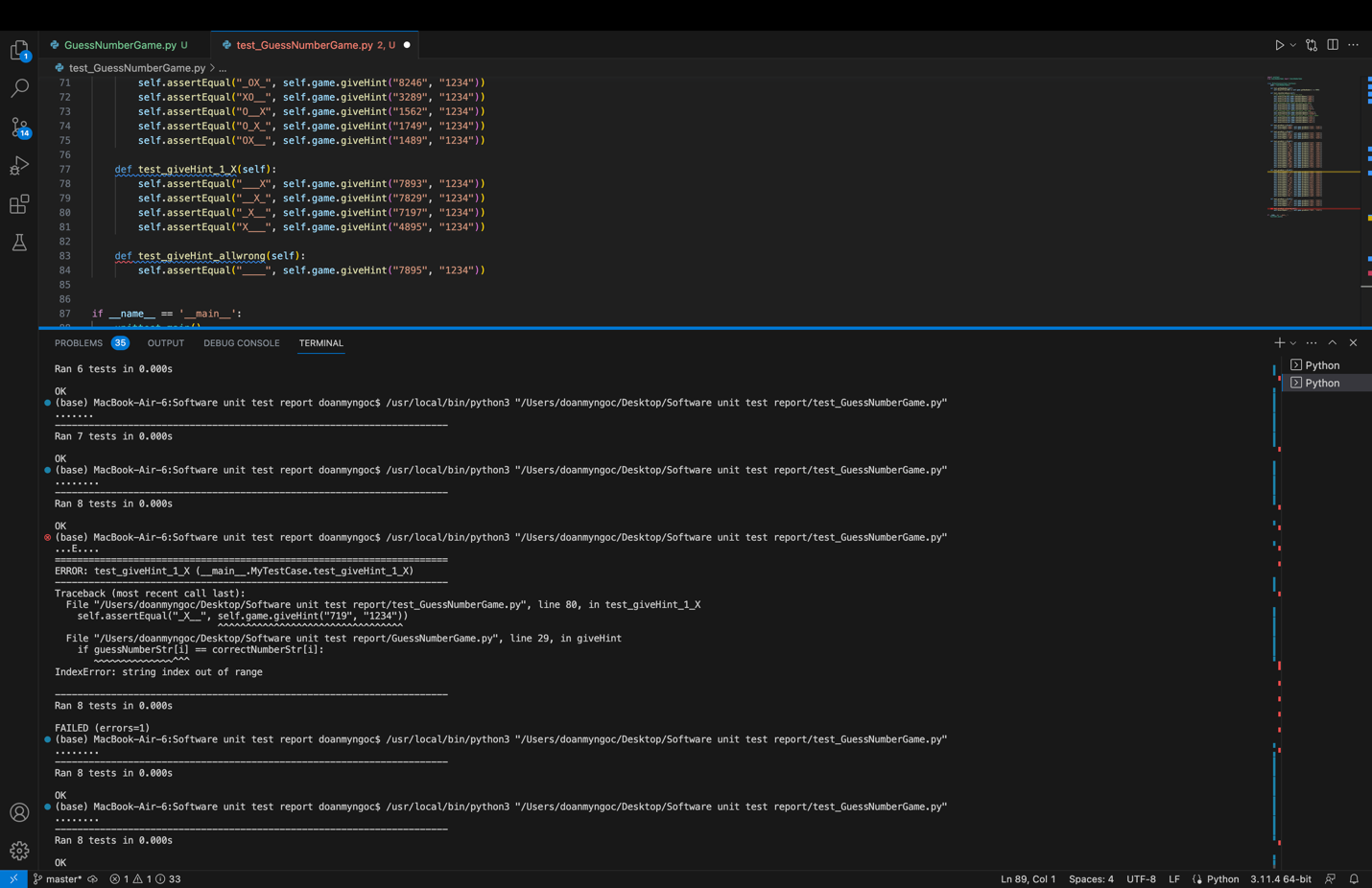
* There are 1 digits that are correct, in the right position.
* There is no digit that is correct with the right position but it can be correct with the wrong position.
* Lastly, none of the digits is correct.









Lastly, when every function in the program is tested and fixed code when we combine all the functions in the program, the program will work effectively as we expected.

# CONCLUSION

The “Guess the Number game” is one example showing the effectiveness of TDD in writing code. In this case, TDD helps to build the code of 3 functions to create a random number, check the validity of user input and give hints by improving the code after testing its test cases. In the function create random numbers and check valid input, we can cover all the test cases, and we can make sure all the possibilities are checked. However, in the function of giving hints, there are too many test cases, and we cannot test them one by one, that is the thing that needs to be improved.