Lab Work 9

In Week 9, you will practise white-box testing using PyTest. You are required to submit a zip file containing (1) a test report and (2) the test scripts written in PyTest.

> *Suppose that a python program 'getMin.py' contains two functions, namely 'getMin1' and 'getMin2'. Both functions accept three integers as the input and output the minimal value of them. The control flow of each function is shown by the activity diagram in Figure 1 and Figure 2, respectively.*
>
> *You have been asked to test both functions.*

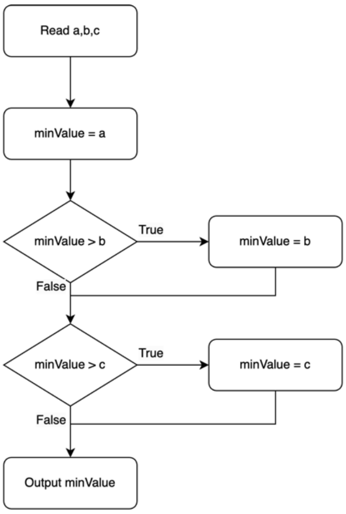

Figure 1 Control flow for 'getMin1'

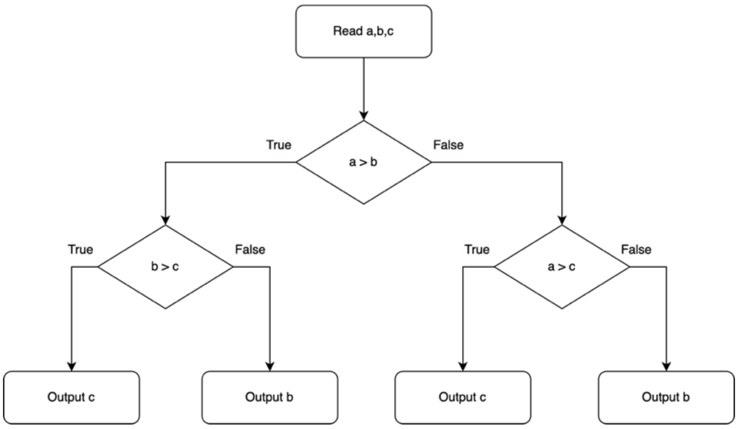

Figure 2. Control flow for 'getMin2'

Please follow the instructions below to complete the lab work:

1. Download the "SQT_lab9_pack.zip" file and unzip it to a local directory.

2. Open the command line interface,
    a. For 'Command Prompt' in Windows: https://www.lifewire.com/how-to-open-command-prompt-2618089
    b. For 'Terminal' In Mac: https://www.businessinsider.com/how-to-open-terminal-on-mac?r=AU&IR=T

3. Go to the directory where you have unzipped in Step 1, using the "cd" command (http://www.openforis.org/tools/sepal/tutorials/using-the-command-line.html).

4. Make sure that both Python and PyTest have been installed in your machine, by typing in the commands "python -h" and "pytest -h", respectively.

5. You are required to test the program, namely "getMin.py" under the folder "lab9", especially the functions "getMin1" and "getMin2" in the program. The input for each function is composed of three integers, *a*, *b*, and *c*, and the output is the minimum value of these three numbers. You need to design

test cases for both functions according to some code coverage criteria, with detailed given in Step 6 and 7, respectively.

6. For the function ‘getMin1’ (refer to the control flow in Figure 1):
    a. Create a set of test cases that can achieve 100% statement coverage.
    b. Create a set of test cases that can achieve 100% branch coverage.
    c. Create a set of test cases that can achieve 100% path coverage[1].

   All created test cases should be written in the test script “test_lab9-1_xxxxxx.py” (where “xxxxxx” should be replaced by your student ID). You can follow the provided template script which can be found after unzipping the “SQT_lab9_pack.zip” file in Step 1.

7. For the function ‘getMin2’ (refer to the control flow in Figure 2):
    a. Create a set of test cases that can achieve 100% statement coverage.
    b. Create a set of test cases that can achieve 100% branch coverage.
    c. Create a set of test cases that can achieve 100% path coverage.

   All created test cases should be written in the test script “test_lab9-2_xxxxxx.py” (where “xxxxxx” should be replaced by your student ID). You can follow the provided template script which can be found after unzipping the “SQT_lab9_pack.zip” file in Step 1.

8. Execute all test cases by running the commands “pytest test_lab9-1_xxxxxx.py” and “pytest test_lab9-2_xxxxxx.py” in the command line interface.

9. Analyse the difference among each set of test cases created in Step 6 and 7, and answer the following questions:
    a. Is there any difference among the three sets of test cases for ‘getMin1’ (Step 6)? Why?
    b. Is there any difference among the three sets of test cases for ‘getMin2’ (Step 7)? Why?

10. Summarise your work in a test report. You can follow the provided template to write the report.

11. Submit a zip file containing the test report (created in Step 10) as well as your test scripts “test_lab9-1_xxxxxx.py” and “test_lab9-2_xxxxxx.py” (created in Step 6 and 7).

[1] Note that although 100% path coverage is not possible for many programs (especially those with loops), it is possible for the two functions under test in this lab work.