Software Test Plan

# 1.0 Test Plan Identifier

Turing Machine (TM) in C# 1.0.

# 2.0 Test Items

Our fault model includes the following items believed to be high risk in the operation of the Turing Machine in C# 1.0

1. When executing the TM the TM definition file is read into the application it must be received by the parsing algorithm in a specific order of keywords to be a valid TM. See 4.2.0.4d of the Requirements documentation for the order of the keywords.
2. The Parsing Algorithm checks for validity of each data set related to keywords. Need to check that the algorithm only accepts valid input for each category, as referenced in the Requirements Document section 4.2.0.
3. Sections of TM that requires users input. Below outlines all scenarios in which a user interfaces the application, commands that require additional input to function will be labeled sub-commands:
   * Help (h)
   * Show (w)
   * View (v)
   * List (l)
   * Insert (i)
     + Sub-command (string)
   * Delete (d)
     + Sub-command (integer)
   * Set (e)
     + Sub-command (integer)
   * Truncate (t)
     + Sub-command (integer)
   * Run (r)
     + Sub-command (integer)
   * Quit (q)
   * Exit (x)

# 3.0 Features to be Tested

1. The method of reading a text upon program execution.
2. Ensure that each section of the text file is being handled properly by the parsing algorithm.
3. Test user controlled actions, all menu commands and their subcommands.

# 4.0 Approach

For test 2.11, reading and importing data from the TM definition text file, we will use a test driver that will check the validity of all keywords discovered defined by the Requirements Document section 4.2.0.4d. “(number of tests i.e. different definition file)” will be used to ensure that errors are discovered at each keyword if an error in formatting is discovered.

For test 2.12, parsing and data handling we will use a test driver that will check that data is being handled correctly by the associated class objects.

For test 2.13 input commands, we will test using a black-box approach testing all menu functions.

# 5.0 Test Cases

**5.1: TM Definition File Validity**

|  |  |
| --- | --- |
| **Unique Test Identifier** | 5.1.1 |
| **Test Description** | This test accomplishes verifying the TM definition file is able to be opened and read by the Application and adheres to the rules of order defined by section 4.2.0.4d of the Requirements Document. |
| **Component** | Function Name (Enter section of Code this algorithm exists) |
| **Input Condition** | Input = valid text document formatted to Requirements specification |
| **Input State** | Values for STATES, INPUT\_ALPHABET, TRANSITION\_FUNCTION, INITIAL\_STATE, BLANK\_CHARECTER, and FINAL\_STATES contain no data. |
| **Expected Result** | TM application will open txt file and read keywords in the correct order required for a valid TM application. |

|  |  |
| --- | --- |
| **Unique Test Identifier** | 5.1.2 |
| **Test Description** | This test accomplishes verifying the TM definition file is able to be opened and read by the Application and adheres to the rules of order defined by section 4.2.0.4d of the Requirements Document. |
| **Component** | Function Name (Enter section of Code this algorithm exists) |
| **Input Condition** | Input = invalid text document with error in the STATES keyword |
| **Input State** | Values for STATES, INPUT\_ALPHABET, TRANSITION\_FUNCTION, INITIAL\_STATE, BLANK\_CHARECTER, and FINAL\_STATES contain no data. |
| **Expected Result** | TM application will open txt file and read keywords discovering error condition in the STATES keyword resulting in invalid message. |

|  |  |
| --- | --- |
| **Unique Test Identifier** | 5.1.3 |
| **Test Description** | This test accomplishes verifying the TM definition file is able to be opened and read by the Application and adheres to the rules of order defined by section 4.2.0.4d of the Requirements Document. |
| **Component** | Function Name (Enter section of Code this algorithm exists) |
| **Input Condition** | Input = invalid text document with error in the INPUT\_ALPHABET keyword |
| **Input State** | Values for STATES, INPUT\_ALPHABET, TRANSITION\_FUNCTION, INITIAL\_STATE, BLANK\_CHARECTER, and FINAL\_STATES contain no data. |
| **Expected Result** | TM application will open txt file and read keywords discovering error condition in the INPUT\_ALPHABET keyword resulting in invalid message. |

|  |  |
| --- | --- |
| **Unique Test Identifier** | 5.1.4 |
| **Test Description** | This test accomplishes verifying the TM definition file is able to be opened and read by the Application and adheres to the rules of order defined by section 4.2.0.4d of the Requirements Document. |
| **Component** | Function Name (Enter section of Code this algorithm exists) |
| **Input Condition** | Input = invalid text document with error in the TRANSITION\_FUNCTION keyword |
| **Input State** | Values for STATES, INPUT\_ALPHABET, TRANSITION\_FUNCTION, INITIAL\_STATE, BLANK\_CHARECTER, and FINAL\_STATES contain no data. |
| **Expected Result** | TM application will open txt file and read keywords discovering error condition in the TRANSITION\_FUNCTION keyword resulting in invalid message. |

