Assignment #5

Please submit a single zip or PDF file.

Problem 1		Problem 2			Total
1.1	**1.2**	**2.1**	**2.2**	**2.3 (a, b, c)**	
20 pts	20 pts	20 pts	20 pts	10+5+5 pts	100 pts

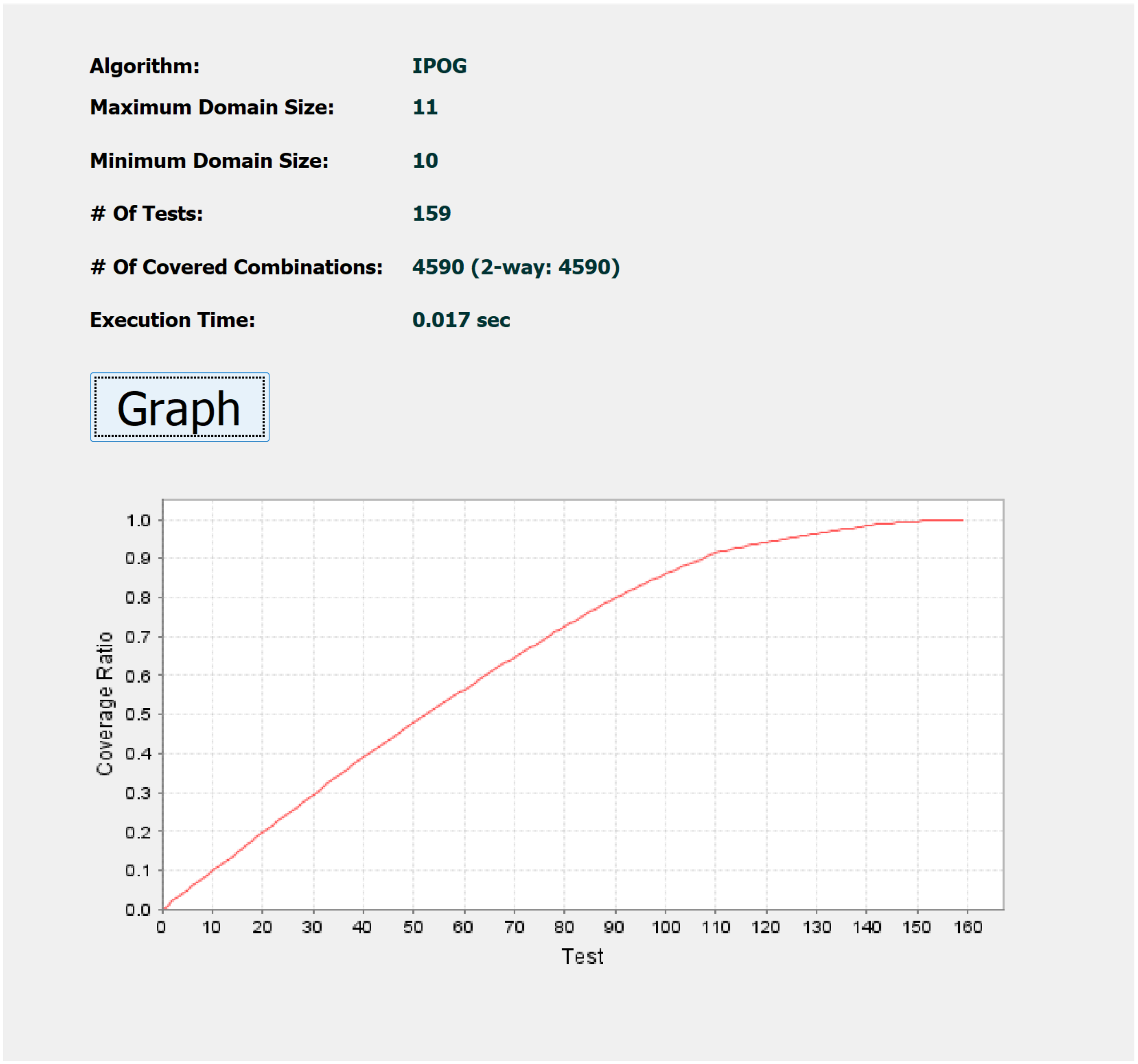
1. Combinatorial Testing

Download and unzip the ACTS tool (https://umkc.box.com/s/4id921gjqs6aza866swp14hncnht86vm) and use its default IPOG method to generate combinatorial tests for the following problems. To run acts_3.2.jar, ensure that the Java runtime environment is installed and properly configured. On macOS, you may also need to adjust your security settings to allow acts_3.2.jar to be executed.

1.1 ISBN10

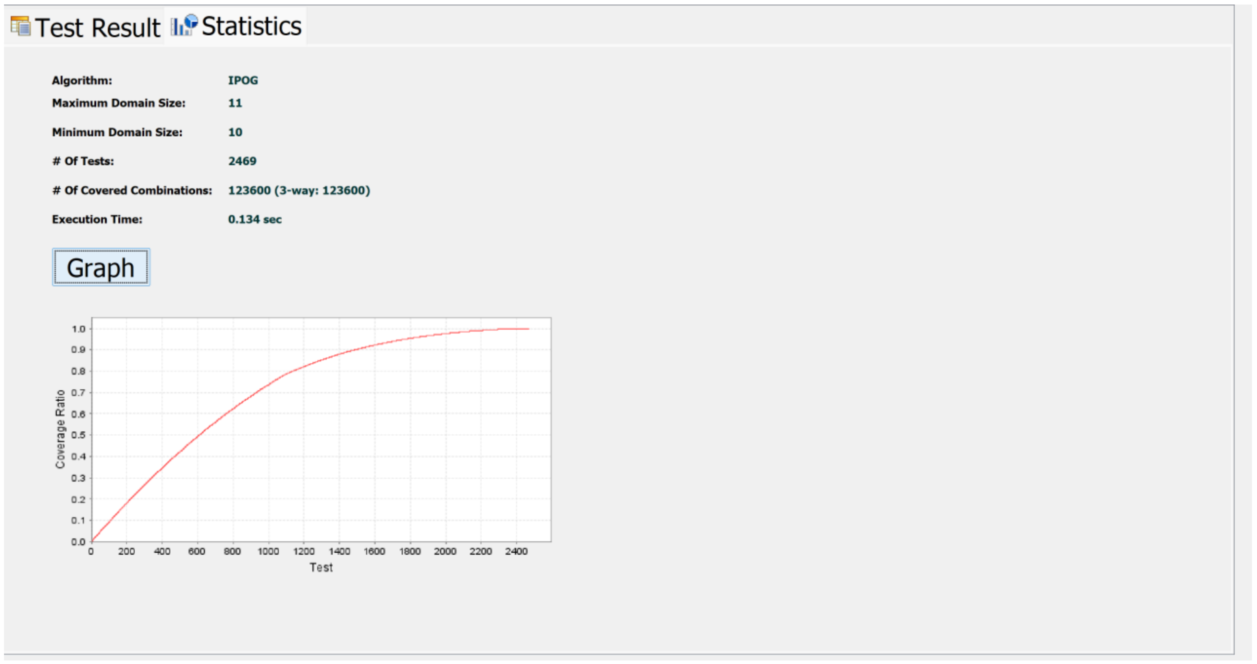
In a valid ISBN-10 number, each of the first 9 characters is a digit (i.e., 0-9) and the 10th character is a digit or X (i.e., 0-10). Assuming the weighted sum of the 10 digits is not considered, use ACTS to determine the numbers of 2-way and 3-way combinations. Submit the following: (a) the numbers of 2-way and 3-way combinations, (b) the screenshots of the results reported by ACTS.

A) 2-way Testing: 4590
   3-way Testing:123600



Algorithm:	**IPOG**
Maximum Domain Size:	**11**
Minimum Domain Size:	**10**
# Of Tests:	**159**
# Of Covered Combinations:	**4590 (2-way: 4590)**
Execution Time:	**0.017 sec**

B)



Algorithm: IPOG
Maximum Domain Size: 11
Minimum Domain Size: 10
Of Tests: 2469
Of Covered Combinations: 123600 (3-way: 123600)
Execution Time: 0.134 sec

