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

1. Write a LEX specification containing the regular expressions corresponding to your language specification: (prog.l)

%{

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int currentLine = 1;

%}

%option noyywrap

IDENTIFIER [a-zA-Z\_][a-zA-Z0-9\_]\*

NUMBER\_CONST 0|[+|-]?[1-9][0-9]\*([.][0-9]\*)?|[+|-]?0[.][0-9]\*

STRING\_CONST \"[a-zA-Z0-9 ]+\"

CHAR\_CONST \'[a-zA-Z0-9]\'

%%

READ|WRITE|DATA|IF|ENDIF|WHILE|ENDWHILE|ELSE|STRUC { printf("Reserved word: %s\n", yytext); }

[+\-\*/%]|LE|GE|EQ|NE|LT|GT|= { printf("Operator: %s\n", yytext); }

[{}():,\.] { printf("Separator: %s\n", yytext); }

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

{NUMBER\_CONST} { printf("Number: %s\n", yytext); }

{STRING\_CONST} { printf("String: %s\n", yytext); }

{CHAR\_CONST} { printf("Character: %s\n", yytext); }

[ \t]+ { /\* skip whitespace \*/ }

[\n]+ { currentLine++; }

[0-9][a-zA-Z0-9\_]\* { printf("Illegal identifier at line %d\n", currentLine); }

[+\-]0 { printf("Illegal numeric constant at line %d\n", currentLine); }

[+\-]?[0][0-9]\*([.][0-9]\*)? { printf("Illegal numeric constant at line %d\n", currentLine); }

\'[a-zA-Z0-9 ]{2,}\'|\'[a-zA-Z0-9][a-zA-Z0-9]\' { printf("Illegal character constant at line %d\n", currentLine); }

\"[a-zA-Z0-9\_]+|[a-zA-Z0-9\_]+\" { printf("Illegal string constant at line %d\n", currentLine); }

%%

int main(int argc, char\*\* argv) {

if (argc > 1) {

FILE \*file = fopen(argv[1], "r");

if (!file) {

fprintf(stderr, "Could not open %s\n", argv[1]);

exit(1);

}

yyin = file;

}

yylex();

return 0;

}

Demo for P1.txt:

A screen shot of a computer

Description automatically generated

Demo for P1err.txt:

A screen shot of a computer

Description automatically generated

Demo for P2.txt:

A screen shot of a computer screen

Description automatically generated

Demo for P3.txt:

A screen shot of a computer

Description automatically generated