C Tokenizer

Summary:

The program C Tokenizer takes in a string input and splits it into tokens of different types. The different types of tokens are:

- WORD
- DECIMAL INTEGER
- OCTAL INTEGER
- HEXADECIMAL INTEGER
- FLOAT
- OPERATOR

Each token corresponds to a specific type and the program will print out every token and it's type that the passed in string contains.

Logic:

C Tokenizer utilizes a while loop to parse the string input. Because the program must print each token and it's type as it is detected, a pointer is necessary to remember the location in the string each time the while loop returns a token. Defining a struct T with a pointer makes this possible.

Once the while loop detects a token, it returns the part of the string that is the token. However, the program does not know what type of token the returned string is yet. So, another function checks to see what type of type the returned string is. This function returns an enum so that the program is easier to read and understand.

Operators:

There are 41 possible operators that C Tokenizer recognizes. Each different operator is checked for in a switch case statement. Although the switch case statement is about 200 lines of code, it accounts for every possible combination of operators as efficiently as possible. The switch case statement also accounts for any invalid operators that may be passed in, such as "_", or "{".