<<author name redacted>>

Test Plan

# Test XX

<<The purpose of this section is to:

* document test input, specific test procedures, and outcomes.
* establish test methods,
* explain the nature and extent of each test >>

<< for each test case, complete the following: >>

## Test 1

**Objective: The test objective of test 1 is to check that the TableSorter program handles an NXN table of one single input, hence and 1x1 table.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 1 | | | | Current Status: Pending | | |
| Test title: Accepts NXN Table | | | | | | |
| Testing approach: | | | | | | |
| STEP  1  2  3 | OPERATOR ACTION  Include a text file named “nn\_table.txt” that contain an unsorted NXN table.  Create a method: Refer to Appendix in Figure 7.    public void squaredTest () throws Exception{  }  Run Junit class TableSorterTest.java | PURPOSE  To verify that an NXN table is valid.  To test the method return value.  To conduct test | | | EXEPCTED RESULTS  Test returns no error for a NXN table. | COMMENTS  SRS: Table holds an NXN table of integers |
| Concluding Remarks:  If the test case accepts 1X1 (NXN) test, It will accept any NXN table. | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

**Notes:** You must have a text file with one single number.

## Test 2

**Objective: The test objective of test 2 is to check that the TableSorter program handles a table that is not an NXN size table**

**Notes:** <<This area provides general notes concerning the test procedure. Such notes might include comments on how to execute the test procedure, an estimate of the test duration, the requirements of the procedure tests, or a statement of resources needed for this test.>>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 2 | | | | Current Status: Pending | | |
| Test title: Does not accept NXM table | | | | | | |
| Testing approach: <<Included in this section is a description of test harnesses, testing frameworks, environmental requirements, test tools and test automation that will be employed to achieve testing. Include naming conventions for tests and test scripts if appropriate. Provide requirements traceability and test priority.  >> | | | | | | |
| STEP  1  2 | OPERATOR ACTION  Include a text file named “nm\_table.txt” that contains an unsorted NXM size table . Where N and M are different numbers.  Create a method: Refer to Appendix in Figure 6.    public void notSquaredSizeTest () throws Exception{  }  Run Junit class TableSorterTest.java | PURPOSE  To verify that a NXM table is invalid  To test the method return value.  To conduct test | | | EXEPCTED RESULTS  Test returns error for a NxM table. | COMMENTS  SRS: Table holds an NXN table of integers |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

## 

## Test 3

**Objective: Test table that holds positive integers**

**Notes:** <<This area provides general notes concerning the test procedure. Such notes might include comments on how to execute the test procedure, an estimate of the test duration, the requirements of the procedure tests, or a statement of resources needed for this test.>>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 3 | | | | Current Status: Passed | | |
| Test title: Program accepts table that holds all positive integers | | | | | | |
| Testing approach: <<Included in this section is a description of test harnesses, testing frameworks, environmental requirements, test tools and test automation that will be employed to achieve testing. Include naming conventions for tests and test scripts if appropriate. Provide requirements traceability and test priority.  >> | | | | | | |
| STEP  1  2  3 | OPERATOR ACTION  Include a text file named “positiveInt\_array.txt” that contains an unsorted NXN table with all positive numbers.  Create a method: Refer to Appendix in Figure 5.    public void isSortedPositiveInt() throws Exception{  }  Run Junit class TableSorterTest.java | PURPOSE  To verify that a NXM table with all positive numbers is valid  To test the method return value.  To conduct test | | | EXEPCTED RESULTS  Test returns true for table with all positive numbers if its sorted. | COMMENTS  SRS: Table holds an NXN table of integers |
| Concluding Remarks:  0 is considered positive. | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

## Test 4

**Objective: <**< Define the objective of Test XX.Y. >>

**Notes:** <<This area provides general notes concerning the test procedure. Such notes might include comments on how to execute the test procedure, an estimate of the test duration, the requirements of the procedure tests, or a statement of resources needed for this test.>>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 4 | | | | Current Status: Pending | | |
| Test title: Program accepts table that holds all negative integers | | | | | | |
| Testing approach: Using the IDE Eclipse create New>Junit Test Case> Junit 4 Test . | | | | | | |
| STEP  1  2 | OPERATOR ACTION  Include a text file named “negativeInt\_array.txt” that contains an unsorted NXN table with all negative numbers.  Create a method: Refer to Appendix in Figure 4    public void isSortedNegativeInt() throws Exception{  }  Run Junit class TableSorterTest.java | PURPOSE  To verify that a NXM table with all positive numbers is valid  To test the method return value.  To conduct test | | | EXEPCTED RESULTS  Test returns true for table with all negative numbers if its sorted | COMMENTS |
| Concluding Remarks:  After a test returns true for all positive and all negative numbers as acceptable input. You can assume that a combination of both is acceptable. | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

