COMP10050 Assignment1 Description

Nan Wu 18210184

For question 1, *'input_file'* function is used to read from the input file because this function can figure out the number of characters in each string in the file, and read them correctly.

For question 2, the insertion sort algorithm is used to complete this task, because it is efficient for small data sets, like this task. *"insertionSort"* is built to achieve it.

For question 3, to implement the functionality to find anagrams, *'find_anagrams'* is set up and two functions are declared in this function, '*count_characters'* and '*anagrams'*.

'count_characters' function counts the characters alphabetically in the sentences and stored in InputStrings and stores them in the 2D array *charCount*

'Find_anagrams' function is based on the 2D array *charCount*, it contains many functions. First, a new 2D array called *add_array* is declared and initialised to all 0.

A loop is made here: from the first to the last line, column *p* of row *i* and the column *p* of row *j* of *charCount* are added up together, assigning this value to the position of row *i* and column *j* of *add_array*, then if the value of this position (row *i*, column *j*) is 26, representing all elements in line *i* and *j* in *charCount* are the same, also means these two line's strings are anagrams. If this element is not 26, it will be changed to 0. All elements which are the only 26 in its line will also be changed to 0, because only one element cannot make anagrams.

After this, *'delete_repeat'* function is set for checking if there are two or more rows that are the same, if there are, change the elements in the latter lines to 0.

Last, '*valid_print'* function create two loops which are used to print all elements whose representing strings out. Here, a boolean variable *valid_print* is used, initialised it to false, if there is valid print in this line, *valid_print* becomes true and start a new line to print.

For question 4, *'missing_anagrams'* is set up to find the missing anagrams, three functions are declared in this function, '*sample_array'*, *'delete_ps'* and '*find_missing_anagrams'*.

'*sample_array'* function makes a copy of original *inputStrings* (including punctuation marks and spaces), and then, *'delete_ps'* function deletes all punctuations marks and spaces of the original strings, there are only letters in *inputStrings*.

'*find_missing_anagrams'* contains four functions, 'lettercount' function counts the total number of letters in each string, 'tmp_array' function makes a copy of current *inputStrings* (no punctuation marks and spaces). Then, there is a loop, from the first to the last strings, set a sample to be compared with. In this loop, another loop is built to unify the number of characters in all strings. Then, there are *'count_characters'* function and *'compare'* function.

'count_characters' function counts the characters alphabetically in the sentences and stored in *inputStrings* and stores them in the 2D array *charCount*.

'compare' function determines if every two strings are anagrams after the number of their characters become the same. To achieve it, a variable *counter* is declared and initialised to 0, all 26 elements of *charCount* of two strings are compared, if there exists at least one element is different, *counter* becomes 1. When it's done, If *counter* is still 0, print these strings and number of characters' differences out in its specific format.