```
# Psuedocode for project 2 CITS1402

# by Luke Bray 22754415


Function main takes in textFile1Path,  textFile2Path and ProfileMethod:
    Check textFile1Path and textFile2Path are valid
    Read in textFile1Path as string
    Read in textfile2Path as string
    For each character in textFile1Path:
        If character is an escape character, delete it
        If the character is “-“ and the character after is “-“:
            Replace the two characters with a space

    If the feature is “punctuation”:
        Call punctuationProfileCalculator with textFile1Data
        Call punctuationProfileCalculator with textFile2Data
    If the feature is “unigrams”:
        Call unigramProfileCalculator with textFile1Data
        Call unigramProfileCalculator with textFile2Data
    If the feature is “conjunctions”:
        Call conjunctionProfileCalculator with textFile1Data
        Call conjunctionProfileCalculator with textFile2Data
    If the feature is “composite”:
        Call compositeProfileCalculator with textFile1Data
        Call compositeProfileCalculator with textFile2Data
    Else:
        Exit the program

    Call calculateScore with calculatedProfile1 and calculatedProfile2
    Output calculatedScore

Function punctuationProfileCalculator takes in textData:
    Define punctuationDict = {“;” : 0,   “,” : 0,   “-“: 0,  “ ‘ “ : 0}
    For character in textData:
        If character = “;” then punctuationDict[“;”] += 1
        If character = “,” then punctuationDict[“,”] += 1
        If character = “-“:
            If the character before and after is a letter:
                punctuationDict[“-“] += 1
        If character = “ ‘ “:  #the space in the string is just to make it easier to read
            If the character before and after is a letter:
                punctuationDict[“ ‘ “] += 1
    Return punctuationDict


#pseudocode continues on next page

```

```
Function unigramProfileCalculator takes in textData:
    Set unigramProfile to blank dictionary
    SplitTextData = split textData on space characters
    For word in splitTextData:
        unigramProfile[word] += 1
    Return(unigramProfile)

Function conjunctionProfileCalculator takes in textData:
    Set conjunctionProfile to blank dictionary
    ValidConjunctions = ["also", "although", "and", "as", "because", "before", "but", "for", "if", "nor", "of", "or", "since", "that", "though", "until", "when", "whenever", "whereas", "which", "while", "yet"]

    For word in (split textData on space characters):
        If word is in ValidConjunctions:
            conjunctionProfile[word] += 1
    Return conjunctionProfile

Function CompositeProfile takes in textData:
    PunctuationProfile = call punctuationProfileCalculator with textData
    conjunctionProfile = call conjunctionProfileCalculator with textData
    For item in conjunctionProfile:
        punctuationProfile[item] = conjunctionProfile[item]
    Return(punctuationProfile)

Function CalculateScore takes in profile1 and profile2:
    sumOfDifferences = 0
    For item in profile1:
        sumOfDifferences +=  (profile1[item] – profile2[item]) ^2
    Score = squareroot(sumOfDifferences)
    Return(score)
```