https://github.com/lichirea/FLCD/tree/main/Lab12

lang.lxi:

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

int lines = 0;

%}

%option noyywrap

constant 0|[+-]*[1-9][0-9]|[a-zA-Z0-9]*
id [a-zA-Z]*

%%

"slither"|"is"|"if"|"then"|"else"|"while"|"num"|"char"|"input"|"output"|"rattle"|"snake"|"fun"|"print"
{printf("Reserved word: %s\n", yytext);}

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

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

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

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

[\n]+ {lines++;}

[ ]+ {}

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

%%
int main(int argc, char** argv) {
    yyin = stdin;
    yylex();

}
```

Demo:
I create the executable using flex and gcc

```
D:\Data\UBB\An3\SEM5\Languages and Compilers\labs\Lab12>flex lang.lxi

D:\Data\UBB\An3\SEM5\Languages and Compilers\labs\Lab12>gcc lex.yy.c -o my_lex

D:\Data\UBB\An3\SEM5\Languages and Compilers\labs\Lab12>
```

Output of my_lex<p1.rtl (my first program)

```
Identifier: countBiggerThanTwo
Identifier: a
Identifier: b
Identifier: c
Reserved word: rattle
Identifier: count
Reserved word: is
Constant: 0
Separator: ;;
Reserved word: if
Identifier: a
Operator: >
Constant: 2
Reserved word: then
Identifier: count
Reserved word: is
Identifier: count
Operator: +
Constant: 1
Separator: ;;
Reserved word: if
Identifier: b
Operator: >
Constant: 2
Reserved word: then
Identifier: count
Reserved word: is
Identifier: count
Operator: +
Constant: 1
Separator: ;;
Reserved word: if
Identifier: c
Operator: >
Constant: 2
Reserved word: then
Identifier: count
Reserved word: is
Identifier: count
Operator: +
Constant: 1
Separator: ;;
Reserved word: slither
Identifier: count
Separator: ;;
```

Reserved word: snake
Identifier: x
Reserved word: is
Constant: 5
Separator: ;;
Identifier: y
Reserved word: is
Constant: 6
Separator: ;;
Identifier: w
Reserved word: is
Constant: 1
Separator: ;;
Identifier: result
Reserved word: is
Identifier: countBiggerThanTwo
Identifier: x
Identifier: y
Identifier: w
Separator: ;;
Reserved word: print
Identifier: result
Separator: ;;

Output of my_lex<p2.rtl (second program)

Identifier: solveSecondOrder
Identifier: a
Separator: ,
Identifier: b
Separator: ,
Identifier: c
Separator: ,
Identifier: result
Reserved word: rattle
Identifier: c
Reserved word: is
Identifier: c
Operator: -
Identifier: result
Separator: ;;
Identifier: delta
Reserved word: is
Separator: {
Identifier: b
Operator: *
Operator: *
Constant: 2
Separator: }
Operator: -
Separator: {
Constant: 4

Operator: *
Identifier: a
Operator: *
Identifier: c
Separator: }
Constant: sol1
Reserved word: is
Separator: {
Operator: -
Identifier: b
Operator: -
Identifier: sqrt
Identifier: d
Separator: }
Operator: /
Separator: {
Constant: 2
Operator: *
Identifier: a
Separator: }
Constant: sol2
Reserved word: is
Separator: {
Operator: -
Identifier: b
Operator: +
Identifier: sqrt
Identifier: d
Separator: }
Operator: /
Separator: {
Constant: 2
Operator: *
Identifier: a
Separator: }
Reserved word: slither
Constant: sol1
Constant: sol2
Reserved word: snake
Reserved word: print
Identifier: solveSecondOrder
Constant: 1
Constant: 5
Operator: -
Constant: 1
Constant: 5
Separator: ;;

Output of my_lex<p1err.rtl (the incorrect version of p1)

Identifier: countBiggerThanTwo
Identifier: a

Identifier: b
Identifier: c
Reserved word: rattle
Identifier: count
Reserved word: is
Constant: 0
Separator: ;;
Reserved word: if
Identifier: a
Operator: >
Constant: 2
Reserved word: then
Identifier: count
Reserved word: is
Identifier: count
Operator: +
Constant: 1
Separator: ;;
Reserved word: if
Identifier: b
Operator: >
Constant: 2
Reserved word: then
Identifier: count
Reserved word: is
Identifier: count
Operator: +
Constant: 1
Separator: ;;
Reserved word: if
Identifier: c
Operator: >
Constant: 2
Reserved word: then
Identifier: count
Reserved word: is
Identifier: count
Operator: +
Constant: 1
Separator: ;;
Reserved word: slither
Identifier: count
Separator: ;;
Reserved word: snake
Identifier: x
Reserved word: is
Constant: 5
Error at token . at line 12 < -- here is the error
Separator: ;;
Identifier: y
Reserved word: is
Constant: 6

Separator: ;;
Identifier: w
Reserved word: is
Constant: 1
Separator: ;;
Identifier: result
Reserved word: is
Identifier: countBiggerThanTwo
Identifier: x
Identifier: y
Identifier: w
Separator: ;;
Reserved word: print
Identifier: result
Separator: ;;