

Tokenizer.c takes from terminal user input a string and breaks the string down into tokens, each token is then outputted along with what kind of token it is (INTEGER, ALPHANUMERIC, FLOAT, etc.). Our tokenizer.c implements the various mandatory states given in the PDF as well as the extra credit states. Each token is adjacent or separated by white space. We tried our best to emulate what this project seemed to be based off of which is a compiler scanner. Even the terminology is very similar (tokens). For our program, we decided to not use any type of backtracking/forward-tracking functions, which meant our pointer could only make a decision based on the state and the current character it was pointing to. "100." would be an error because at '.' the code would expect this to be a float (as it can not look forward to determine whether it is an integer or float), upon leaving the '.' and moving to '\0', tokenizer will believe this token is malformed. A ZERO state is also implemented for any isolated 0. ".14" is also not a float as according to the FSM, a float must start with an integer. The character that causes an error in each token is outputted in it's hex value. For example, in "0xT", terminal would output ERROR: 0xT Error caused by: [0x54].

C keywords (if, else, for, etc.) are treated as distinct tokens, and thus are recognized as separate from "word" tokens.

Comments of both multiline and single line flavors are also ignored. Anything that follows a multiline comment "/\*" is assumed to be part of that comment until end of file or when a closing comment "\*/" is found. Likewise, anything that follows a single line comment "//" is assumed to be part of that comment until end of file or a new line is detected, to which new tokens will be created.

Anything that follows a double quote " (not including the first double quote from terminal) is assumed to be part of a single token until end of file or when a closing quote " is found. Similarly, anything that follows a single quote ' is assumed to be part of a single token until end of file or a closing quote ' is found.