Name: Aw Yong Jia Min Kellie
Student ID: 1005466

From the lesson Blackbox Testing, the use case diagram is essential to be used as reference such that input space should be derived for each use case. From the input spaces derived, for each input space, an equivalence partition could then be done. Subsequently, for each equivalence partition, one can then proceed to choose a boundary and middle value.

<u>**Equivalence Classes**</u>
Using the idea that input in the same class should behave similarly in the program, for each of the more prominent use cases, I came up with equivalence classes such that the behaviour(output) is similar for each of the classes.

1. **Use Case: Checks number of columns in both CSV files**
   This use case takes in two CSV files as input and produces a boolean value, true if the number of columns in both files are equal and the order of the headers are equal in both files, and false otherwise.
   - ➔ Both empty files:
     The output of all instances of this equivalence class produces a boolean value of true.

   - ➔ At least one filled file:
     The output of all instances of this equivalence class produces a boolean value of false.

2. **Use Case: Get the headers of a CSV file**
   This use case takes in a CSV file as input and produces a list containing the headers of that CSV file.
   - ➔ Empty file:
     The output of all instances of this equivalence class produces an empty list.

   - ➔ Filled file:
     The output of all instances of this equivalence class produces a list containing the headers of the CSV file.

3. **Use Case: Get the index of columns in a CSV file that should be compared**
   This use case takes in the list of headers obtained through a user input and another list of headers obtained from the second use case above and produces a list containing the index of headers that should be used for comparison.
   - ➔ Empty list of headers:
     The output of all instances of this equivalence class produces an empty list.

   - ➔ Two lists of headers with different values:
     The output of all instances of this equivalence class produces a list that contains only the differences such that those differences exist within the original list of headers of the CSV file.

   - ➔ Two lists of headers with equal values:

The output of all instances of this equivalence class produces an empty list.

4. **Use Case: Comparison between 2 CSV files for differences**
   This use case takes in two lists with each containing the content of each of the CSV files (except for the headers) and the list of indexes obtained from the third use case described above and outputs a list containing the differences found.
   - ➔ Empty lists of content of the CSV files:
     The output of all instances of this equivalence class produces an empty list.

   - ➔ At least one filled list of content of the CSV files:
     - ◆ Empty list of index:
       The output of all instances of this equivalence class produces a filled list.

     - ◆ Filled list of index that does not contain the indexes of all elements in either CSV files:
       The output of all instances of this equivalence class produces a filled list.

     - ◆ Filled list of index that contains the indexes of all elements in either CSV files:
       The output of all instances of this equivalence class produces an empty list.

Boundary Value Analysis

1. Use Case: Checks number of columns in both CSV files
   - ➔ Both empty files:
     As this is a special equivalence class where it is a singleton, it does not have any boundary or middle value.

   - ➔ At least one filled file:
     The ***boundary value*** chosen for this would be one empty CSV file and another CSV file having one entry. One would only have to remove one entry in the filled CSV file for it to become another equivalence class, both being an empty file.
     The ***middle value*** chosen for this would be both files being filled. One would have to delete all entries in both files for it to become another equivalence class, both being an empty file.

2. Use Case: Get the headers of a CSV file
   - ➔ Empty file:
     As this is a special equivalence class where it is a singleton, it does not have any boundary or middle value.

   - ➔ Filled file:
     The ***boundary value*** chosen for this would be a CSV file with one column. One would only have to remove one header for it to become another equivalence class, an empty file since it would produce an empty list.

The ***middle value*** chosen for this would be a CSV file with several columns. One would have to remove many headers in order for it to become another equivalence class, an empty file.

3. Use Case: Get the index of columns in a CSV file that should be compared
    - ➔ Empty list of headers:
      As this is a special equivalence class where it is a singleton, it does not have any boundary or middle value.

    - ➔ Two lists of headers with different values:
      The ***boundary value*** chosen for this would be both lists having equal length and having equal values with only one different value. One would only have to change this one value to match the other list in order for it to become another equivalence class, two lists of headers with equal values.
      The ***middle value*** chosen for this would be both lists having different lengths with only a few equal values. One would have to change the length and change multiple values in order for it to become another equivalence class, two lists of headers with equal values.

    - ➔ Two lists of headers with equal values:
      The ***boundary value*** chosen for this would be both lists having one entry each. One would only have to remove one entry in each list for it to become another equivalence class, empty list of headers. One would also only have to change one entry either lists for it to become another equivalence class, two lists of headers with different values.
      The ***middle value*** chosen for this would be both lists having multiple entries. One would have to change all the entries in either lists for it to become another equivalence class, two lists of headers with different values.

4. Use Case: Comparison between 2 CSV files for differences
    - ➔ Empty list of content of the CSV files:
      As this is a special equivalence class where it is a singleton, it does not have any boundary or middle value.

    - ➔ At least one filled list of content of the CSV files:
        - ◆ Empty list of indexes:
          As this is a special equivalence class where it is a singleton, it does not have any boundary or middle value.

        - ◆ Filled list of indexes that does not contain the indexes of all elements in either CSV files:
          The ***boundary value*** chosen for this would be a list of indexes that only has one entry. One only has to remove one entry for it to become another equivalence class, an empty list of indexes.
          The ***middle value*** chosen for this would be a list of indexes that has multiple entries. One would have to remove multiple entries for it to become another equivalence class, an empty list of indexes.

- Filled list of indexes that contains the indexes of all elements in either CSV files:
  The ***boundary value*** chosen for this would be a list of indexes that contains only entry. One would only have to remove that entry for it to become another equivalence class, empty lists of indexes. One would also only have to change one entry for it to become another equivalence class, a filled list of indexes that does not contain the indexes of all elements in either CSV files.
  The ***middle value*** chosen for this would be a list of indexes that has multiple entries. One would have to remove multiple entries or change multiple entries for it to become another equivalence class, either an empty list of indexes or a filled list of indexes that does not contain the indexes of all elements in either CSV files.