CSC 849: Information Retrieval
Assignment 1
Inverted Index & Boolean Query Evaluation
100 Points

This assignment consists of three parts.

Part 1: Inverted Index Construction

Write a program that creates an inverted index for a given set of documents. As part of inverted index creation, your program should apply the following pre-processing steps to the input documents, in the given order.

1. Tokenization: Consider every non-alphanumeric character as word boundary.
2. Case normalization: Use all lower case.
3. Stemming: For C++ and Java use the Krovetz stemmer at: http://sourceforge.net/projects/lemur/files/lemur/KrovetzStemmer-3.4/KrovetzStemmer-3.4.tar.gz
   For Python use the Porter stemmer at: http://www.nltk.org/api/nltk.stem.html
4. Stopwords removal: Remove the following words: *the*, *is*, *at*, *of*, *on, and, a*

Part 2: (Simplified) Boolean Query Evaluation

Write a program that can evaluate conjunctive queries with two operands. That is, queries of the form: Term1 AND Term2

Note: Your program does not have to handle any other forms of queries.

The algorithm for evaluating conjunctive queries is as follows:

```
INTERSECT($p_1, p_2$)
answer ← ⟨ ⟩
while $p_1 \neq$ NIL and $p_2 \neq$ NIL
do if $docID(p_1) = docID(p_2)$
      then ADD($answer, docID(p_1)$)
           $p_1 \leftarrow next(p_1)$
           $p_2 \leftarrow next(p_2)$
      else if $docID(p_1) < docID(p_2)$
              then $p_1 \leftarrow next(p_1)$
              else $p_2 \leftarrow next(p_2)$
return answer
```

Part 3: Using your Search Engine

The Parts 1 and 2, together, make a simple search engine. In this last part we will make use of this search engine.

a. Use your program developed for Part 1 to create an inverted index of the collection of documents in *documents.txt* which is on ilearn.
b. Use your program developed for Part 2 along with the inverted index of *documents.txt* to evaluate the following queries. Remember to apply the same pre-processing steps that were applied to the documents (tokenization, case normalization etc.), to the queries too.
    1. asus AND google
    2. screen AND bad
    3. great AND tablet

Finally, upload to ilearn five files:

1. (30 Points) Program from Part 1,
2. (20 Points) Program from Part 2,
3. (20 Points) The inverted index from Part 3a. That is, a file containing the dictionary and the posting lists.
4. (20 Points) The query results from Part 3b.
5. A Readme.txt file giving step by step instructions about how to run your programs. List all the external libraries/programs that your programs need to execute. Also, provide instructions on how to obtain and install these dependencies.

You may use any of the following programming languages: C++, Java, Python/Perl, for this assignment. If you wish to use a different programming language, please contact the course instructor at ak@sfsu.edu.

Important note: Your programs must be well documented. (10 Points).

Good luck and start early!