1.2 Pizza Ordering

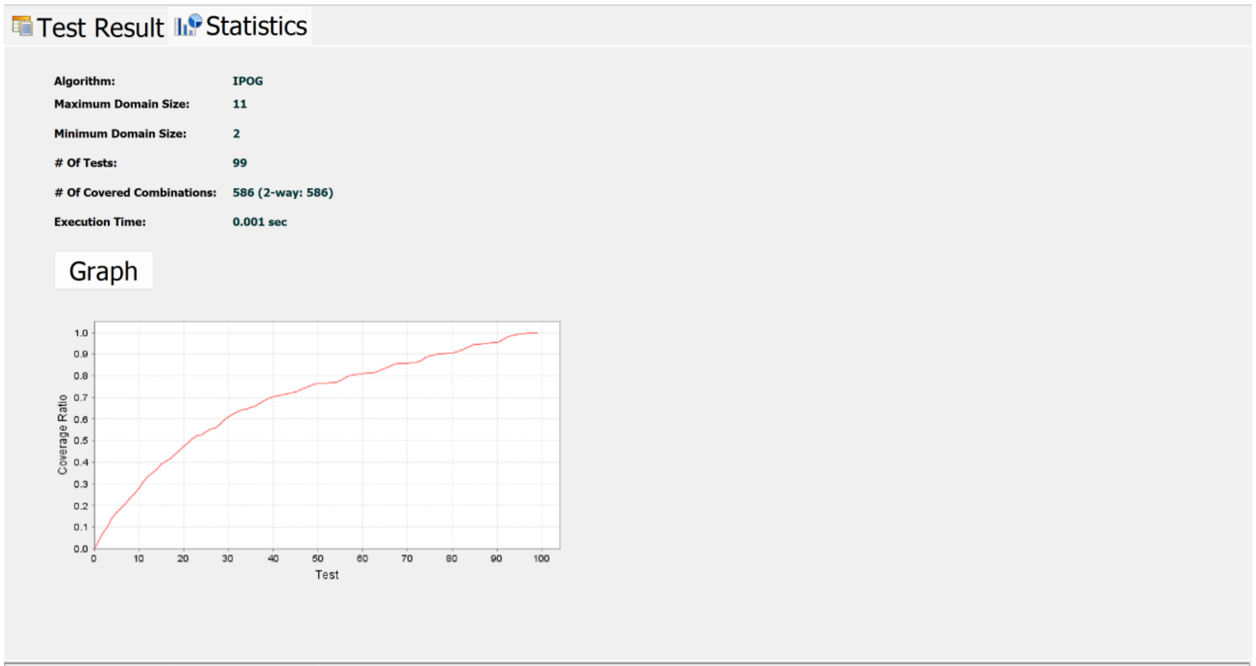
The following table shows some of the options for customizing a pizza at papajohns.com. For simplicity, here we pick only one value for each variable.

Variable	**Values**
Cut	Normal Cut Square Cut Clean Cut No Cut
Sauce	BBQ Ranch Original Buffalo Alfredo Sauce
How much sauce?	Normal Sauce Light Sauce Extra Sauce No Sauce
Bake	Normal Bake Well Done
How much cheese?	Normal Cheese Light Cheese No Cheese
Meats	Salami Bacon Anchovies Sausage

	Grilled Chicken Meatball Philly Steak Canadian Bacon Pepperoni Spicy Italian Sausage Beef
Veggies	Fresh Spinach Onions Pineapple Roma Tomatoes Mushrooms Jalapeño Peppers Banana Peppers Green Peppers Black Olives

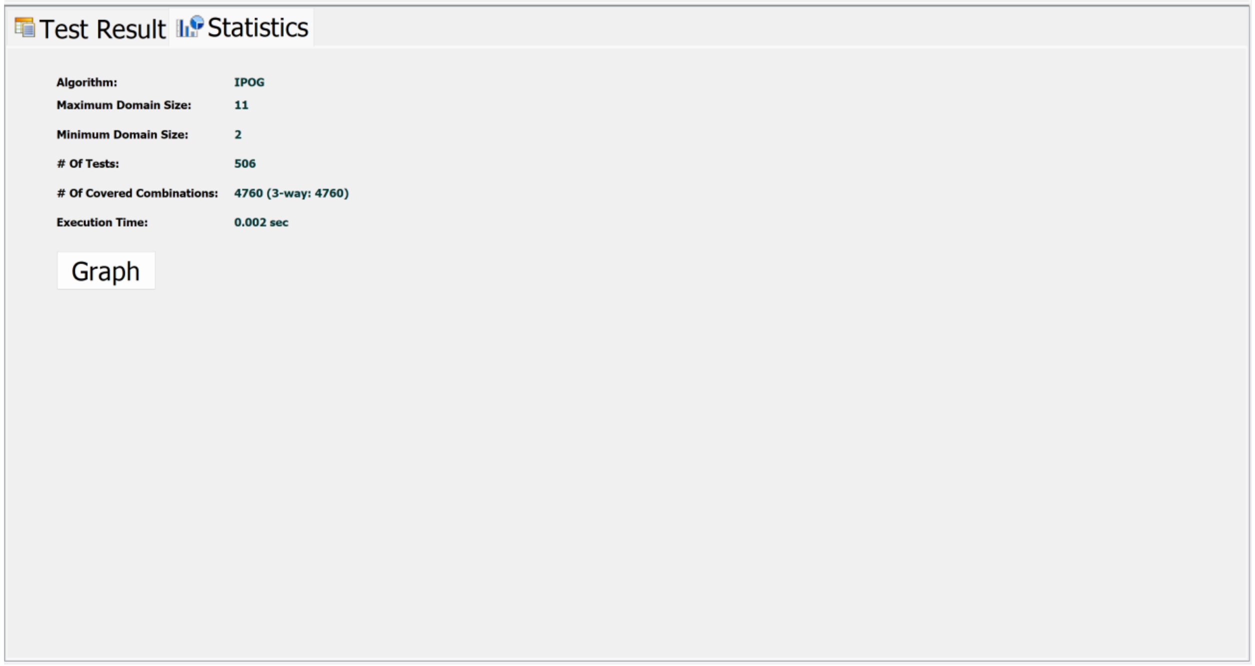
Use ACTS to determine the number of 2-way and 3-way combinations. Submit the following: (a) the numbers of 2-way and 3-way combinations, (b) the screenshots of the results reported by ACTS.

A) 2-way combinations:586
   3-way combinations:4760



Algorithm: IPOG
Maximum Domain Size: 11
Minimum Domain Size: 2
Of Tests: 99
Of Covered Combinations: 586 (2-way: 586)
Execution Time: 0.001 sec

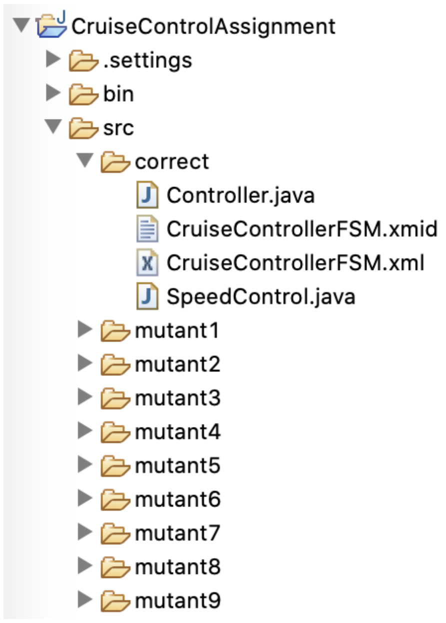
B)



Algorithm: IPOG
Maximum Domain Size: 11
Minimum Domain Size: 2
Of Tests: 506
Of Covered Combinations: 4760 (3-way: 4760)
Execution Time: 0.002 sec

2. Model-Based Testing

Download and unzip the MISTA tool (https://umkc.box.com/s/2w3xzklrga1fds60onhfipq33si7a8jz). To run MISTA.jar, ensure that the Java runtime environment is installed and properly configured. On macOS, you may also need to adjust your security settings to allow MISTA.jar to be executed. The problems are based on a simplified cruise control simulation program in Java (i.e., CruiseContorlAssignment project). The Java project is structured as follows.

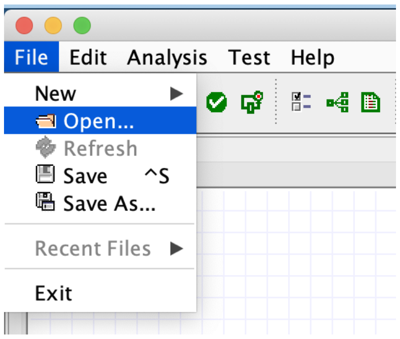


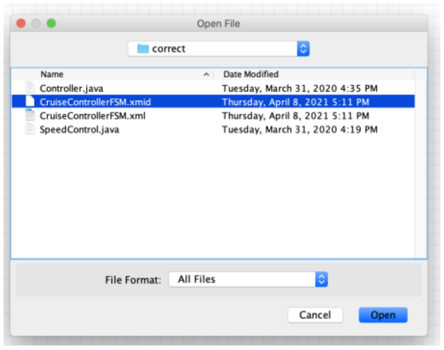
```
▼ CruiseControlAssignment
  ▶ .settings
  ▶ bin
  ▼ src
    ▼ correct
        Controller.java
        CruiseControllerFSM.xmid
        CruiseControllerFSM.xml
        SpeedControl.java
    ▶ mutant1
    ▶ mutant2
    ▶ mutant3
    ▶ mutant4
    ▶ mutant5
    ▶ mutant6
    ▶ mutant7
    ▶ mutant8
    ▶ mutant9
```

The source code consists of ten folders: correct, mutant1, mutant2, …., and mutant9. The correct folder includes correct classes Controller.java and SpeedControl.java, whereas mutant (X=1, …9) contains a mutant of Controller.java. The correct folder also includes two files CruiseControllerFSM.xmid and CruiseControllerFSM.xml for model-based test generation. You will use MISTA to open CruiseControllerFSM.xmid.

2.1 Round-trip tree

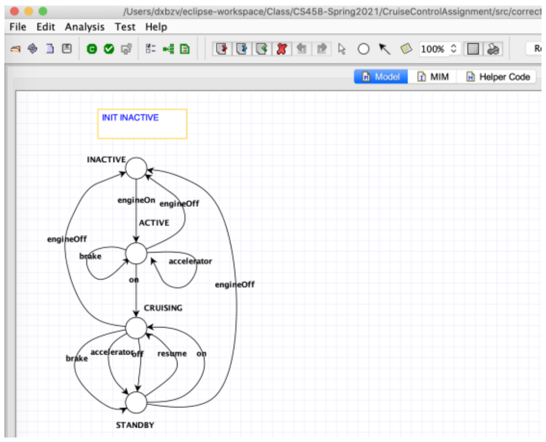
Step 1. Run MISTA and open CruiseControllerFSM.xmid.





