<https://github.com/pauladam2001/Sem5_FormalLanguagesAndCompilerDesign/tree/master/Lab8>

**Lex Specification File**

%{

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int currentLine = 1;

%}

%option noyywrap

IDENTIFIER \_[a-z0-9]+\_\*

NUMBER\_CONST ^0|[+|-][1-9]([0-9])\*|[1-9]([0-9])\*$

STRING\_CONST ^"[a-zA-Z0-9\_?!#\*./%+=<>;)(}{ ]+"

CHAR\_CONST ^'[a-zA-Z0-9\_?!#\*./%+=<>;)(}{ ]'

%%

"integer"|"char"|"string"|"FOR"|"WHILE"|"if"|"elseif"|"else"|"puts"|"break"|"return"|"next"|"end" {printf("Reserved word: %s\n", yytext);}

"+"|"-"|"\*"|"/"|"%"|"<="|">="|"=="|"!="|"<"|">"|"=" {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]+ {}

[\n]+ {currentLine++;}

[a-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);}

%%

void main(argc, argv)

int argc;

char\*\* argv;

{

if (argc > 1)

{

FILE \*file;

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

if (!file)

{

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

exit(1);

}

yyin = file;

}

yylex();

}

**Demo**

First run:



Then run:



Then run:



Or the error one:

