AC50002 Programming Languages for Data Engineering  
Python Assignment

## Program Description

I decided that the program would consist of four functions. The main function, one for getting letter values/scores, one for reading in the names from the input file, and one for generating the abbreviations.

In the main function, the first thing that happens is the getValues() function is called. getValues() then opens the values.txt file and stores it in a dictionary for use at a later point. This dictionary is then returned to the main function. Next the getNames() function is called. This function asks the user to input which .txt file they would like to open. This file is then read into the program and returned to the main function. An array to store the abbreviations is then created. The abbreviations are then added to the array by calling the function generateBestAbbr() with the values returned from the previous two functions passed into it.

The first thing to happen in generateBestAbbr() is the creation of two arrays. One for containing the scores for each name, and one for containing all the letters in a name. A for loop is then used to iterate through each name in the array of names. For each name an array called ‘words’ is created using split() which uses spaces as delimiter e.g. if the current name was ‘Crab Apple’ the array will hold ‘crab’ in index 0 and ‘apple’ in index 1. An upper case and joined (with no spaces) version of the current name is then added to the array combined\_names e.g. ‘CRABAPPLE’ is added to the end of the combined\_names array. This array will be used later when generating abbreviations.

After this an empty array is created called scores (this will then be empty each iteration of the for loop). Another for loop is then created within the current for loop which iterates each word in the split name contained in the array ‘words’. A third for loop is then used to iterate over each letter in the current word. The index i used here is able to keep track of the position of the letter in the word. As per the instructions for letter scoring, several if statements are used to give scores to the letters based on their positions in the word. The letter frequency scores are also used where appropriate. These scores are then added to the ‘scores’ array. The accumulated score for the current word are then added to the word\_scores array. All this repeats for each name until there are no names left.

After the scores for each word have been calculated, the complicated task of calculating abbreviations begins. A for loop is used to iterate though each row in combined names. Each row of combined names is a single string of uppercase letters for each name (with no spaces). We then initialise some variables. The variable bestAbbreviation score is initialised as infinity to ensure the first valid score replaces it. Nested for loops are then used iterate though all possible three letter abbreviations. An if statement is used to skip non alphabet characters so they don’t appear in abbreviations. Using the word\_scores array from earlier, a score is generated for each abbreviation. If said abbreviation has a better(lower) score than the current best, the best abbreviation is updated. After checking all possible abbreviations, we’ll have the best which is then added to the end of bestAbbreviations which holds all the best abbreviations for all the names. This is then returned to the main function and outputted to a text file.

To test the program, used the test cases in the assignment brief and a file of lecturers names that I created.