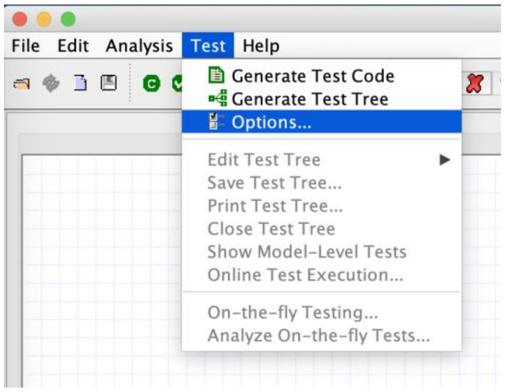
Name	Date Modified
Controller.java	Tuesday, March 31, 2020 4:35 PM
CruiseControllerFSM.xmid	Thursday, April 8, 2021 5:11 PM
CruiseControllerFSM.xml	Thursday, April 8, 2021 5:11 PM
SpeedControl.java	Tuesday, March 31, 2020 4:19 PM

You should get the following test model:

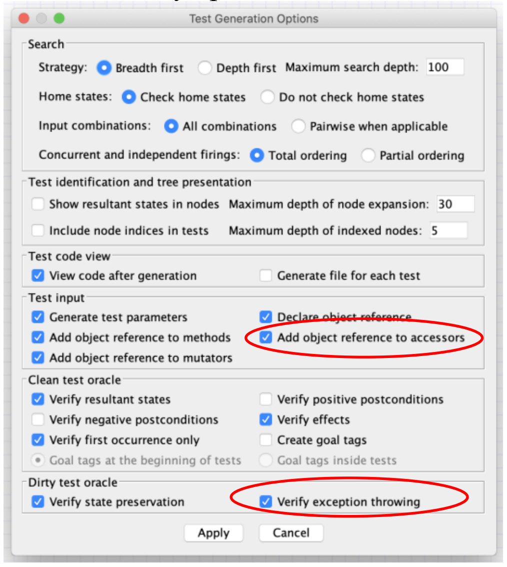


Step 2. Set up test generation options

Select menu "Test" and then menu item "Options"



There are many options as shown below.



Test Generation Options

Search
- Strategy: [✓ Breadth first] Depth first Maximum search depth: 100
- Home states: [✓ Check home states] Do not check home states
- Input combinations: [✓ All combinations] Pairwise when applicable
- Concurrent and independent firings: [✓ Total ordering] Partial ordering

Test identification and tree presentation
- [ ] Show resultant states in nodes Maximum depth of node expansion: 30
- [ ] Include node indices in tests Maximum depth of indexed nodes: 5

Test code view
- [✓] View code after generation
- [ ] Generate file for each test

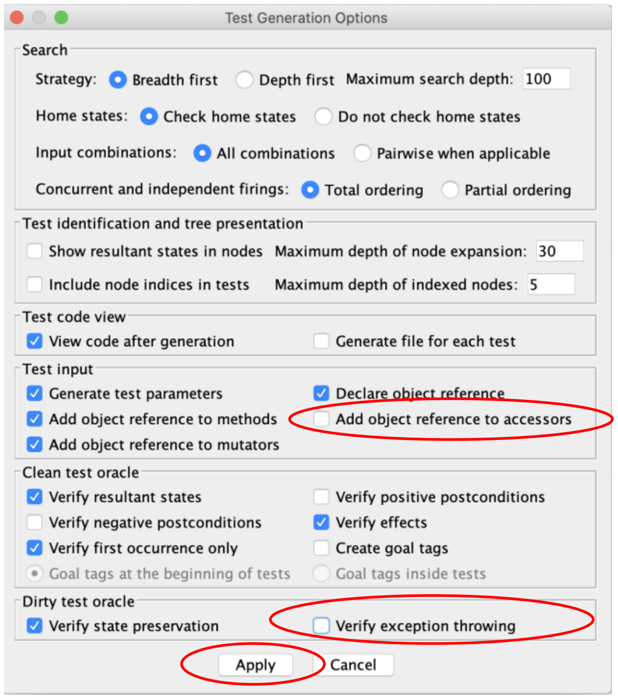
Test input
- [✓] Generate test parameters
- [✓] Declare object reference
- [✓] Add object reference to methods
- [✓] Add object reference to accessors
- [✓] Add object reference to mutators

Clean test oracle
- [✓] Verify resultant states
- [ ] Verify positive postconditions
- [ ] Verify negative postconditions
- [✓] Verify effects
- [✓] Verify first occurrence only
- [ ] Create goal tags
- Goal tags at the beginning of tests
- Goal tags inside tests

Dirty test oracle
- [✓] Verify state preservation
- [✓] Verify exception throwing

Apply Cancel

You only need to uncheck two checkboxes: “Add object reference to accessors” and “Verify excepting handling”


Test Generation Options

Search

Strategy: [✓ Breadth first] Depth first Maximum search depth: 100

Home states: [✓ Check home states] Do not check home states

Input combinations: [✓ All combinations] Pairwise when applicable

Concurrent and independent firings: [✓ Total ordering] Partial ordering

Test identification and tree presentation

Show resultant states in nodes Maximum depth of node expansion: 30

Include node indices in tests Maximum depth of indexed nodes: 5

Test code view

[✓ View code after generation] Generate file for each test

Test input

[✓ Generate test parameters] [✓ Declare object reference]

[✓ Add object reference to methods] Add object reference to accessors

[✓ Add object reference to mutators]

Clean test oracle

[✓ Verify resultant states] Verify positive postconditions

Verify negative postconditions [✓ Verify effects]

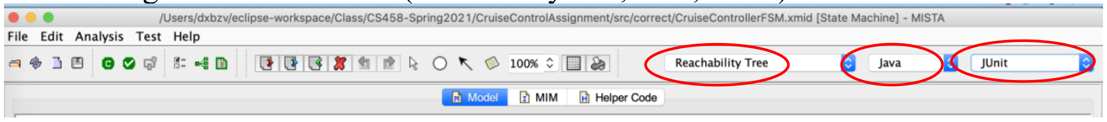
[✓ Verify first occurrence only] Create goal tags

[✓ Goal tags at the beginning of tests] Goal tags inside tests

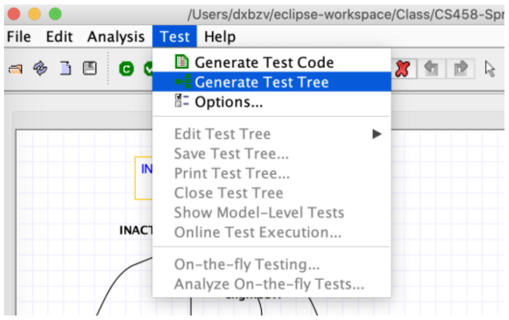
Dirty test oracle

[✓ Verify state preservation] Verify exception throwing

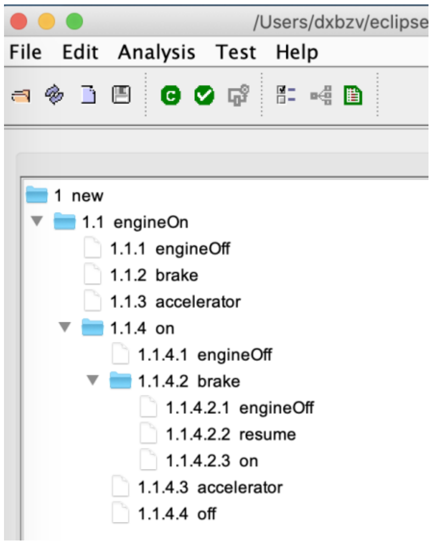
Step 3. Select test generation methods (reachability tree, Java, JUnit)


/Users/dxbzv/eclipse-workspace/Class/CS458-Spring2021/CruiseControlAssignment/src/correct/CruiseControllerFSM.xmid [State Machine] - MISTA

Step 4. Generate the test tree



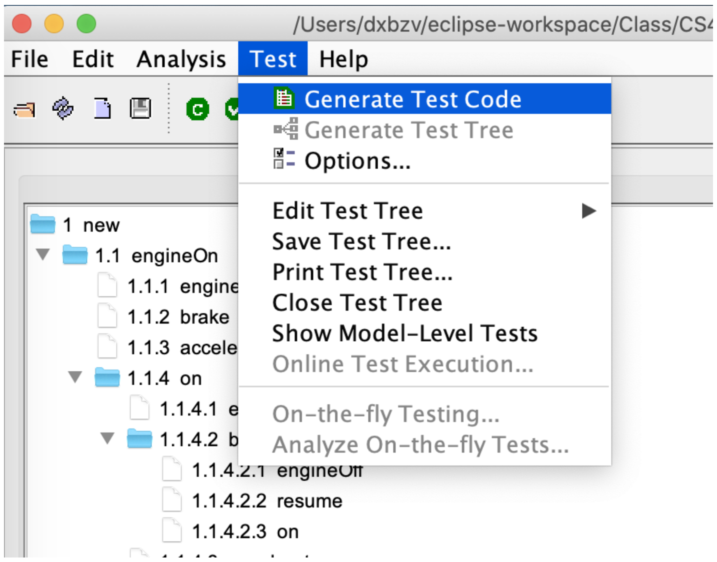
The tree should look like the following:


