Mohit Mori (mam987)
CS211 Computer Architecture

Assignment 1: Tokenizer

Description:

This program takes a string as input and finds the hex, octal, decimal, and float tokens in it. It passes over whitespace characters, and any characters that are not part of a valid token are escape characters. As each token is found, it is printed along with its type, and this continues until the end of the input string is reached. When an escape character is found, it is printed individually as type malformed along with the hex representation of the character.

My tokenizer was implemented using a finite state machine approach. There is a state for each valid type, as well as a malformed state. The program looks at the input string character by character, and depending on what characters are in each token, the program will change the current state to whichever state follows from the current character. Each state function can return the current token to the main function to be printed once the end of token is reached.

Special Cases:

- Multiple 0s (ex. 00000) will be considered an octal
- A decimal point with an E right after (12.E34) will cause the 12 to be considered a decimal with two malformed characters after and then the 34 as a decimal
- An octal with a non-octal digit (0129) will print the the octal (012) and non-octal digit as a decimal (9)