Test Set Review for 6120

We have provided feedback on the individual steps of this test plan on the Testing Notes section.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test: ***Black Box #1*** | | | Current Status: ***In progress*** | | |
| Test title: Invalid N input. | | | | | |
| Testing approach: This test consists of four (4) separate steps with different input variations, to help determine how the program responds to invalid ***and*** valid inputs. Test will cover invalid N inputs, such as: (1) Perfect square input, (2) None perfect square input, (3) Empty input and (4) Negative input.  This test is categorized as a ***Black Box (Boundary Value Analysis)*** testing. | | | | | |
| STEP  1.1  1.2  1.3  1.4 | OPERATOR ACTION  ***N =*** 9  ***Values =*** [1, 2,…, 5]  ***N =*** 5  ***Values =*** [1, 2,…, 5]  ***N =*** <*empty*>  ***Values =*** [1, 2,…, 5]  ***N =*** -9  ***Values =*** [1, 2,…, 5] | PURPOSE  To verify that the program will catch and throw an exception if the N input is invalid. | | EXEPCTED RESULTS  *Valid*: The test will go through, and the program will proceed accordingly. Output will be a Table.  *Invalid*: An exception will be thrown stating the N input is invalid. Program will stop. There will be no output. | COMMENTS  Step(s) 1.1 area *Valid.*  Step(s) 1.2, 1.3 and 1.4 are *Invalid.*  N inputs are assuming the Values [] input is correct. |
| **Testing Notes:**  How are input values for N generated and passed through to the system? Do we create a text file with N values? Or do we create a Table object in the program of size N?  For each test, the assumption was made that the tester is required to create text files with N numeric values. The additional assumption was made that the values should be separated by a single space within this text file.  1.1  We created a text file with 9 values {10,9,32,5,6,58,7,8,45}  After inputting the name of the text file with these values, test1.txt we got the following results:    As indicated on the expected results column, this test passed.  1.2  Recreating the same steps as in the previous test, but now with the values {10,9,32,5,8} gave us the following:    Behaving exactly as was indicated on the expected results column.  1.3  Inserting the name of an empty text file resulted in the following:    Behaving exactly as was indicated on the expected results column.  1.4  Following the previously made assumptions, this test cannot be recreated in the same fashion because we cannot make a file with -9 entries | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test: ***Black Box #2*** | | | Current Status: ***In progress*** | | |
| Test title: Valid Values [] input. | | | | | |
| Testing approach: This test consists of three (3) separate steps with different variations, to help determine how the program responds to invalid ***and*** valid inputs. Test will cover valid Values [] inputs, such as: (1) Sorted array, (2) None sorted array and (3) Empty array.  This test is categorized as a ***Black Box (Boundary Value Analysis)*** testing. | | | | | |
| STEP  1.1  1.2  1.3 | OPERATOR ACTION  ***N =*** 9  ***Values =*** [0, 1, 2,…, 5]  ***N =*** 9  ***Values =*** [1, 4, 3, 2, 5, 0]  ***N =*** 9  ***Values =*** [*empty*] | PURPOSE  To verify that the program will be able to handle multiple types of the standard Values [] input. | | EXEPCTED RESULTS  Step(s) 1.1 and 1.2 should both output a sorted 3x3 table containing the input in ascending order.  Step(s) 1.3 should output an empty 3x3 table, since no integers are present in the Values []. | COMMENTS  Values [] inputs are assuming the N input is valid. |
| **Testing Notes:**  For each test, the assumption was made that the tester is required to create text files with 9 numeric values separated by spaces. For test 1.2 the assumption was made that ***Values*** should have 9 entries, not 6 as indicated on the “Operator Action” column.   * 1. and 1.2 With values = {0,1,2,3,4,5,6,7,8} and values {1,4,3,2,5,0,8,6,7}, respectively, the following table was generated.   This is the behavior indicated on the expected results column.  1.3 Following the previously made assumptions, this test cannot be recreated in the same fashion because we cannot make a file with 9 entries that creates an empty table. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test: ***Black Box #3*** | | | Current Status: ***In progress*** | | |
| Test title: Valid/Invalid N & Values [] inputs. | | | | | |
| Testing approach: This test consists of four (4) separate steps with different input variations to help determine how the program responds to invalid ***and*** valid inputs. Test will cover (1) Invalid N / Invalid Values [], (2) Invalid N / Valid Values [], (3) Valid N / Invalid Values [] and (4) Valid N / Valid Values [].  This test is categorized as a ***Black Box*** testing. | | | | | |
| STEP  1.1  1.2  1.3  1.4 | OPERATOR ACTION  ***N =*** -2  ***Values =*** [1, 3, a, 6, b, @]  ***N =*** 3  ***Values =*** [1, 4, 3, 6, 7, 8]  ***N =*** 9  ***Values =*** [1, 3, a, 6, b, @]  ***N =*** 9  ***Values =*** [1, 4, 3, 6, 7, 8] | PURPOSE  To verify that the program will be able to handle and catch multiple types of the standard N and Values [] inputs. | | EXEPCTED RESULTS  Step 1.1: The program will stop and throw an exception on the invalid N and/or invalid Values [] input.  Step 1.2: The program will stop and throw an exception on ***only*** the invalid N input.  Step 1.3: The program will stop and throw an exception on ***only*** the invalid Values [] input.  Step 1.4: The program will run and output an ascending ordered table, containing the values from the Values []. | COMMENTS  Step(s) 1.2 and 1.4 are the basic requirements the program should handle.  Step(s) 1.1 and 1.3 are extra requirements the program is ***recommended*** to handle. To fully test all inputs and variations. |
| **Testing Notes:**   * 1. Following the assumption of creating a text file with N entries to create a table, this test cannot be recreated as it is not possible to create a text file with -2 entries.   2. Following the assumptions that the number of entries on the text file is equal to N, it is not possible to set N=3 and have the values {1,4,3,6,7,8}. We ran the test with the values being {1,4,3} and got the following:     Which, as indicated on the expected behavior column, throws an exception for invalid input.  1.3 We made the assumption that ***Values*** must have 9 entries, rather than 6 as indicated on the “Operator Action” column. We therefore created a text file with N= 9 and values = {0,1,2,@,a,b,6,7,8} which gave the following:    Which, as indicated on the expected behavior column, throws an exception for invalid input.  1.4 We made the assumption that ***Values*** must have 9 entries, rather than 6 as indicated on the “Operator Action” column. We therefore created a text file with N= 9 and values = {0,1,4,3,6,7,8,2,5} which gave the following: | | | | | |

Conclusion

Overall, this test plan works as intended, although many assumptions had to be made and there are some cases where the “Operator Actions” are vague and therefore, the tests did not work. It successfully tests valid and invalid N values and defines what is a “successful” result.