/Users/dxbzv/eclipse

File Edit Analysis Test Help

1 new
- 1.1 engineOn
  - 1.1.1 engineOff
  - 1.1.2 brake
  - 1.1.3 accelerator
  - 1.1.4 on
    - 1.1.4.1 engineOff
    - 1.1.4.2 brake
      - 1.1.4.2.1 engineOff
      - 1.1.4.2.2 resume
      - 1.1.4.2.3 on
    - 1.1.4.3 accelerator
    - 1.1.4.4 off

Step 5. Generate test code


/Users/dxbzv/eclipse-workspace/Class/CS4

File Edit Analysis Test Help

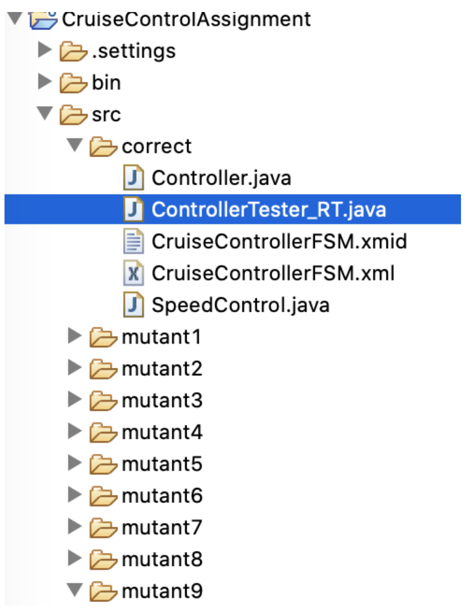
Generate Test Code
Generate Test Tree
Options...

Edit Test Tree
Save Test Tree...
Print Test Tree...
Close Test Tree
Show Model-Level Tests
Online Test Execution...

On-the-fly Testing...
Analyze On-the-fly Tests...

1 new
- 1.1 engineOn
  - 1.1.1 engine
  - 1.1.2 brake
  - 1.1.3 accele
  - 1.1.4 on
    - 1.1.4.1 e
    - 1.1.4.2 b
      - 1.1.4.2.1 engineOff
      - 1.1.4.2.2 resume
      - 1.1.4.2.3 on

The test code should be generated and saved under the “correct” folder. If you refresh the CruiseControlAssignment project, it should look like the following, where ControllerTester_RT.java is the generated test code.



```
▼ CruiseControlAssignment
  ▶ .settings
  ▶ bin
  ▼ src
    ▼ correct
        Controller.java
        ControllerTester_RT.java
        CruiseControllerFSM.xmid
        CruiseControllerFSM.xml
        SpeedControl.java
    ▶ mutant1
    ▶ mutant2
    ▶ mutant3
    ▶ mutant4
    ▶ mutant5
    ▶ mutant6
    ▶ mutant7
    ▶ mutant8
    ▼ mutant9
```

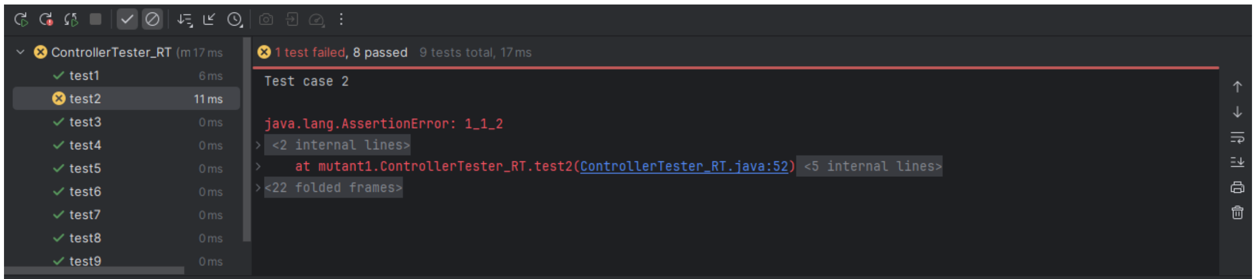
If everything is good, quite MISTA.

Step 6. Open the CruiseControlAssignment project in your Java IDE. You should be able to run ControllerTester_RT.java as JUnit tests. All tests should pass.

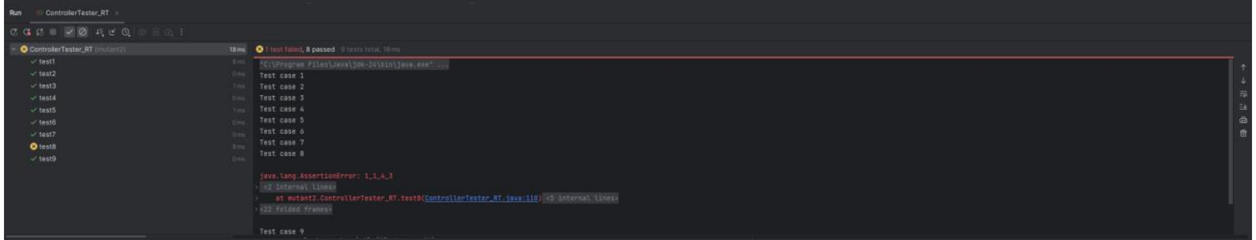
If ControllerTester_RT.java has syntax errors, probably you have performed the above steps incorrectly. You may start it over. If you continue to have problems, contact the instructor.

Step 7. Copy your ControllerTester_RT.java to each mutantX (X=1, 2, …, 9) package and update statement "package correct;" to the correct package, e.g., "package mutant1;". Summarize the test execution results in the following table and provide a screenshot of the test execution result for each mutantX (X=1, 2, …, 9).

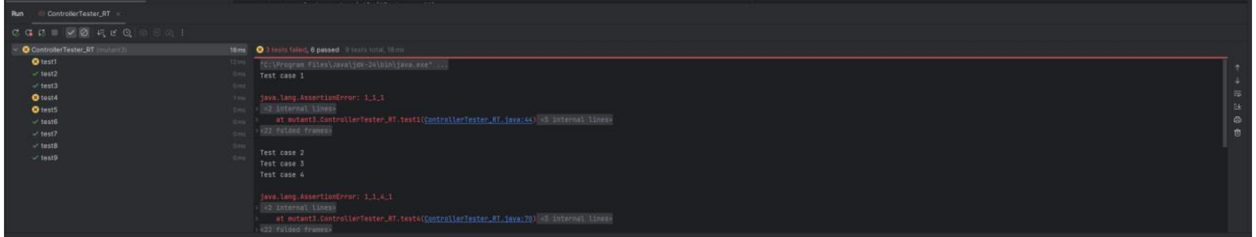
Mutant Version	**Test failure: Yes or No?**
1	Yes
2	Yes
3	Yes
4	Yes
5	No
6	No
7	No
8	No
9	No


```
ControllerTester_RT (m 17 ms
  test1   6 ms
  test2   11 ms
  test3   0 ms
  test4   0 ms
  test5   0 ms
  test6   0 ms
  test7   0 ms
  test8   0 ms
  test9   0 ms
```

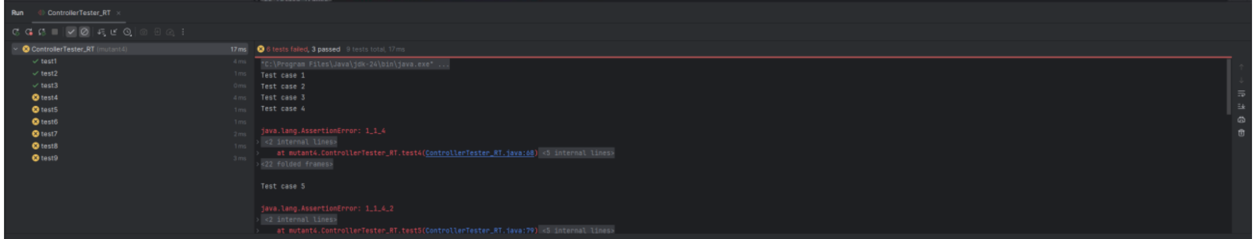
```
1 test failed, 8 passed  9 tests total, 17 ms
Test case 2

java.lang.AssertionError: 1_1_2
<2 internal lines>
    at mutant1.ControllerTester_RT.test2(ControllerTester_RT.java:52) <5 internal lines>
<22 folded frames>
```


```
Run  ControllerTester_RT
ControllerTester_RT (mutant2)  18 ms
  test1 … test9 (test8 failed)
```

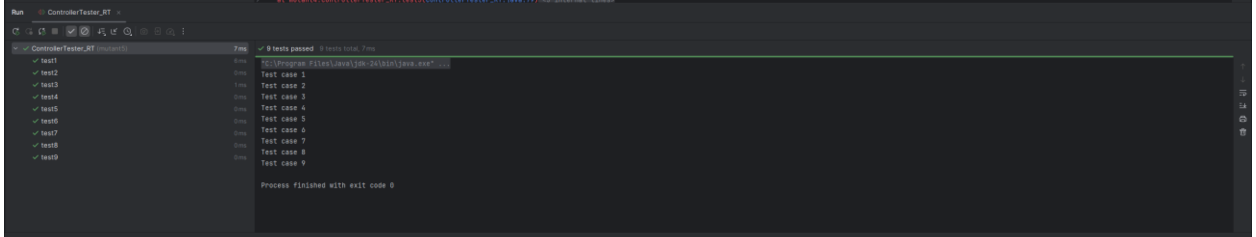
```
1 test failed, 8 passed  9 tests total, 18 ms
"C:\Program Files\Java\jdk-24\bin\java.exe" ...
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6
Test case 7
Test case 8

java.lang.AssertionError: 1_1_4_3
<2 internal lines>
    at mutant2.ControllerTester_RT.test8(ControllerTester_RT.java:110) <5 internal lines>
<22 folded frames>

Test case 9
```


```
Run  ControllerTester_RT
ControllerTester_RT (mutant3)  18 ms
  test1 … test9 (test1, test4, test5 failed)
```

