CS4100 Project 1 Report - Izaak White and Joshua Moner

1. We identified several categories of tokens we need to have. We made a section for comments (single and multi-line). Ex: "//".* and "/*"(.|\n)*?"*/"
   A pair for '#include <_>'. Ex: "#include" and "<"[^>]*">"
   A section for reserve words contained in the programs. Ex: void, int, char, return, while, if, else, do, printf, scanf
   A section for symbols. Ex: ' (',') ',' { ',' } ' , ',' ; '.
   A section for Boolean expressions. <, >, <=, >=, ==, !=
   A section for arithmetic operators and a pair for &<variable> and *<variable>.
   +,-,*,/,%
   "&"[A-Za-z_][A-Za-z0-9_]* and "*"[A-Za-z_][A-Za-z0-9_]* - & and *
   A section to identify the basic types, a regex expression to identify variable names, and we ignored all white spaces.
   [0-9]+ - Numbers
   \"([^\\\n]|(\\.))*?\" - Strings
   [A-Za-z_][A-Za-z0-9_]* - Variable Names
   [\t\n]+ - White Spaces
   . – Wild card
   The program takes in an input file and matches it against a qualify regex expressions and returns a token number based on what it matches to.
2. The implementation of the Winnowing algorithm involved systematically breaking tokenized input strings into overlapping substrings of fixed length, known as k-mers. Each k-mer was individually hashed to produce numeric values, enabling efficient comparison. A sliding window then traversed these hashed values, selecting the minimum hash within each window to establish representative fingerprints for the documents. These fingerprints greatly reduced data complexity while maintaining critical features necessary for accurate similarity analysis. To quantify the similarity between document fingerprints, the Jaccard similarity coefficient was employed. The Jaccard similarity coefficient, often referred to simply as "Jaccard," is a measure used to evaluate how similar two sets are by comparing their commonality. Specifically, it calculates the similarity between two sets by taking the size of their intersection (the number of shared elements) and dividing it by the size of their union (the total number of unique elements present across both sets). The resulting value ranges from 0 to 1, where a value of 0 indicates no commonality, meaning the sets share no elements, and a value of 1 implies complete overlap, indicating the sets are identical. The Jaccard similarity is particularly useful in text analysis, as it effectively accounts for varying document lengths and content structures, providing a normalized comparison measure. After computing

Jaccard similarity scores for each pair of submissions, these pairs were sorted in descending order according to their similarity scores, forming the basis for a comprehensive and interpretable plagiarism report.

3. I believe our approach work very well we found 7 matches of plagiarism and upon inspection they were all identical correctly detecting 100% plagiarism despite comments, names, and whitespace changes.
   Group 1: bills_03.c, bills_09.c, bills_53.c, bills_54.c
   Group 2: bills_10.c, bills_30.c
   From our output of:
   billls_03.c and bills_09.c: 1
   bills_09.c and bills_53.c: 1
   bills_09.c and bills_54.c: 1
   bills_10.c and bills_30.c: 1
   bills_03.c and bills_54.c: 1
   bills_03.c and bills_53.c: 1
   bills_53.c and bills_54.c: 1
   Looking at one that is not a 100% match
   bills_26.c and bills_37.c: 0.977778
   I found that the only difference between these two programs, besides comments, white space and variable names, was that the function ‘print_bills’ in 37 had no ‘ints’ for its parameters in the implementation of it. So I believe that this was a good indicators.
   Looking at a bad match:
   bills_27.c and bills_38.c: 0
   and upon inspection they were entirely different indicating that we were successful again.
   Looking at a ~50% match:
   bills_23.c and bills_24.c: 0.5
   These two had similars but also had some key differences indicating that our approach was successful yet again.
   Upon comparing our data with hand inspection I believe we successfully implementing our Winnowing algorithm and Tokenizer.