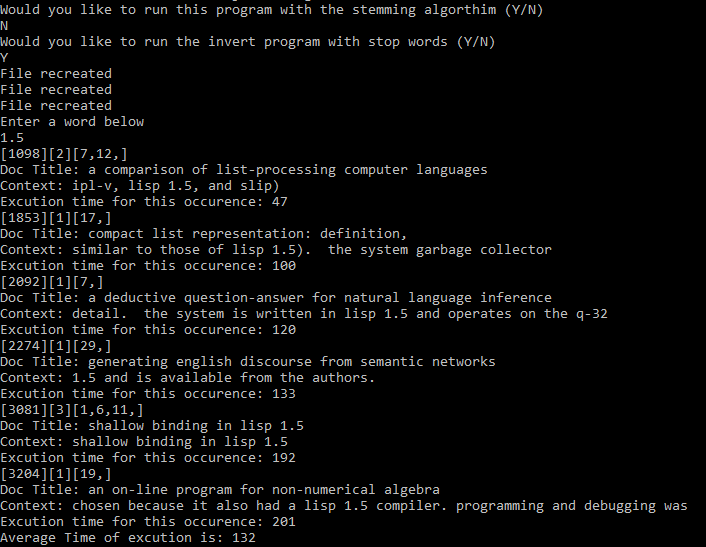
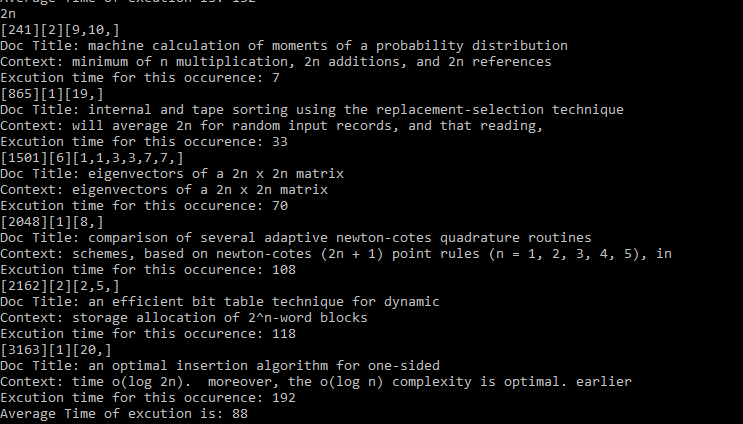
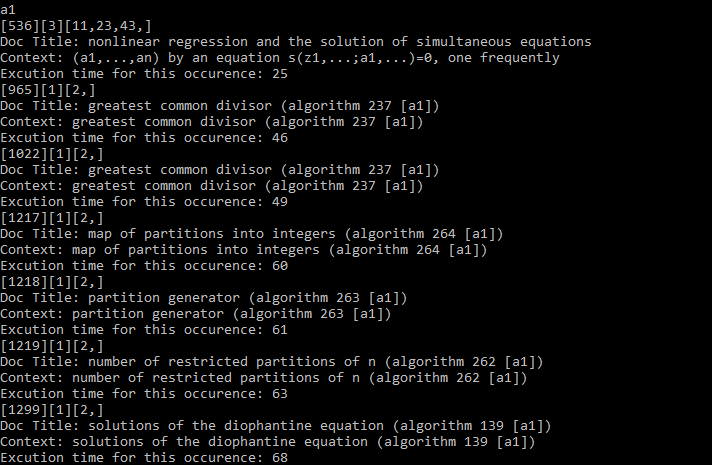
**CPS 842 Assignment 1 Report**

Firstly, is the main method where I create a new file called cacm2.all which is the file that stores the terms from the title and the abstract. This file separates the terms using a delimiter and the replace all function. This was done to ensure that when a word has an apostrophe that the two words are connected into one word. Next, two while loops are used to go through the collections file and print out the terms to cacm2.all file. In the first loop, there are if statements that check if the start of the line begins with a dot and if the next letter is either a I, X, N, B. This was to prevent those lines from being written to the file so that only the terms from the title and the abstract would be written to the file. The second while loop inside the first while loop is used to write the terms to the file. Also, the two while loops are repeated but only runs when the user answers no when they are asked to run the invert program with the stemming algorithm.

Secondly, is the createDictionary method which creates the dictionary file with the terms from the title and the abstract that was stored in the cacm2.all file. The beginning of the method creates the dictionary file and then will move on to the same two while loops used in the main method but with a few modifications. The createDictionary method uses four different data structures which are: hash maps, Array List, Tree, and a Set. In this method there are two hash maps, one is used for storing the word as the key and the document frequency for that word as the value. In addition, the second hash map is used a key value of the word append with the document id and the value was the term frequency in the specific document. Furthermore, there are also three array lists, the first one stores the stop words from the stop words file. The second array list stores the words from the document so that when the while loops go through the second collection file it will prevent the document id from incrementing if the word repeats multiple times. The third array list is a two-dimensional array list that stores the term with all the document ids it appears in. The fourth array list is also two-dimensional, that stores the word appended with the doc id found and the location of the term in the document. The two array lists mentioned above are used when creating the postings file and are created so that the program only traverse the second collection file once. The tree and set are used to sort the first hash map alphabetically before writing it into the dictionary file. Also, the two while loops have if statements that check if the term that the program is reading is a stop word, its length must be greater then 1, and that the term does not begin with “.I”.

Thirdly, is the postingsfile method, when called it creates the postings file using the third and forth array list from the createDictionary method. Before writing to the postings file the postingsfile method sorts both array lists by the first term in the inner array list. Next, the method goes through the sorted hash map one term at a time and goes through both array lists to find the term. It then takes the document id, its term frequency, and the location of the term and prints it out to the postings file.

******Screenshots of example runs**

****