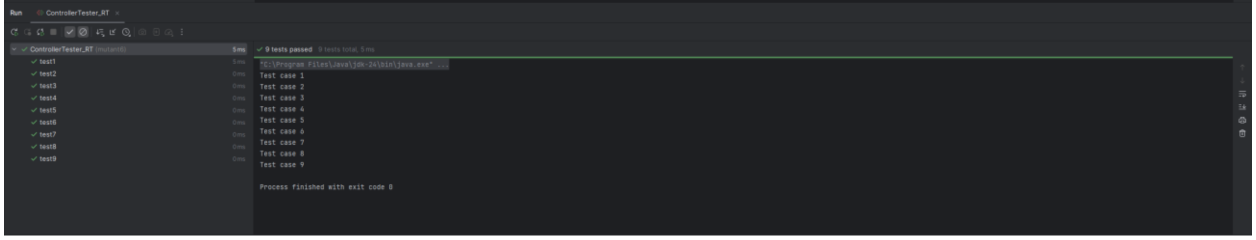
```
3 tests failed, 6 passed  9 tests total, 18 ms
"C:\Program Files\Java\jdk-24\bin\java.exe" ...
Test case 1

java.lang.AssertionError: 1_1_1
<2 internal lines>
    at mutant3.ControllerTester_RT.test1(ControllerTester_RT.java:44) <5 internal lines>
<22 folded frames>

Test case 2
Test case 3
Test case 4

java.lang.AssertionError: 1_1_4_1
<2 internal lines>
    at mutant3.ControllerTester_RT.test4(ControllerTester_RT.java:70) <5 internal lines>
<22 folded frames>
```


```
Run  ControllerTester_RT
ControllerTester_RT (mutant4)  17 ms
  test1 … test9 (test4–test9 failed)
```

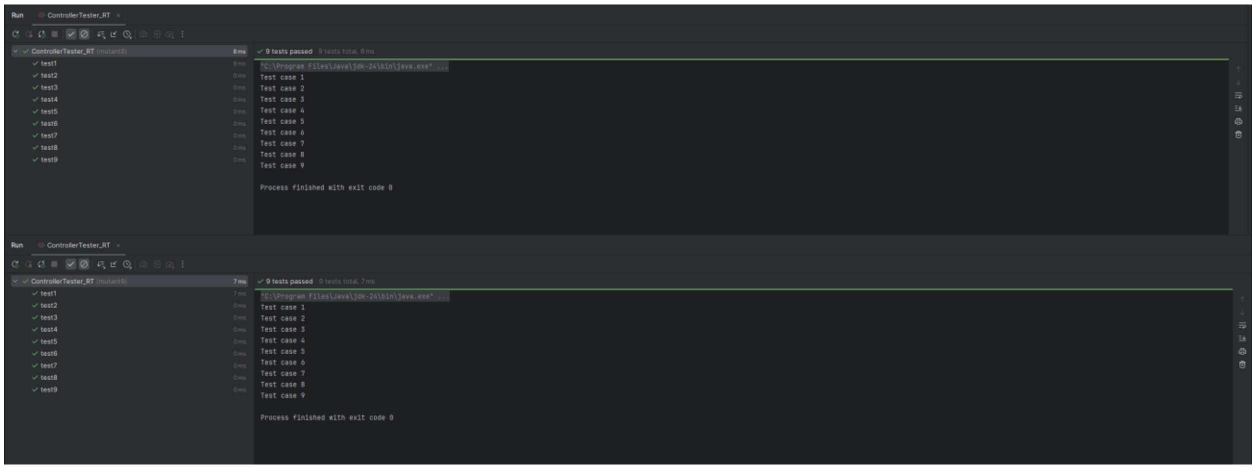
```
6 tests failed, 3 passed  9 tests total, 17 ms
"C:\Program Files\Java\jdk-24\bin\java.exe" ...
Test case 1
Test case 2
Test case 3
Test case 4

java.lang.AssertionError: 1_1_4
<2 internal lines>
    at mutant4.ControllerTester_RT.test4(ControllerTester_RT.java:68) <5 internal lines>
<22 folded frames>

Test case 5

java.lang.AssertionError: 1_1_4_2
<2 internal lines>
    at mutant4.ControllerTester_RT.test5(ControllerTester_RT.java:79) <5 internal lines>
```


```
Run  ControllerTester_RT
ControllerTester_RT (mutant5)  7 ms
  test1 … test9
```

```
9 tests passed  9 tests total, 7 ms
"C:\Program Files\Java\jdk-24\bin\java.exe" ...
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6
Test case 7
Test case 8
Test case 9

Process finished with exit code 0
```


```
Run  ControllerTester_RT
ControllerTester_RT (mutant6)  5 ms
  test1   5 ms
  test2   0 ms
  test3   0 ms
  test4   0 ms
  test5   0 ms
  test6   0 ms
  test7   0 ms
  test8   0 ms
  test9   0 ms
```

```
9 tests passed  9 tests total, 5 ms
"C:\Program Files\Java\jdk-24\bin\java.exe" ...
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6
Test case 7
Test case 8
Test case 9

Process finished with exit code 0
```



```
ControllerTester_RT (mutant8)    8 ms
  test1    8 ms
  test2    0 ms
  test3    0 ms
  test4    0 ms
  test5    0 ms
  test6    0 ms
  test7    0 ms
  test8    0 ms
  test9    0 ms

9 tests passed  9 tests total, 8 ms
"C:\Program Files\Java\jdk-24\bin\java.exe" ...
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6
Test case 7
Test case 8
Test case 9

Process finished with exit code 0
```

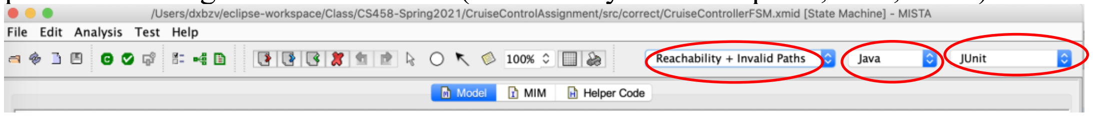
```
ControllerTester_RT (mutant9)    7 ms
  test1    7 ms
  test2    0 ms
  test3    0 ms
  test4    0 ms
  test5    0 ms
  test6    0 ms
  test7    0 ms
  test8    0 ms
  test9    0 ms

9 tests passed  9 tests total, 7 ms
"C:\Program Files\Java\jdk-24\bin\java.exe" ...
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6
Test case 7
Test case 8
Test case 9

Process finished with exit code 0
```

2.2 Round-trip tree with sneak paths

Repeat all the steps in Problem 2.1. The differences are:

Step 2: The two checkboxs may have already been unchecked.

Step 3: Select test generation methods (reachability + invalid paths, Java, JUnit)



/Users/dxbzv/eclipse-workspace/Class/CS458-Spring2021/CruiseControlAssignment/src/correct/CruiseControllerFSM.xmid [State Machine] - MISTA

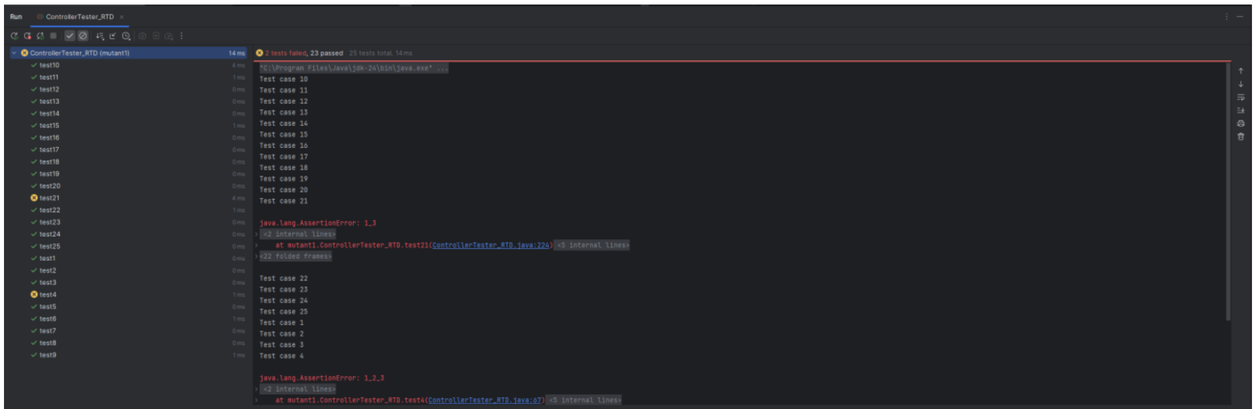
Reachability + Invalid Paths | Java | JUnit

Step 4. Generate the test tree: the tree will be different.

