Software Test Plan

# Test Plan Identifier

Turing Machine (TM) in C# 1.0.

# 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

* 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.
* 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.
* 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)

# Features to be Tested

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

# Approach

For the approach bullet one above, we will have a series of states to test validity of the input. Waiting for code to further detail this approach.

For the user input commands we will test using black-box approach to test menu functions.

Describe the overall approach to testing: extent of verification of requirements, use of test methods, use of test input files, simulated versus actual use scenarios, differences between test environment and actual use environment,

# Item Pass/Fail Criteria (optional based on risk level)

Specify the criteria to be used to determine whether each feature has passed or failed testing.

# Test Deliverables (optional based on risk level)

Minimum deliverables:

* This document
* Actual test results

# Procedure Steps (optional based on risk level)

The steps required to perform the testing. Typically documents in a checklist. See the attached table below as one example.

# Test Log

Provide an entry for each test run including the version of the software tested, date, tester, associated files used or generated.

# Test Summary

Summary of results with references to any unresolved problem reports.