COS80022 – Software Quality and Testing

Test Report for Lab 9

Student Name: Arun Ragavendhar Arunachalam Palaniyappan
Student ID: 104837257
Student email: 104837257@student.swin.edu.au

Submission date: 05/05/2024

1. For ‘getMin1’

1.1 Test cases

Table 1. Test cases for 100% statement coverage on ‘getMin1’

#	Input Data	Expected Result (Oracle)	Actual Result (Log)
TC1	7,5,3	3	3

Table 2. Test cases for 100% branch coverage on ‘getMin1’

#	Input Data	Expected Result (Oracle)	Actual Result (Log)
TC1	7,5,3	3	3

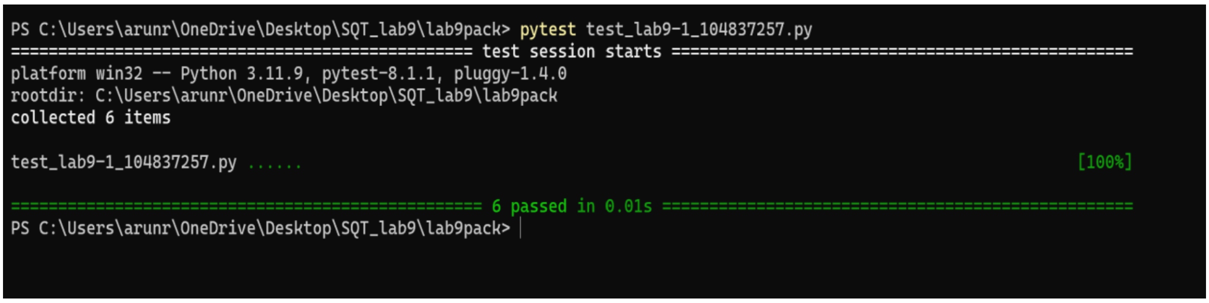
Table 3. Test cases for 100% path coverage on ‘getMin1’

#	Input Data	Expected Result (Oracle)	Actual Result (Log)
TC1	7,5,3	3	3
TC2	3,5,7	3	3
TC3	7,3,5	3	3
TC4	5,7,3	3	3

1.2 Screenshot of test results

All test cases passed and 100 percent coverage of statement testing, branch testing and path testing has been achieved for getMin1() function.

Test Case for statement coverage - 1
Test Case for branch coverage - 1
Test cases for path coverage - 4



```
PS C:\Users\arunr\OneDrive\Desktop\SQT_lab9\lab9pack> pytest test_lab9-1_104837257.py
================================================= test session starts =================================================
platform win32 -- Python 3.11.9, pytest-8.1.1, pluggy-1.4.0
rootdir: C:\Users\arunr\OneDrive\Desktop\SQT_lab9\lab9pack
collected 6 items

test_lab9-1_104837257.py ......                                                                                  [100%]

================================================== 6 passed in 0.01s ==================================================
PS C:\Users\arunr\OneDrive\Desktop\SQT_lab9\lab9pack>
```

2.For ‘getMin2’

2.1 Test cases

Table 4. Test cases for 100% statement coverage on ‘getMin2’

#	Input Data	Expected Result (Oracle)	Actual Result (log)
TC1	2,5,9	2	2
TC2	5,9,2	2	2
TC3	5,2,9	2	2
TC4	9,5,2	2	2

Table 5. Test cases for 100% branch coverage on ‘getMin2’

#	Input Data	Expected Result (Oracle)	Actual Result (log)
TC1	2,5,9	2	2
TC2	5,9,2	2	2
TC3	5,2,9	2	2
TC4	9,5,2	2	2

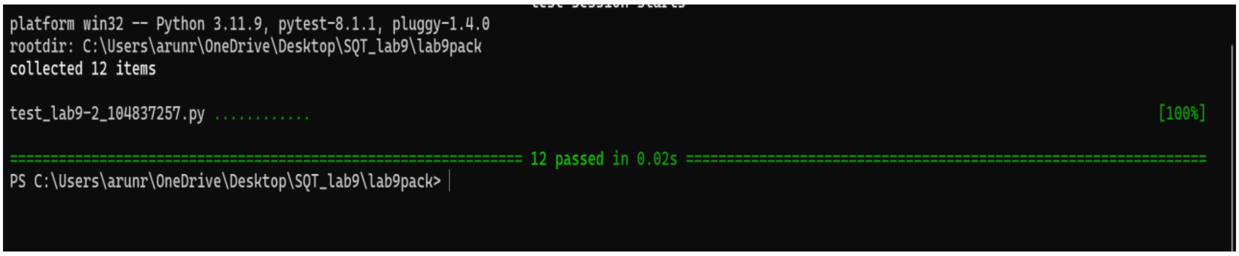
Table 6. Test cases for 100% path coverage on ‘getMin2’

#	Input Data	Expected Result (Oracle)	Actual Result (log)
TC1	2,5,9	2	2
TC2	5,9,2	2	2
TC3	5,2,9	2	2
TC4	9,5,2	2	2

2.2 Screenshot of test results

All test cases passed and 100 percent coverage of statement testing, branch testing and path testing has been achieved for getMin2() function.

Test Case for statement coverage - 4
Test Case for branch coverage - 4
Test cases for path coverage - 4



```
platform win32 -- Python 3.11.9, pytest-8.1.1, pluggy-1.4.0
rootdir: C:\Users\arunr\OneDrive\Desktop\SQT_lab9\lab9pack
collected 12 items

test_lab9-2_104837257.py ............                                                   [100%]

========================================================== 12 passed in 0.02s ==========================================================
PS C:\Users\arunr\OneDrive\Desktop\SQT_lab9\lab9pack>
```

3.Analysis of test cases

3.1 Is there any difference among the three sets of test cases (Table 1-3) for 'getMin1'? Why?

Yes, there is a difference among the test cases.

For getMin1() function – it has 2 linear 'IF' statements. To achieve 100 percent coverage in statement as well as branch testing here, we just need one test case where,

a>b>c - (7 ,5 ,3) - test case for 100 percent statement coverage

a>b>c - (7 ,5 ,3) - test case for 100 percent branch coverage

But, for path coverage , we have to check all the paths

Path 1: 1st IF is 'true' and 2nd IF is 'true' - a>b>c (7,5,3)

Path 2: 1st IF is 'false' and 2nd IF is 'false' - a<b<c (3,5,7)

Path 3: 1st IF is 'true' and 2nd IF is 'false' - a>b and b<c (7,3,5)

Path 4: 1st IF is 'false' and 2nd IF is 'true' - a<b and a>c (5,7,3)

So, for 100 percent path coverage we have 4 test cases.

3.2 Is there any difference among the three sets of test cases (Table 4-6) for 'getMin2'? Why?

No. There is no difference among the test cases.

The function getMin2() – has nested 'IF' and 'ELSE' statements. To achieve 100 percent coverage in statement, branch and path testing here, we need 4 test cases for each as all three represent the same scenarios here and are listed in the below cases.

Only when every path is traversed, all branches and all statements can get executed.

Case 1: a<b and a<c

Case 2: a<b and a>c

Case 3: a>b and b<c

Case4: a>b and b>c