

**Compilers**  
**Phase 1**  
**Lexical Analyzer**  
Due – September 14  
30 Points

Use flex to implement a lexical analyzer for the Modula-2 language. Input to the lexical analyzer comes from `stdin` and output goes to `stdout`.

A link to the software needed for a PC can be found on my home page. There are also links to locations for additional software.

Consider the following table for the extra information that needs to be returned.

Class	Token	Lexeme
Identifier	IDENTSY	Name of identifier
Number	INTCONSTSY	Actual value of number
String	STRCONSTSY	The string
Character	CHARCONSTSY	The ASCII value of character
Reserved word	Assigned token	
Separator	Assigned token	
White space		
Comments		

**Files:** The following files are available from the course home page. These files are written using C. Use them if you want to.

- `main.c` – The driver file that allows you to test your lexical analyzer.
- `m2.tab.h` – The file that defines all the different tokens.

**Hints:**

- All error messages should include a line number. Thus, make sure you count lines in your lexical analyzer.
- Make sure you give an error message for an unterminated string. This is a string for which you reach the newline character before the terminating double quote. I did this as a separate rule for a string.
- Make sure you give an error message for an unterminated comment. This is when you've started a comment and reach end-of-file without reaching the termination characters for the comment.
- If none of your rules match for a token, give an error message.