

<https://github.com/hognogicristina/FLCD/tree/main/Lab8>

Lex file

```
%{
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int lines = 1;
}%

%option noyywrap
%option caseless

DIGIT [0-9]
NON_ZERO_DIGIT [1-9]
INT_CONSTANT [+ -]?{NON_ZERO_DIGIT}{DIGIT}*|0
LETTER [a-zA-Z_]
SIGNS [ !#%^*+ - / < = > _ . , : ; ]
STRING_CONSTANT (\"({LETTER}|{DIGIT}|_|{SIGNS})*\)
IDENTIFIER ({LETTER}|{DIGIT})+
BAD_IDENTIFIER ({DIGIT})+({LETTER})+({LETTER}|{DIGIT})+

%%

"prog"|"int"|"real"|"str"|"char"|"bool"|"read"|"if"|"else"|"write"|"while"|"
"arr"|"sys"|"and"|"or"|"rad" {printf("RESERVED WORD: %s\n", yytext);}

"+"|"-"|"*"|"/"|">"|"<"|"="|"=="|">="|<="|"<>"|"%"|"!=" {printf("OPERATOR:
%s\n", yytext);}

"["|"]"|"{"|"}"|"."|","|":"|";"|"end"|"begin"|"endl"
printf("SEPARATORS: %s\n", yytext);

{IDENTIFIER} {printf("IDENTIFIER: %s\n", yytext);}

{BAD_IDENTIFIER} {printf("Error at token %s at line %d\n", yytext, lines);
exit(1);}

{INT_CONSTANT} {printf("INTEGER CONSTANT: %s\n", yytext);}

{STRING_CONSTANT} {printf("STRING CONSTANT: %s\n", yytext);}

[ \t]+ {}

"/"/(.)*[\n]+ {++lines;}

[\n]+ {++lines;}

. {printf("Error at token %s at line %d\n", yytext, lines); exit(1);}

%%

int main(int argc, char** argv) {
    if (argc > 1)
        yyin = fopen(argv[1], "r");
    else
        yyin = stdin;
}
```

```
yylex();  
}
```

Demo

1. Install flex on MacOS:

```
hognogicristina@Cristinas-MacBook-Air Laboratory 8 % brew install flex
```

2. Generate the Lexer Code:

```
hognogicristina@Cristinas-MacBook-Air Laboratory 8 % flex scanner.lxi
```

3. Compile the Generated C Code:

```
hognogicristina@Cristinas-MacBook-Air Laboratory 8 % gcc -o scanner lex.yy.c -ll
```

4. Run the Lexer:

```
hognogicristina@Cristinas-MacBook-Air Laboratory 8 % ./scanner p1.txt
```

Output:

<https://github.com/hognogicristina/FLCD/blob/main/Lab8/output.txt>