|  |  |
| --- | --- |
| **Unique Test Identifier** | 5.1.5 |
| **Test Description** | This test accomplishes verifying the TM definition file is able to be opened and read by the Application and adheres to the rules of order defined by section 4.2.0.4d of the Requirements Document. |
| **Component** | Function Name (Enter section of Code this algorithm exists) |
| **Input Condition** | Input = invalid text document with error in the INITIAL\_STATE keyword |
| **Input State** | Values for STATES, INPUT\_ALPHABET, TRANSITION\_FUNCTION, INITIAL\_STATE, BLANK\_CHARECTER, and FINAL\_STATES contain no data. |
| **Expected Result** | TM application will open txt file and read keywords discovering error condition in the INITIAL\_STATE keyword resulting in invalid message. |

|  |  |
| --- | --- |
| **Unique Test Identifier** | 5.1.6 |
| **Test Description** | This test accomplishes verifying the TM definition file is able to be opened and read by the Application and adheres to the rules of order defined by section 4.2.0.4d of the Requirements Document. |
| **Component** | Function Name (Enter section of Code this algorithm exists) |
| **Input Condition** | Input = invalid text document with error in the BLANK\_CHARECTER keyword |
| **Input State** | Values for STATES, INPUT\_ALPHABET, TRANSITION\_FUNCTION, INITIAL\_STATE, BLANK\_CHARECTER, and FINAL\_STATES contain no data. |
| **Expected Result** | TM application will open txt file and read keywords discovering error condition in the BLANK\_CHARECTER keyword resulting in invalid message. |

|  |  |
| --- | --- |
| **Unique Test Identifier** | 5.1.7 |
| **Test Description** | This test accomplishes verifying the TM definition file is able to be opened and read by the Application and adheres to the rules of order defined by section 4.2.0.4d of the Requirements Document. |
| **Component** | Function Name (Enter section of Code this algorithm exists) |
| **Input Condition** | Input = invalid text document with error in the FINAL\_STATES keyword |
| **Input State** | Values for STATES, INPUT\_ALPHABET, TRANSITION\_FUNCTION, INITIAL\_STATE, BLANK\_CHARECTER, and FINAL\_STATES contain no data. |
| **Expected Result** | TM application will open txt file and read keywords discovering error condition in the FINAL\_STATES keyword resulting in invalid message. |

**5.2 TM Parsing Algorithm**

|  |  |
| --- | --- |
| **Unique Test Identifier** | 5.2.1 |
| **Test Description** | This test accomplishes verifying the TM definition file is able to be opened and read by the Application and adheres to the rules of order defined by section 4.2.0.4d of the Requirements Document. This test will further address the assignment of data related to section to the appropriate class object. |
| **Component** | Function Name (Enter section of Code this algorithm exists) |
| **Input Condition** | Input = valid text document formatted to Requirements specification including valid data for each keyword. |
| **Input State** | Values for STATES, INPUT\_ALPHABET, TRANSITION\_FUNCTION, INITIAL\_STATE, BLANK\_CHARECTER, and FINAL\_STATES contain no data. |
| **Expected Result** | TM application will open txt file and read keywords in the correct order and assign data to appropriate class object, return of valid will be derived from each class object receiving data. |

|  |  |
| --- | --- |
| **Unique Test Identifier** | 5.2.2 |
| **Test Description** | This test accomplishes verifying the TM definition file is able to be opened and read by the Application and adheres to the rules of order defined by section 4.2.0.4d of the Requirements Document. This test will further address the assignment of data related to section to the appropriate class object. |
| **Component** | Function Name (Enter section of Code this algorithm exists) |
| **Input Condition** | Input = valid text document formatted to Requirements specification containing invalid entries of the keyword data. |
| **Input State** | Values for STATES, INPUT\_ALPHABET, TRANSITION\_FUNCTION, INITIAL\_STATE, BLANK\_CHARECTER, and FINAL\_STATES contain no data. |
| **Expected Result** | TM application will open txt file and read keywords in the correct order and return an appropriate error for each class object receiving invalid data. |

# 5.3 User Input

# 6.0 Test Summary

Summary of results with references to any unresolved problem reports.