Steps 5-7: The generated test code file is named ControllerTester_RTD.java

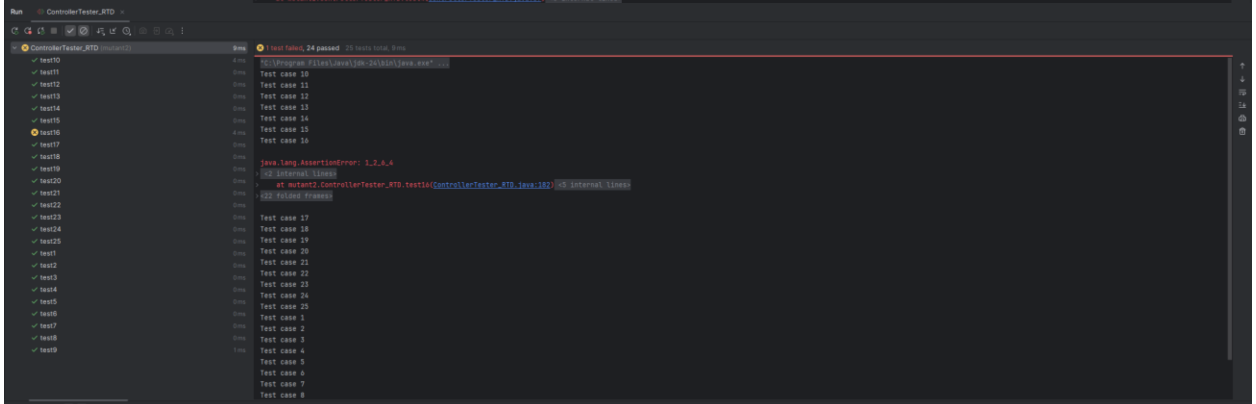
Report the results of step 7 in the following table and provide a screenshot of the test execution result for each mutantX (X=1, 2, …, 9).

Mutant Version	Test failure: Yes or No?
1	Yes
2	Yes
3	Yes
4	Yes
5	Yes
6	Yes
7	No
8	No
9	No



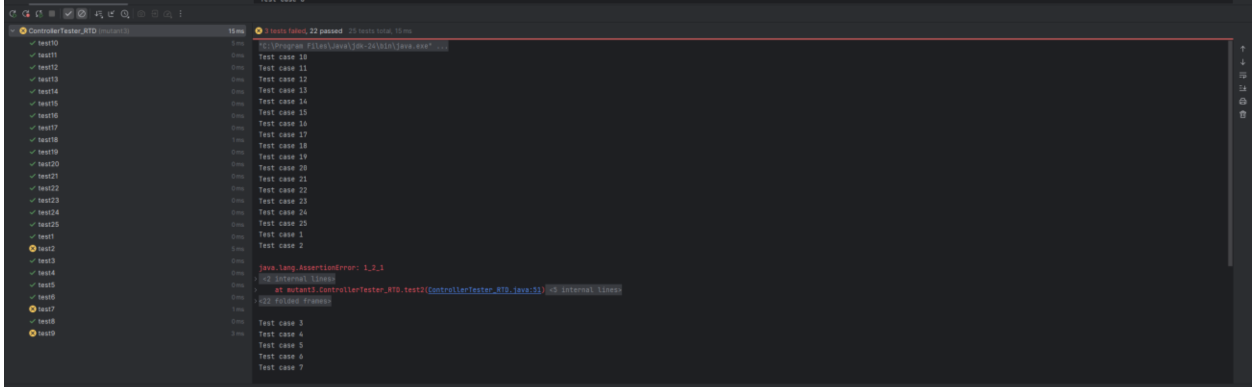
```
ControllerTester_RTD (mutant1)   14 ms
  ✓ test10   4 ms
  ✓ test11   1 ms
  ✓ test12   0 ms
  ✓ test13   0 ms
  ✓ test14   0 ms
  ✓ test15   1 ms
  ✓ test16   0 ms
  ✓ test17   0 ms
  ✓ test18   0 ms
  ✓ test19   0 ms
  ✓ test20   0 ms
  ✗ test21   4 ms
  ✓ test22   1 ms
  ✓ test23   0 ms
  ✓ test24   0 ms
  ✓ test25   0 ms
  ✓ test1    0 ms
  ✓ test2    0 ms
  ✓ test3    0 ms
  ✗ test4    1 ms
  ✓ test5    0 ms
  ✓ test6    1 ms
  ✓ test7    0 ms
  ✓ test8    0 ms
  ✓ test9    1 ms
```

```
2 tests failed, 23 passed  25 tests total, 14 ms
"C:\Program Files\Java\jdk-24\bin\java.exe" ...
Test case 10
Test case 11
Test case 12
Test case 13
Test case 14
Test case 15
Test case 16
Test case 17
Test case 18
Test case 19
Test case 20
Test case 21

java.lang.AssertionError: 1_3
<2 internal lines>
    at mutant1.ControllerTester_RTD.test21(ControllerTester_RTD.java:224) <5 internal lines>
<22 folded frames>

Test case 22
Test case 23
Test case 24
Test case 25
Test case 1
Test case 2
Test case 3
Test case 4

java.lang.AssertionError: 1_2_3
<2 internal lines>
    at mutant1.ControllerTester_RTD.test4(ControllerTester_RTD.java:67) <5 internal lines>
```



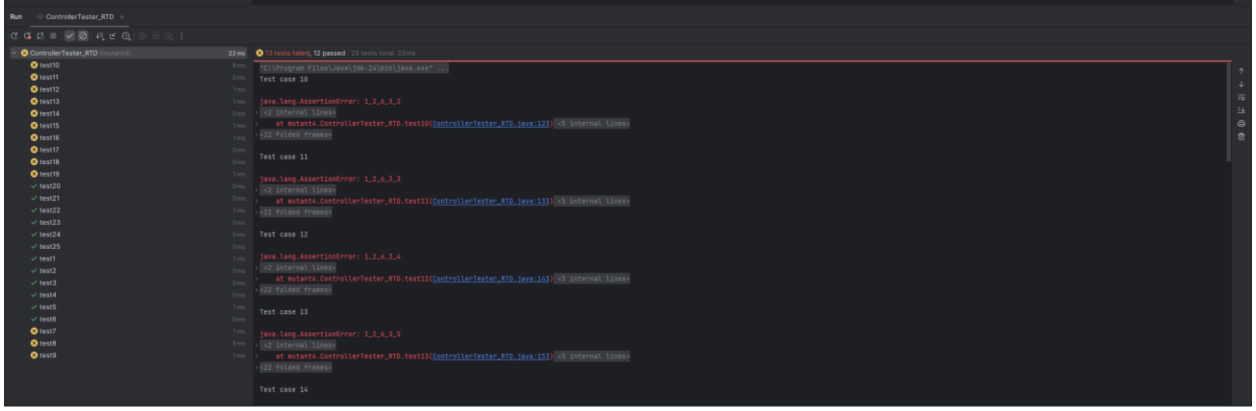
```
ControllerTester_RTD (mutant2)   9 ms
  ✓ test10   4 ms
  ✓ test11   0 ms
  ✓ test12   0 ms
  ✓ test13   0 ms
  ✓ test14   0 ms
  ✓ test15   0 ms
  ✗ test16   4 ms
  ✓ test17   0 ms
  ✓ test18   0 ms
  ✓ test19   0 ms
  ✓ test20   0 ms
  ✓ test21   0 ms
  ✓ test22   0 ms
  ✓ test23   0 ms
  ✓ test24   0 ms
  ✓ test25   0 ms
  ✓ test1    0 ms
  ✓ test2    0 ms
  ✓ test3    0 ms
  ✓ test4    0 ms
  ✓ test5    0 ms
  ✓ test6    0 ms
  ✓ test7    0 ms
  ✓ test8    0 ms
  ✓ test9    1 ms
```

```
1 test failed, 24 passed  25 tests total, 9 ms
"C:\Program Files\Java\jdk-24\bin\java.exe" ...
Test case 10
Test case 11
Test case 12
Test case 13
Test case 14
Test case 15
Test case 16

java.lang.AssertionError: 1_2_6_4
<2 internal lines>
    at mutant2.ControllerTester_RTD.test16(ControllerTester_RTD.java:182) <5 internal lines>
<22 folded frames>

Test case 17
Test case 18
Test case 19
Test case 20
Test case 21
Test case 22
Test case 23
Test case 24
Test case 25
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6
Test case 7
Test case 8
```



