PROGRAM DESCRIPTION

Operation of Program: The program first looks for all the files in the specified directory, and for each file, gets the tokens by performing various operations (explained below). After all the files are parsed, all the tokens are stored in a list. Next, the open source implementation of PorterStemmer from NLTK is used to stem the gathered tokens. The stems are stored in a list.

Design Decisions: Explained below during discussing the major algorithms and data structures.

1. **How long the program took to acquire the text characteristics.**
   After running the program 5 times and taking the average on UTD csgrads1 UNIX server, the following times were obtained:
   Tokenization time: 1.02 seconds
   Stemming time: 2.52 seconds
   Total time: 3.66 seconds

2. How the program handles:

A. Upper and lower case words (e.g. "People", "people", "Apple", "apple");
All upper-case characters are converted to lower case

B. Words with dashes (e.g. "1996-97", "middle-class", "30-year", "tean-ager")
Words with dashes are considered as separate words (or tokens).

C. Possessives (e.g. "sheriff's", "university's")
The apostrophe is removed, to generate two words (or tokens): for e.g., “sheriff’s” becomes “sheriff” and “s”.

D. Acronyms (e.g., "U.S.", "U.N.")
The dots are removed, and the words are concatenated: for e.g., “U.N.” becomes “UN”.

3. **Briefly discuss your major algorithms and data structures.**
   - Glob is used to get all the files inside the directory
   - ElementTree is used from xml.etree library to parse the file and segregate SGML DOM elements from their values
   - Open source implementation of PorterStemmer from NLTK is used for stemming purposes
   - Preceding and trailing spaces are removed
   - End of line escape character “\n” is removed
   - All non-alphanumeric characters are removed
   - All letters are converted to lowercase
   - Time library is used to calculate the elapsed time

- Counter library is used to calculate the frequency of each occurring word (token)
- Major data structures used are python lists and sets.

Tokenization

1. The number of tokens in the Cranfield text collection: 238060

2. The number of unique words in the Cranfield text collection: 10954

3. The number of words that occur only once in the Cranfield text collection: 4752

4. The 30 most frequent words in the Cranfield text collection: [('the', 19455), ('of', 12717), ('and', 6677), ('a', 5977), ('in', 4645), ('to', 4563), ('is', 4114), ('for', 3493), ('are', 2429), ('with', 2265), ('on', 1944), ('flow', 1849), ('at', 1834), ('by', 1755), ('that', 1570), ('an', 1389), ('be', 1272), ('pressure', 1207), ('boundary', 1156), ('from', 1116), ('as', 1113), ('this', 1081), ('layer', 1002), ('which', 975), ('number', 973), ('results', 885), ('it', 857), ('mach', 824), ('theory', 789), ('shock', 712)]

5. The average number of word tokens per document: 170

Stemming

1. The number of distinct stems in the Cranfield text collection: 8237

2. The number of stems that occur only once in the Cranfield text collection: 3678

4. The 30 most frequent stems in the Cranfield text collection: [('the', 19455), ('of', 12717), ('and', 6677), ('a', 5977), ('in', 4645), ('to', 4563), ('is', 4114), ('for', 3493), ('ar', 2458), ('with', 2265), ('on', 2262), ('flow', 2080), ('at', 1834), ('by', 1755), ('that', 1570), ('an', 1389), ('pressur', 1382), ('be', 1369), ('number', 1347), ('boundari', 1185), ('layer', 1134), ('from', 1116), ('as', 1113), ('result', 1087), ('thi', 1081), ('it', 1044), ('effect', 998), ('which', 975), ('method', 887), ('theori', 882)]

5. The average number of word-stems per document: 170