**Project-Lexer**

Build both the Lexical Analyzer (Lexer or Scanner) and Parser for the Pasc500 programming language. For some random input program (pasc500test1.p or pasc500test2.p), your Parser is called only once, to decide whether the program is syntactically correct, while Lexer is called repeatedly by Parser, each time the latter needs to advance to the next token of the program.

For the Lexer, I use the flex meta-tool where the tokens are encoded according to the Pasc500 programming language. Also, the information returned by Lexer (token code) will be returned via the yylval variable.

The Lexer:

1. Returns the token code it finds each time it’s called.
2. Returns the names of the identifiers encountered.
3. Returns the values of the constants it encounters. For the non-numeric constants, it returns the string of the word it recognized.
4. Handle parsing errors by printing a message that includes the line number and the line of the input file where the error occurred. Try to continue by erasing the wrong character, and if this is not possible, then terminate it.
5. For the purposes of this project, it should correctly print all the values it returns, in addition to any error messages.

**Project-Bison**