https://github.com/911-Blajan-Denisa/FLCD

Lex specification file

```
%{
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int current_line = 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 ][\']
```

```
%%


"main"|"program"|"dek"|"strang"|"lasa"|"array"|"and"|"or"|"now"|"medan"|"om"|"annan"|"skriv
a"      {printf("Reserved word: %s\n", yytext);}

"+"|"-"|"*"|"/"|"%"|":="|"IS"|"<"|">"|"<="|">="|"<>"|"!"
{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]+    {current_line++;}


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

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

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

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

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


%%
```

```
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 file %s\n", argv[1]);
        exit(1);
    }
    yyin = file;
}
yylex();
}
```

Demo

1. The first command is: flex lang.lxi

2. The second one is: gcc lex.yy.c

After the second command a.out is generated. For this it can be ran on 4 programs: p1.txt, p2.txt, p3.txt and p1err.txt.

3. The third and final command is: ./a.out p1.txt

After this command we will be able to see the output of the program.