Readme

Our Tokenizer program takes in a string as an argument. It then creates a tokenizer object that consists of two char pointers. One called current that points to where in the original string the program is at and another called token that points to each token encountered. The program parses the string until reaching the end character by continuing to call the function "TKGetNextToken". This function returns the next token in the string and also identifies what type of token it is. If the token is a C_Operator, the function "printOperator" is called. This function uses a switch statement on the first character of the token, and additional checks within each case to determine what C_Operator it is exactly. It then prints the name of that operator, and advances the pointer to the next token in the string. If the token encountered starts with a digit, then the function "getNumber" is called, which identifies what type of number it is (Hex, Decimal, etc.). If no special type, such as C_Operator, C_Keyword, etc., is identified, then the token is given the type "Word", meaning that it is just a word. After identifying exactly what the token is, the program prints the type followed by the actual token itself. It does this for each token on separate lines until the end of the original string is reached. "TKdestroy" is then called before terminating the program to free up the dynamically allocated memory for the tokenizer.

Our program also checks for the extra credit types: Quotes, comments, and C_Keywords.