Andrew Khazanovich
122008344
CS 2111

Tokenizer

This tokenizer program, programmed in C takes a text file f any length, limited to 50 bytes per line and lexes the file into “tokens” which are are limited to valid hexadecimal, decimal, octal and floating point decimals, or single character invalid tokens delineated by white spaces or line breaks,. It stores the tokens in a structure called TokenizerT which is a Linked List with nodes containing the tokens and their types, which are determined by the program, and prints their types, along with the token in a file called results, and non valid tokens are stored in a file called error.msg as their corresponding hexadecimal value in brackets. The program utilizes the files, main.c, tokenizer.h, and tokenizer.c. main.c calls the methods to create, print and destroy the the data in the tokenizer structure. Tokenizer.h is a header file with signatures for all the methods used in the program. It also includes ctype.h to allow for the use of isspace() method for easier ability to determine white space in all its forms, such as blank character, feed-line and tab.

This program makes a few assumptions about inputs

- The file will have at most 50 bytes per line. If there are more characters then 50 bytes on a line, everything after the 50th byte will be ignored, if a token extends past the 50th byte then the entire token will note be included and ignored
- Each token will either be a valid token of type hexadecimal, decimal, floating point, octal, or will be a single character invalid token. If the file contains invalid tokens that do not match this scope it may be parsed as a valid token of the wrong type.