Deep Shah

Shashank Seeram

CS 214 – Assignment 1

Tokenizer

The program Tokenizer, takes in one string as an input from command line, the string will be then tokenized. In our program the tokens have three attributes; type, length and a pointer. Type specifies the type of token (whether it's word, decimal, hexadecimal, operator, etc.). Length gives the length of the token and the pointer is the actual data with a pointer to the next token. newToken creates a new token with those 3 attributes and also allocated memory for the token.

There is a method delToken which deletes the tokens and frees the memory. The next method is prettyPrint, as the name suggests, the method recognizes each token and prints out the case-specific details in a pretty manner. We tried to incorporate a bunch of operators from the C reference card, however for the operators we missed, the function would still recognize it as an operator and print "operator". We figured the best way to do this was with switch statements. We have a different case for specific types and because the operator type has a lot of operators, we decided to have if-else if statements to recognize specifically which operator the string contains.

isOperator is a method that takes a character and character pointer as an input and it's used to find out whether a certain token is an operator or not. It was a bit tricky to create this because there were certain operators that can be used in a combination (such as less than or equal to, greater than or equal to, etc.). There are methods isLetter and isAlphaNumeric, both take a character as an input and find out whether that certain token is a letter or not and whether a certain token is an alphanumeric or not, respectively. Of course, we need a method that takes the input from the user and tokenizes it, that is done by tokenize method. Tokenize also looks at the input to figure out if the token is alphanumeric, whether it contains hexadecimal/octal values, if the number is a float or if it's a letter.

Then last, but certainly the most important, we have the main method. Main method has an error check so the user must enter a string as an input and there cannot be more than 1 string as the input. The main method then runs the tokenize function with the input and prints out the output in the neat format the assignment required us to use. It also makes sure to free the memory by using the delToken method. We attempted the extra credit, we have the first part where it recognizes whether the string input has if and for and prints them as C keywords. We had made the program using a different algorithm, which worked but it wasn't very efficient, however since we are going to be graded on efficiency, we decided to make changes and create a much more efficient algorithm.