My Implementation: Dictionary Handling and SpellCheckTwoDistance

- First we implement splits that take a word and return a list of tuples that contain parts of a string
- Then we create a list of edited version of a word. This includes adding a letter, deleting a letter, substituting a letter into any position in the word, using the split function
- Then we handle the dictionary file so that it produce a list of words to compare with the list of edited words that we have
- After that, we handle the misspelled file that produce a list of misspelled words
- We then create a function that take a word from the misspelled list, edit it twice, compare it with the dictionary and then produce the suggestions for the word in an output file
- Realizing some flaws to that method, we create another function that use Levenshtein Distance to find the closest suggestion to the original word as possible and produce enough suggestion to use

Working with Anh: We meet up and discuss the problems at least 5-6 times, each session 3 hours. If we don't meet, we write what we think of and push it onto GitHub so that the other can understand/fix it if necessary. We meet up one last time to finalize and submit.