**Jeremy Ng - 500882192**

**CPS 842 - Assignment 1 Report**

Please note that the Porter Stemmer Algorithm is taken from:
https://tartarus.org/martin/PorterStemmer/python.txt

## Usage

### invert.py

Run code using the command line "python invert.py".
Parameters ("-s", "-stop") to enable stopword removal.
Parameters ("-p", "-porter") to enable porter stemming algorithm.
Example of argument usage: "python invert.py -s -p".

### test.py

Run code using the command line "python test.py".

## Algorithm

### invert.py

- Initiate DocumentCollection class object.
- If stopword removal is enabled then read the stopword file and add all words to set variable DocumentCollection.stopWords.
- Read document collection file and store all relevant information to DocumentCollection.index.
    - Replace punctuation with space and operators with words.
    - If stopword removal is enabled then check each word that is going into the index variable.
- If stemming is enabled then stem every word in DocumentCollection.index
- Create dictionary variable DocumentCollection.dictionary while counting document frequency and term frequency and finding the position of each term.
- Write information from DocumentCollection.dictionary to dictionary.txt and postingsLists.txt.

### test.py

- Initiate Dictionary class object.
- Initiate PorterStemmer class object.
- Query search term until "ZZEND" where the program will stop
    - Get all information relating to the search term: Document frequency, Document index, term frequency, positions, title and relevant context.

**Jeremy Ng - 500882192**

**CPS 842 - Assignment 1 Report**

## Parsing

Characters in the array: ["'s", "'", "-", ".", "(", ")", "{", "}", "[", "]", ":", ";", ",", "", "*", "/", "?", "!", "$", "`"] are replaced with a whitespace.

The following operators are also replaced with words since they represent some meaning.
"<=": " less than or equal to ",
">=": " greater than or equal to ",
"=": " equal to ",
"<": " less than ",
">": " greater than ",
"+": " add ",
"^": " raised to the power of ",
"&": " and ",
"%": " percent ",
"+": " plus "

## Data Structures

Data structures that I have used are dictionaries, arrays and sets.

```
# index key and date are strings. title, abstract and authors are string arrays
DocumentCollection.index[index] = {"title": title, "abstract": abstract, "date": date, "authors": authors}

# set to store all stopwords
DocumentCollection.stopWords = set()

# word key and index key is a string. "df" and "tf" stores int number. position is an int array
DocumentCollection.dictionary[word] = {"df": df, "docID": { index: { "tf": tf, "position": [position]}}}

# Same as DocumentCollection.dictionary
Dictionary.dictionary[word]
```