## Test 5

**Objective: <**< Define the objective of Test XX.Y. >>

**Notes:** <<This area provides general notes concerning the test procedure. Such notes might include comments on how to execute the test procedure, an estimate of the test duration, the requirements of the procedure tests, or a statement of resources needed for this test.>>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 5 | | | | Current Status: Pending | | |
| Test title: Program does not accepts table that holds non integers | | | | | | |
| Testing approach: Using the IDE Eclipse create New>Junit Test Case> Junit 4 Test . | | | | | | |
| STEP  1  2  3 | OPERATOR ACTION  Include a text file named “nonInt\_array.txt” that contains an unsorted NXN table with at least one String (non-integer).  Create a method: Refer to Appendix in Figure 3  public void isSortedNonIntArray() throws Exception{  Run Junit class TableSorterTest.java | PURPOSE  To verify that a NXM table with at least one String(non-integer) is not acceptable.  To test the method return value.  To conduct test | | | EXEPCTED RESULTS  Test returns an error for table with Strings in a table. | COMMENTS |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

## Test 6

**Objective: The objective is test that the output of an unsorted table returns false.**

**Notes:** <<This area provides general notes concerning the test procedure. Such notes might include comments on how to execute the test procedure, an estimate of the test duration, the requirements of the procedure tests, or a statement of resources needed for this test.>>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 6 | | | | Current Status: Pending | | |
| Test title: Tests the output of an unsorted table. | | | | | | |
| Testing approach: Testing approach: Using the IDE Eclipse create New>Junit Test Case> Junit 4 Test . | | | | | | |
| STEP  1  2  3 | OPERATOR ACTION  Include a text file named “unsorted\_array.txt” that contains an NXN table that is unsorted.  Create a method: Refer to Appendix in Figure 2    public void isSortedReturnsFalse() throws Exception{  }  Run Junit class TableSorterTest.java | PURPOSE  To test the method return value.  To verify that a NXM table is unsorted | | | EXEPCTED RESULTS  Test returns an false. | COMMENTS |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

## Test 7

**Objective: The objective is test that the output of a sorted table returns true.**

**Notes:** <<This area provides general notes concerning the test procedure. Such notes might include comments on how to execute the test procedure, an estimate of the test duration, the requirements of the procedure tests, or a statement of resources needed for this test.>>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 7 | | | | Current Status: Pending | | |
| Test title: Test the output of a sorted table | | | | | | |
| Testing approach: Using the IDE Eclipse create New>Junit Test Case> Junit 4 Test . | | | | | | |
| STEP  1  2  3 | OPERATOR ACTION  Include a text file named “sorted\_array.txt” that contains an NXN table that is sorted.  Create a method: Refer to Appendix in Figure 1    public void isSortedReturnsTrue() throws Exception{  }  Run Junit class TableSorterTest.java | PURPOSE  The Junit test will need a file to read from.  To test the method return value.  To verify that a NXM table is sorted. | | | EXEPCTED RESULTS  Test returns true. | COMMENTS |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

## Test 8

**Objective: The test objective of test 1 is to check that the TableSorter program handles an NXN table of one single input, hence and 1x1 table.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: 8 | | | | Current Status: Pending | | |
| Test title: Accepts 1x1 Table | | | | | | |
| Testing approach: | | | | | | |
| STEP  1  2  3 | OPERATOR ACTION  Include a text file named “onebyone\_table.txt” that contain 1x1 table.  Create a method: Refer to Appendix in Figure 8.    public void oneDigitTable() throws Exception{  }  Run Junit class TableSorterTest.java | PURPOSE  To verify that an 1x1 table is valid.  To test the method return value.  To conduct test | | | EXEPCTED RESULTS  Test returns true for a 1x1 table. | COMMENTS  SRS: Table holds an NXN table of integers |
| Concluding Remarks:  If the test case accepts 1X1 (NXN) test, It will accept any NXN table. | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

**Notes:** You must have a text file with one single number.

# Appendix

<< possibly more readable to put the expected output here and refer to it in the previous sections. Might also provide explicit directions for analysis of output, if it’s easier to read as an appendix or if analysis is post execution. >>

Figure 1.

