Raport Testare cu Tool AI (ChatGPT)

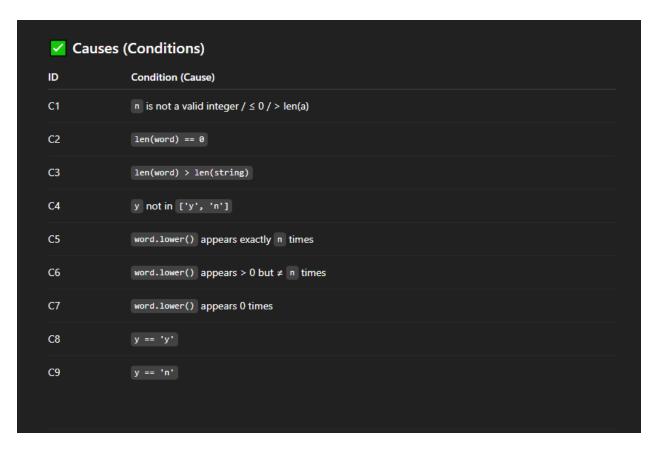
Test 1) Testare functionala (cauza-efect)

Se trece functia de test in prompt si se cere efectuarea testarilor functionale (cauza – efect):

Please make functional tests on it (cause-effect type) and show all of the results

To perform **cause-effect (CE) functional testing** on the word_has_n_occurrences_in_text method, we'll first identify **causes (inputs/conditions)** and **effects (outputs/results)**, then build a **cause-effect graph** leading to a **decision table**, and finally create test cases based on that.

Dupa care determina cauzele:



Si efectele:

© Effects (Results)									
ID	Effect (Result)	Code Added to							
E1	Print "Please enter a valid number of occurrences"	1							
E2	Print "Please enter a valid word"	2							
E3	Print "Please enter a valid text and word"	3							
E4	Print "Please enter a valid continue word"	4							
E5	Print "The word is present in the text with n occurrences"	5							
E6	Print "The word is NOT present in the text with n occurrences"	6							
E7	Print "The word is NOT present in the text"	7							
E8	Print "Enter new info"	8							
E9	Print "Exiting"	9							

Comparativ cu rezultatele mele, ChatGPT a gasit mai putine conditii decat la testarea mea, deoarece a grupat mai multe conditii intre ele

Tabelul de decizie:

■ Decision Table												
тс	C1	C2	C3	C4	C5	C6	C7	C8	C9	Expected r		
1	T	F	F	F				Т	F	18		
2	F	Т	F	F				T	F	28		
3	F	F	T	F				Т	F	38		
4	F	F	F	Т				T*	F	48 (y is set to 'y')		
5	F	F	F	F	T	F	F	Т	F	58		
6	F	F	F	F	F	Т	F	Т	F	68		
7	F	F	F	F	F	F	Т	Т	F	78		
8	F	F	F	F	Т	F	F	F	Т	59		
9	T	Т	T	T	-	-	-	T	F	12348		

Dupa care a generat un script cu care pot sa fac aceste teste:

```
class TestFunction(unittest.TestCase):
    def setUp(self):
        self.func = Function()

    def test_tc1(self):
        self.assertEqual(self.func.word_has_n_occurrences_in_text("hello world", "hello",
    3, 'y'), 18)

    def test_tc2(self):
        self.assertEqual(self.func.word_has_n_occurrences_in_text("hello world", "", 1,
        'y'), 28)

    def test_tc3(self):
        self.assertEqual(self.func.word_has_n_occurrences_in_text("short", "toolongword",
    1, 'y'), 38)

    def test_tc4(self):
        self.assertEqual(self.func.word_has_n_occurrences_in_text("hello world", "hello",
        1, 'x'), 48)

    def test_tc5(self):
```

```
self.assertEqual(self.func.word_has_n_occurrences_in_text("hello world hello",
"hello", 2, 'y'), 58)

def test_tc6(self):
    self.assertEqual(self.func.word_has_n_occurrences_in_text("hello world hello",
"hello", 1, 'y'), 68)

def test_tc7(self):
    self.assertEqual(self.func.word_has_n_occurrences_in_text("hello world", "test",
1, 'y'), 78)

def test_tc8(self):
    self.assertEqual(self.func.word_has_n_occurrences_in_text("hello world hello",
"hello", 2, 'n'), 59)

def test_tc9(self):
    self.assertEqual(self.func.word_has_n_occurrences_in_text("", "", 0, 'maybe'),
12348)

if __name__ == '__main__':
    unittest.main()
```

Ruland acest cod individual:

```
PASSED
                           [ 11%]Please enter a valid number of occurrences
Enter new info
PASSED
                          [ 22%]Please enter a valid word
Enter new info
                          [ 33%]Please enter a valid text and word
PASSED
Enter new info
PASSED
                           [ 44%]Please enter a valid continue word
Enter new info
PASSED
                           [ 55%] The word is present in the text with 2 occurrences
Enter new info
PASSED
                           [ 66%] The word is NOT present in the text with 1 occurrences
Enter new info
PASSED
                           [ 77%] The word is NOT present in the text
Enter new info
                           [ 88%]The word is present in the text with 2 occurrences
PASSED
Exiting
                        [100%]Please enter a valid number of occurrences
Please enter a valid word
Please enter a valid continue word
Enter new info
function_for_test.py:76 (TestFunction.test_tc9)
Expected :1248
Actual :12348
self = <function_for_test.TestFunction testMethod=test_tc9>
function_for_test.py:78: AssertionError
```

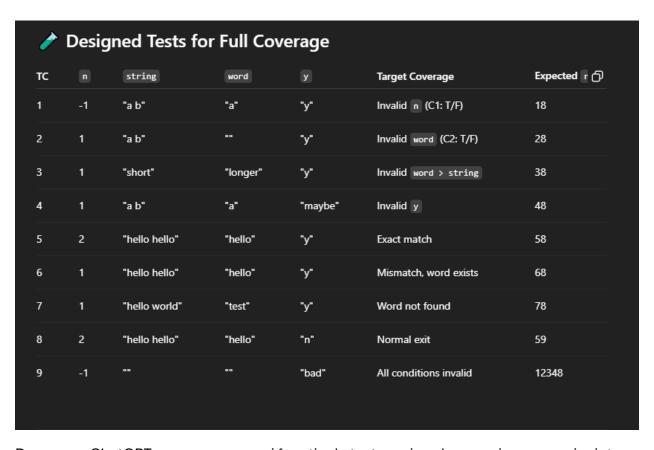
8 teste au fost ok dar ultimul a fost gresit

Se cere crearea testarea structurala a functiei (la nivel de instructiune/decizie/conditie/circuit):

Please make functional tests on the original function (
statement/branch/condition/circuit coverage) and show the results

To test the original function using statement, branch, condition, and multiple condition (circuit) coverage criteria, we'll do the following:

Genereaza textele pentru acoperire totala:



Dupa care ChatGPT genereaza aceasi functie de testare si va da aceasi eroare ca inainte