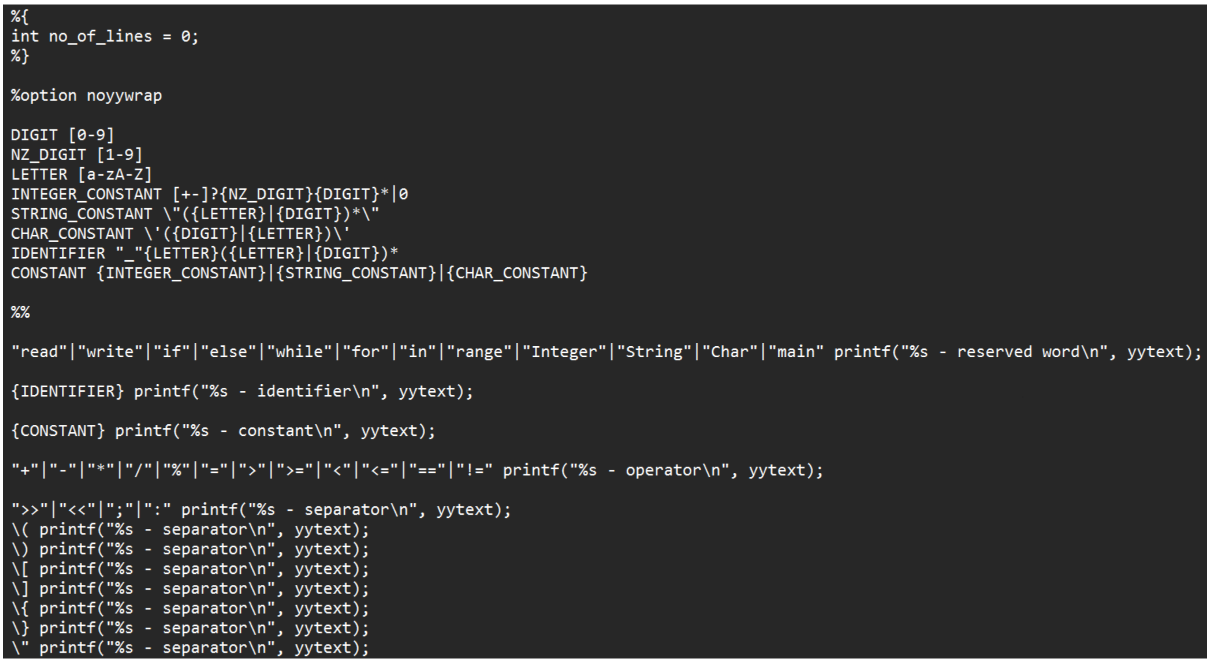
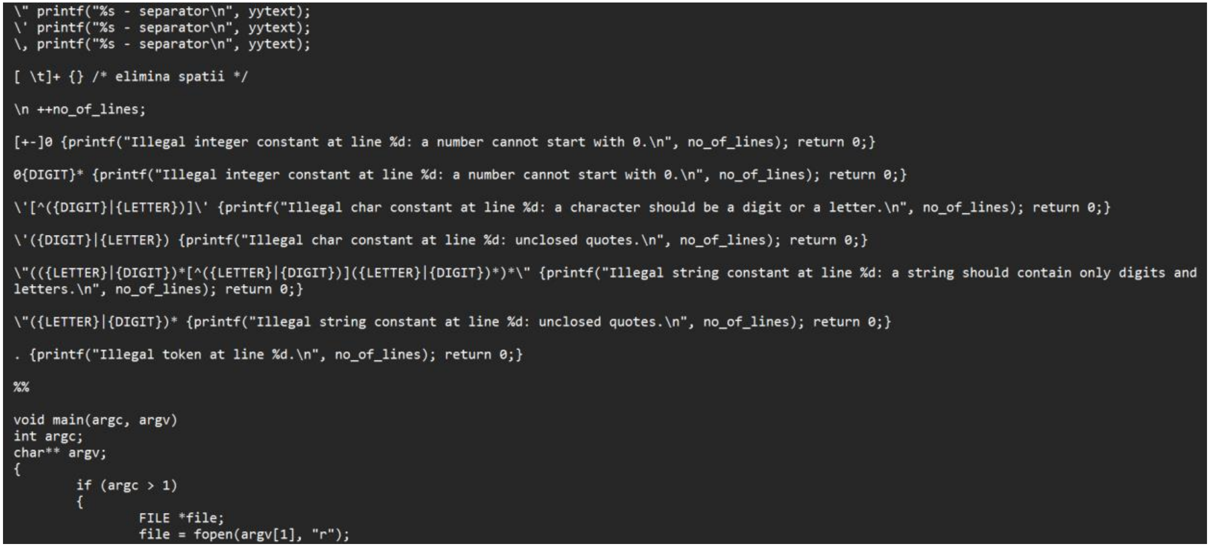
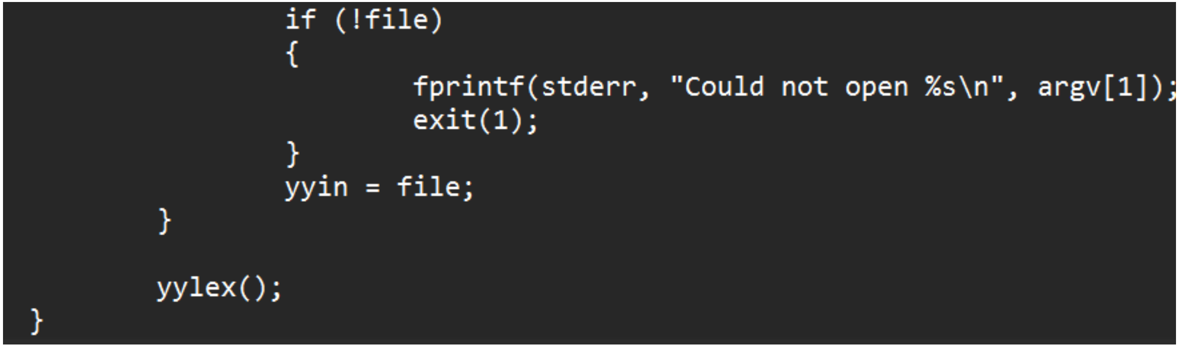
https://github.com/RazvanAndreiMoga/LFTC



```
%{
int no_of_lines = 0;
%}

%option noyywrap

DIGIT [0-9]
NZ_DIGIT [1-9]
LETTER [a-zA-Z]
INTEGER_CONSTANT [+-]?{NZ_DIGIT}{DIGIT}*|0
STRING_CONSTANT \"({LETTER}|{DIGIT})*\"
CHAR_CONSTANT \'({DIGIT}|{LETTER})\'
IDENTIFIER "_"{LETTER}({LETTER}|{DIGIT})*
CONSTANT {INTEGER_CONSTANT}|{STRING_CONSTANT}|{CHAR_CONSTANT}

%%

"read"|"write"|"if"|"else"|"while"|"for"|"in"|"range"|"Integer"|"String"|"Char"|"main" printf("%s - reserved word\n", yytext);

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

{CONSTANT} printf("%s - constant\n", yytext);

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

">>"|"<<"|";"|":" printf("%s - separator\n", yytext);
\( printf("%s - separator\n", yytext);
\) printf("%s - separator\n", yytext);
\[ printf("%s - separator\n", yytext);
\] printf("%s - separator\n", yytext);
\{ printf("%s - separator\n", yytext);
\} printf("%s - separator\n", yytext);
\" printf("%s - separator\n", yytext);
```



```
\" printf("%s - separator\n", yytext);
\' printf("%s - separator\n", yytext);
\, printf("%s - separator\n", yytext);

[ \t]+ {} /* elimina spatii */

\n ++no_of_lines;

[+-]0 {printf("Illegal integer constant at line %d: a number cannot start with 0.\n", no_of_lines); return 0;}

0{DIGIT}* {printf("Illegal integer constant at line %d: a number cannot start with 0.\n", no_of_lines); return 0;}

\'[^({DIGIT}|{LETTER})]\' {printf("Illegal char constant at line %d: a character should be a digit or a letter.\n", no_of_lines); return 0;}

\'({DIGIT}|{LETTER}) {printf("Illegal char constant at line %d: unclosed quotes.\n", no_of_lines); return 0;}

\"(({LETTER}|{DIGIT})*[^({LETTER}|{DIGIT})]({LETTER}|{DIGIT})*)*\" {printf("Illegal string constant at line %d: a string should contain only digits and
letters.\n", no_of_lines); return 0;}

\"({LETTER}|{DIGIT})* {printf("Illegal string constant at line %d: unclosed quotes.\n", no_of_lines); return 0;}

. {printf("Illegal token at line %d.\n", no_of_lines); return 0;}

%%

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();
}
```