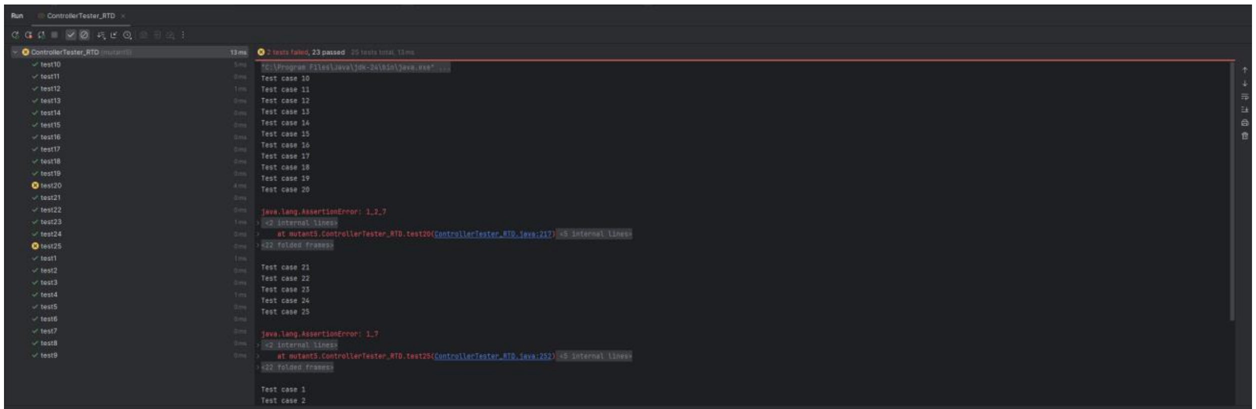
```
ControllerTester_RTD (mutant3)   15 ms
  ✓ test10   5 ms
  ✓ test11   0 ms
  ✓ test12   0 ms
  ✓ test13   0 ms
  ✓ test14   0 ms
  ✓ test15   0 ms
  ✓ test16   0 ms
  ✓ test17   0 ms
  ✓ test18   1 ms
  ✓ test19   0 ms
  ✓ test20   0 ms
  ✓ test21   0 ms
  ✓ test22   0 ms
  ✓ test23   0 ms
  ✓ test24   0 ms
  ✓ test25   0 ms
  ✓ test1    0 ms
  ✗ test2    5 ms
  ✓ test3    0 ms
  ✓ test4    0 ms
  ✓ test5    0 ms
  ✓ test6    0 ms
  ✗ test7    1 ms
  ✓ test8    0 ms
  ✗ test9    3 ms
```

```
3 tests failed, 22 passed  25 tests total, 15 ms
"C:\Program Files\Java\jdk-24\bin\java.exe" ...
Test case 10
Test case 11
Test case 12
Test case 13
Test case 14
Test case 15
Test case 16
Test case 17
Test case 18
Test case 19
Test case 20
Test case 21
Test case 22
Test case 23
Test case 24
Test case 25
Test case 1
Test case 2

java.lang.AssertionError: 1_2_1
<2 internal lines>
    at mutant3.ControllerTester_RTD.test2(ControllerTester_RTD.java:51) <5 internal lines>
<22 folded frames>

Test case 3
Test case 4
Test case 5
Test case 6
Test case 7
```

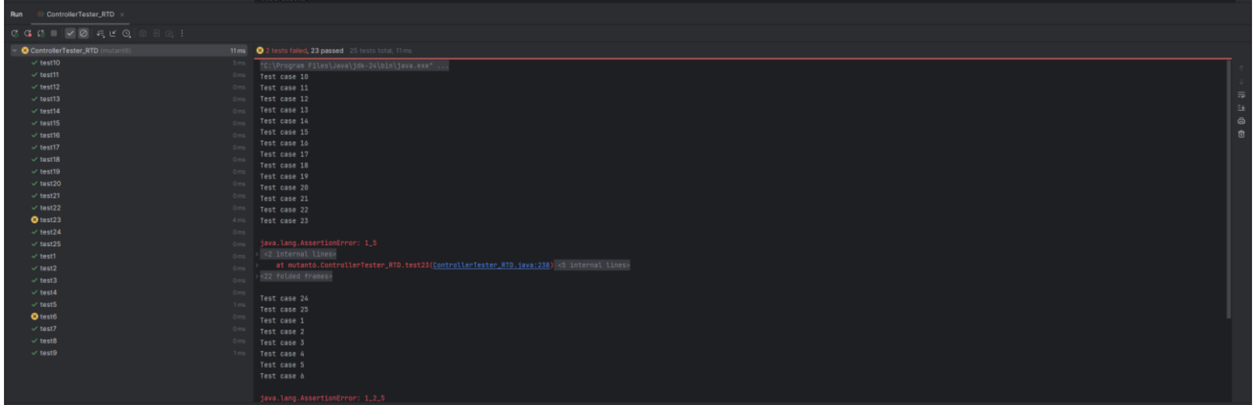


```
ControllerTester_RTD (mutant4)   23 ms
  ✗ test10   8 ms
  ✗ test11   0 ms
  ✗ test12   1 ms
  ✗ test13   1 ms
  ✗ test14   0 ms
  ✗ test15   1 ms
  ✗ test16   1 ms
  ✗ test17   0 ms
  ✗ test18   0 ms
  ✗ test19   1 ms
  ✓ test20   0 ms
  ✓ test21   0 ms
  ✓ test22   1 ms
  ✓ test23   0 ms
  ✓ test24   0 ms
  ✓ test25   0 ms
  ✓ test1    1 ms
  ✓ test2    0 ms
  ✓ test3    0 ms
  ✓ test4    0 ms
  ✓ test5    1 ms
  ✓ test6    0 ms
  ✗ test7    1 ms
  ✗ test8    5 ms
  ✗ test9    1 ms
```

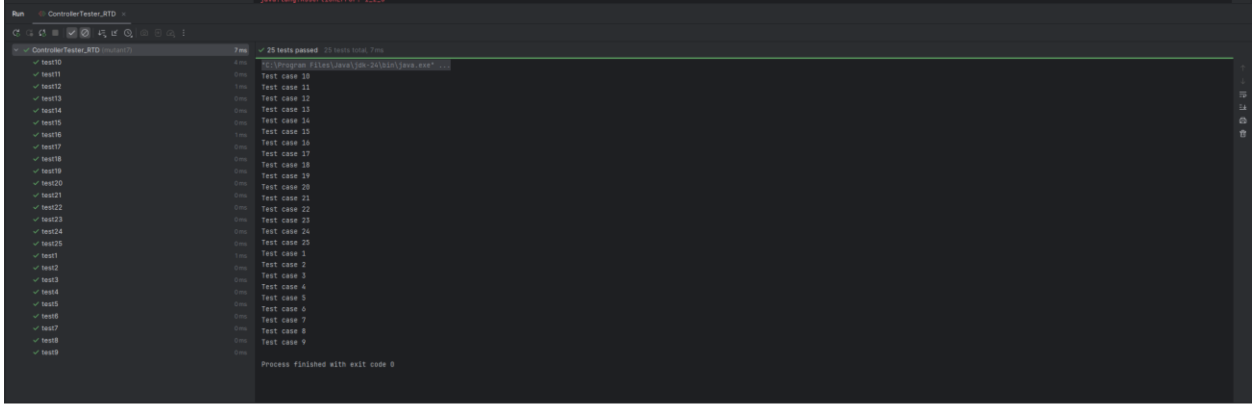
```
13 tests failed, 12 passed  25 tests total, 23 ms
"C:\Program Files\Java\jdk-24\bin\java.exe" ...
Test case 10

java.lang.AssertionError: 1_2_6_3_2
<2 internal lines>
    at mutant4.ControllerTester_RTD.test10(ControllerTester_RTD.java:123) <5 internal lines>
<22 folded frames>

Test case 11

java.lang.AssertionError: 1_2_6_3_3
<2 internal lines>
    at mutant4.ControllerTester_RTD.test11(ControllerTester_RTD.java:133) <5 internal lines>
<22 folded frames>

Test case 12

java.lang.AssertionError: 1_2_6_3_4
<2 internal lines>
    at mutant4.ControllerTester_RTD.test12(ControllerTester_RTD.java:143) <5 internal lines>
<22 folded frames>

Test case 13

java.lang.AssertionError: 1_2_6_3_5
<2 internal lines>
    at mutant4.ControllerTester_RTD.test13(ControllerTester_RTD.java:153) <5 internal lines>
<22 folded frames>

Test case 14
```



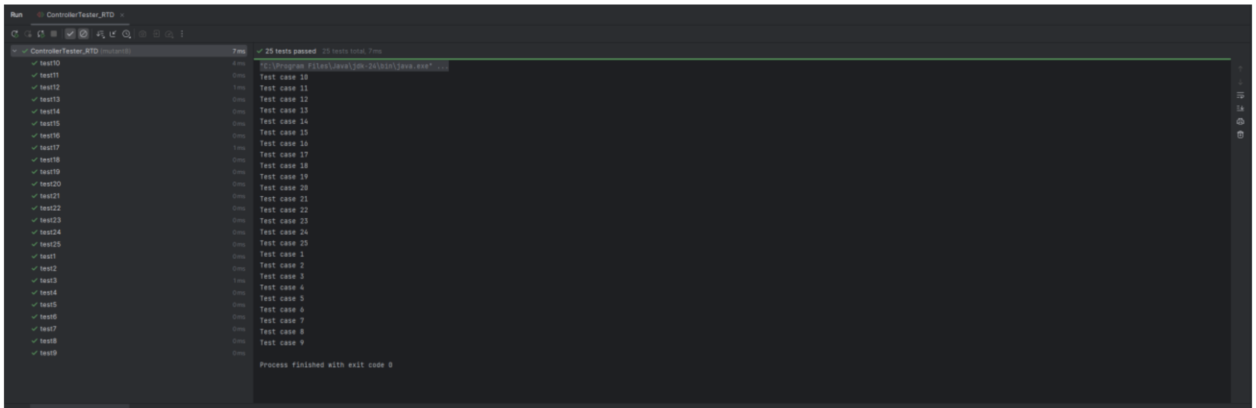
```
Run    ControllerTester_RTD ×

ControllerTester_RTD (mutant5)   13 ms
  test10   5 ms
  test11   0 ms
  test12   1 ms
  test13   0 ms
  test14   0 ms
  test15   0 ms
  test16   0 ms
  test17   0 ms
  test18   0 ms
  test19   0 ms
  test20   4 ms
  test21   0 ms
  test22   0 ms
  test23   1 ms
  test24   0 ms
  test25   0 ms
  test1    1 ms
  test2    0 ms
  test3    0 ms
  test4    1 ms
  test5    0 ms
  test6    0 ms
  test7    0 ms
  test8    0 ms
  test9    0 ms

2 tests failed, 23 passed  25 tests total, 13 ms
"C:\Program Files\Java\jdk-24\bin\java.exe" ...
Test case 10
Test case 11
Test case 12
Test case 13
Test case 14
Test case 15
Test case 16
Test case 17
Test case 18
Test case 19
Test case 20

java.lang.AssertionError: 1_2_7
 <2 internal lines>
    at mutant5.ControllerTester_RTD.test20(ControllerTester_RTD.java:217) <5 internal lines>
<22 folded frames>

Test case 21
Test case 22
Test case 23
Test case 24
Test case 25

java.lang.AssertionError: 1_7
 <2 internal lines>
    at mutant5.ControllerTester_RTD.test25(ControllerTester_RTD.java:252) <5 internal lines>
<22 folded frames>

Test case 1
Test case 2
```