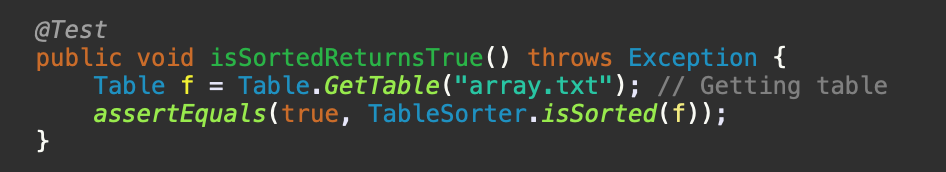


Figure 2.

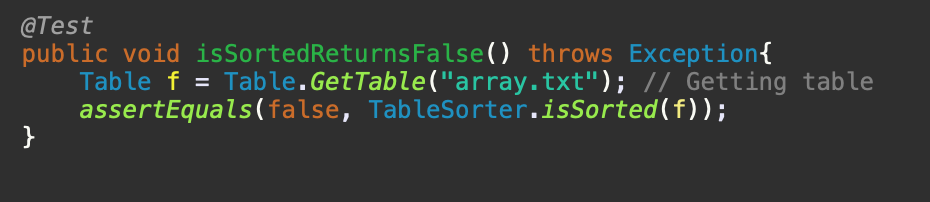


Figure 3.

Test 5

Figure 4.

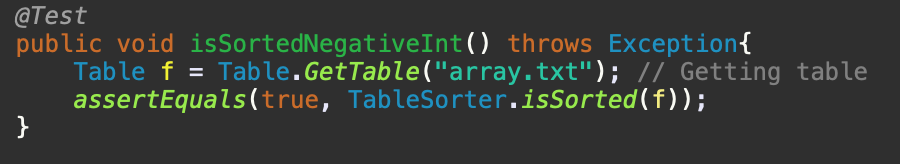


Figure 5.

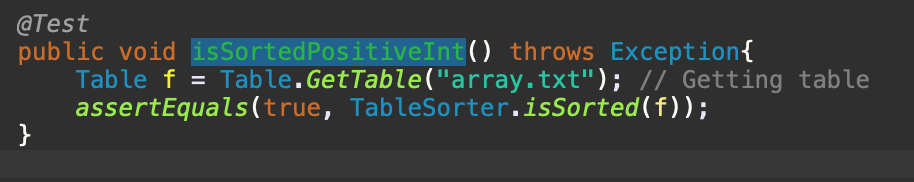


Figure 6.

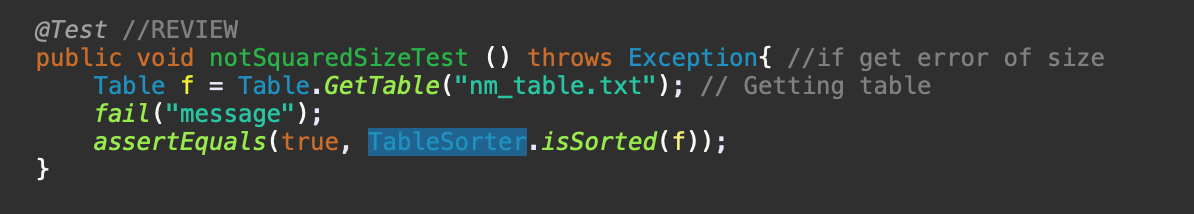


Figure 7.

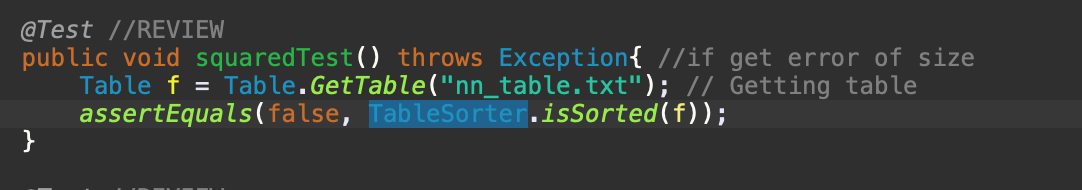


Figure 8.

