CSS 448 Phase 1, Lexical Analyzer Due April 11

From project description:
"Overall, you read in characters and clump them to create a token. A token is a meaningful unit in the language. You will strip out white space and recognize and return in your yylex routine (what the routine must be called):

- reserved words
- identifiers
- constants
- special characters or groups of characters (e.g., operators, one left parenthesis, left bracket, etc.)
- etc."

The lexical analyzer returns a token, an int, that will be associated with a token, a lexical unit, in the program. Tokens are any group of characters that make up some meaningful sequence in the language. A token can be one character. For example a single left or right parenthesis, an equal sign, a plus sign, a colon, etc. is meaningful in most computer languages. A token can also be a string of characters. For example, all the reserved words, for example, “if” “while” “for” etc. are meaningful in most computer languages. The lexical analyzer must recognize all tokens in the Pascal language. See the document describing the lexical conventions and partial grammar for more details.

You can write the lexical analyzer (scanner) either from scratch or by using (f)lex, a tool for generating programs that perform pattern-matching on text. So that this piece works with the parser, you must call the function that returns the token (the int const) yylex.