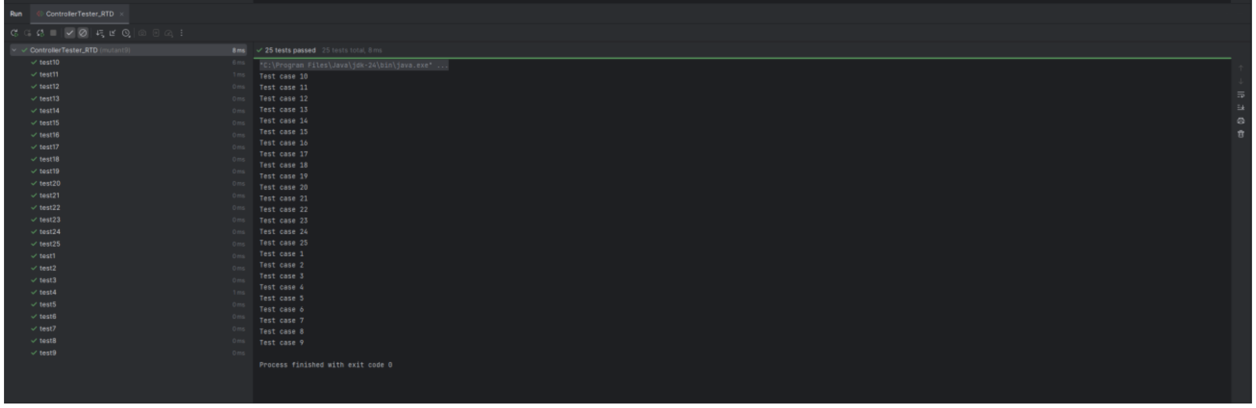
```
Run    ControllerTester_RTD ×

ControllerTester_RTD (mutant6)   11 ms
  test10   5 ms
  test11   0 ms
  test12   0 ms
  test13   0 ms
  test14   0 ms
  test15   0 ms
  test16   0 ms
  test17   0 ms
  test18   0 ms
  test19   0 ms
  test20   0 ms
  test21   0 ms
  test22   0 ms
  test23   4 ms
  test24   0 ms
  test25   0 ms
  test1    0 ms
  test2    0 ms
  test3    0 ms
  test4    0 ms
  test5    1 ms
  test6    0 ms
  test7    0 ms
  test8    0 ms
  test9    1 ms

2 tests failed, 23 passed  25 tests total, 11 ms
"C:\Program Files\Java\jdk-24\bin\java.exe" ...
Test case 10
Test case 11
Test case 12
Test case 13
Test case 14
Test case 15
Test case 16
Test case 17
Test case 18
Test case 19
Test case 20
Test case 21
Test case 22
Test case 23

java.lang.AssertionError: 1_5
 <2 internal lines>
    at mutant6.ControllerTester_RTD.test23(ControllerTester_RTD.java:238) <5 internal lines>
<22 folded frames>

Test case 24
Test case 25
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6

java.lang.AssertionError: 1_2_5
```



```
Run    ControllerTester_RTD ×

ControllerTester_RTD (mutant7)   7 ms
  test10   4 ms
  test11   0 ms
  test12   1 ms
  test13   0 ms
  test14   0 ms
  test15   0 ms
  test16   1 ms
  test17   0 ms
  test18   0 ms
  test19   0 ms
  test20   0 ms
  test21   0 ms
  test22   0 ms
  test23   0 ms
  test24   0 ms
  test25   0 ms
  test1    1 ms
  test2    0 ms
  test3    0 ms
  test4    0 ms
  test5    0 ms
  test6    0 ms
  test7    0 ms
  test8    0 ms
  test9    0 ms

25 tests passed  25 tests total, 7 ms
"C:\Program Files\Java\jdk-24\bin\java.exe" ...
Test case 10
Test case 11
Test case 12
Test case 13
Test case 14
Test case 15
Test case 16
Test case 17
Test case 18
Test case 19
Test case 20
Test case 21
Test case 22
Test case 23
Test case 24
Test case 25
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6
Test case 7
Test case 8
Test case 9

Process finished with exit code 0
```



```
Run    ControllerTester_RTD ×

ControllerTester_RTD (mutant8)   7 ms     ✓ 25 tests passed  25 tests total, 7 ms
  ✓ test10   4 ms                         "C:\Program Files\Java\jdk-24\bin\java.exe" ...
  ✓ test11   0 ms                         Test case 10
  ✓ test12   1 ms                         Test case 11
  ✓ test13   0 ms                         Test case 12
  ✓ test14   0 ms                         Test case 13
  ✓ test15   0 ms                         Test case 14
  ✓ test16   0 ms                         Test case 15
  ✓ test17   1 ms                         Test case 16
  ✓ test18   0 ms                         Test case 17
  ✓ test19   0 ms                         Test case 18
  ✓ test20   0 ms                         Test case 19
  ✓ test21   0 ms                         Test case 20
  ✓ test22   0 ms                         Test case 21
  ✓ test23   0 ms                         Test case 22
  ✓ test24   0 ms                         Test case 23
  ✓ test25   0 ms                         Test case 24
  ✓ test1    0 ms                         Test case 25
  ✓ test2    0 ms                         Test case 1
  ✓ test3    1 ms                         Test case 2
  ✓ test4    0 ms                         Test case 3
  ✓ test5    0 ms                         Test case 4
  ✓ test6    0 ms                         Test case 5
  ✓ test7    0 ms                         Test case 6
  ✓ test8    0 ms                         Test case 7
  ✓ test9    0 ms                         Test case 8
                                          Test case 9

                                          Process finished with exit code 0
```



```
Run    ControllerTester_RTD ×

ControllerTester_RTD (mutant9)   8 ms     ✓ 25 tests passed  25 tests total, 8 ms
  ✓ test10   6 ms                         "C:\Program Files\Java\jdk-24\bin\java.exe" ...
  ✓ test11   1 ms                         Test case 10
  ✓ test12   0 ms                         Test case 11
  ✓ test13   0 ms                         Test case 12
  ✓ test14   0 ms                         Test case 13
  ✓ test15   0 ms                         Test case 14
  ✓ test16   0 ms                         Test case 15
  ✓ test17   0 ms                         Test case 16
  ✓ test18   0 ms                         Test case 17
  ✓ test19   0 ms                         Test case 18
  ✓ test20   0 ms                         Test case 19
  ✓ test21   0 ms                         Test case 20
  ✓ test22   0 ms                         Test case 21
  ✓ test23   0 ms                         Test case 22
  ✓ test24   0 ms                         Test case 23
  ✓ test25   0 ms                         Test case 24
  ✓ test1    0 ms                         Test case 25
  ✓ test2    0 ms                         Test case 1
  ✓ test3    0 ms                         Test case 2
  ✓ test4    1 ms                         Test case 3
  ✓ test5    0 ms                         Test case 4
  ✓ test6    0 ms                         Test case 5
  ✓ test7    0 ms                         Test case 6
  ✓ test8    0 ms                         Test case 7
  ✓ test9    0 ms                         Test case 8
                                          Test case 9

                                          Process finished with exit code 0
```

2.3 Analysis of the testing methods

(a) What are the differences between the round-trip tree in Problem 2.1 and the round-trip tree with sneak paths in Problem 2.2?

The Differences first are that sneak paths catch more errors but also have more tests as in 2.1 we could only catch 1 through 4, but in 2.2 we caught 1 through 6. There is a lot more tests in 2.2 though. The sneak path just negates everything that we have already tested so it checks the negation should be true also.

(b) Compare mutants 7-9 to the correct version and explain why the test suites in Problems 2.1 and 2.2 cannot kill these mutants.

The MISTA-generated tests only check controlState (the Controller's observable state) and do not observe SpeedControl behavior or verify enableControl/disableControl calls. Mutants 7–9 change side effects or guards related to SpeedControl (for example, mutant 7 removes the sc.disableControl() call in engineOff, mutant 8 relaxes the guard for on(), and mutant 9 changes a similar guard/side-effect). Those changes do not alter the controlState on the tested traces, so the tests produced by MISTA do not detect them.

(c) Discuss how the given finite state model can be enhanced so that the tests in the round-trip tree and the round-trip tree with sneak paths generated from the enhanced state model can reveal the bugs in mutants 7-9**.**

The finite state model can be enhanced by adding speed control behavior and is observable into the finite-state model and not just the controller control state. This would enhance it by also checking that we are setting the speed control state when we're testing the controller class.