**Black Box Test Plan**

In this section, you must provide your black-box test plan with at least 5 black-box test cases.

Make sure:

* You describe how to setup the system to begin black-box testing
* Test IDs are uniquely identified and descriptive
* Test descriptions are fully specified with complete inputs, specific values, and preconditions
  + Be sure to provide SPECIFIC INPUTs and VALUEs so that your test cases are repeatable
* Expected results are fully specified with specific output values
* All tests cover scenarios based on the problem statement
* All tests cover unique scenarios for the system
* All strategies for black-box testing are demonstrated in the tests (ECP, BVA, DT)(Equivalence Class Partitioning, Boundary Value Analysis, Decision Table)

Black-box test cases will use the following test files,

valid.txt:   
This is a valid text file. Or is it? No, it is. Unless...? No, it definitely is, for real.

validOneWord.txt:   
dog

validPunctuation.txt:   
Cats... and dogs!? Or dogs, and cats!!

To start the program run CompressionManager.java

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Results** | **Actual Results** |
| **testCompressValid**  **(ECP - Testing compression with a valid .txt file)** | **Preconditions:**   * CompressionManager is running * The file valid.txt exists   **Steps:**  Type **input/valid.txt** when prompted for the input file  Type **c** to compress  Type **output** when prompted for the output directory  *Check results* | The file is compressed into a new file with the path output/valid.316 containing:  0  This is a valid text file. Or 2 it? No, 8 2. Unless...? 9, 8 definitely 2, for real. | The file is compressed into a new file with the path output/valid.316 containing:  0  This is a valid text file. Or 2 it? No, 8 2. Unless...? 9, 8 definitely 2, for real. |  |
| **testDecompressValid**  **(ECP - Testing decompression with a valid .txt file)** | **Preconditions:**   * CompressionManager is running * testCompressValid passed successfully   **Steps:**  Type **output/valid.316** when prompted for the input file  Type **c** to decompress  Type **output** when prompted for the output directory  *Check results* | The file is decompressed into a new file with the path valid.txt containing:  This is a valid text file. Or is it? No, it is. Unless...? No, it definitely is, for real. | The file is decompressed into a new file with the path valid.txt containing:  This is a valid text file. Or is it? No, it is. Unless...? No, it definitely is, for real. |
| **testSingleWordCompress**  **(BVA - Testing compression with a one word file )** | **Preconditions:**   * CompressionManager is running * The file validOneWord.txt exists   **Steps:**  Type **input/validOneWord.txt** when prompted for the input file  Type **c** to compress  Type **output** when prompted for the output directory  *Check results* | The file is compressed into a new file with the path validOneWord.316 containing:  0 dog | The file is compressed into a new file with the path validOneWord.316 containing:  0 dog |
| **testCompressWithPunctuation**  **(DT - Testing valid compression with different input )** | **Preconditions:**   * CompressionManager is running * The file validPunctuation.txt exists   **Steps:**  Type **input/validPunctuation.txt** when prompted for the input file  Type **c** to compress  Type **output** when prompted for the output directory  *Check results* | The file is compressed into a new file with the path validPunctuation.316 containing:  0  Cats... and dogs!? Or 3, 2 cats!! | The file is compressed into a new file with the path validPunctuation.316 containing:  0  Cats... and dogs!? Or 3, 2 cats!! |
| **testDecompressWithPunctuation**  **(DT - Testing valid decompression with different input)** | **Preconditions:**   * CompressionManager is running * testCompressWithPunctuation passed successfully   **Steps:**  Type **output/validPunctuation.316** when prompted for the input file  Type **c** to decompress  Type **output** when prompted for the output directory  *Check results* | The file is decompressed into a new file with the path validPunctuation.txt containing:  Cats... and dogs!? Or dogs, and cats!! | The file is decompressed into a new file with the path validPunctuation.txt containing:  Cats... and dogs!? Or dogs, and cats!! |
| **testGetMostFrequentWords**  **(ECP - Testing most frequent words with valid.txt file)** | **Preconditions:**   * CompressionManager is running * testCompressWithPunctuation passed successfully   **Steps:**  Type **input/valid.txt** when prompted for the input file  Type **f** to enter most frequent words menu  Type **2** when prompted for the number of words to report  *Check results* | Word report is outputted to the console, words are lowercase:  Most Frequent Words Report [  is  it  ] | Word report is outputted to the console, words are lowercase:  Most Frequent Words Report [  is  it  ] |