https://github.com/Horatzioscool/formal-languages-and-compiler-design

LEX specification:

%option noyywrap

%{

%}

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

INCORRECT\_NUMBER 0[0-9]+

CHAR [a-zA-Z0-9]

WORD \"{CHAR}\*\"

CNST {WORD}{1}|{NUMBER}{1}

SYM [a-z][a-z0-9]\*

INCORRECT\_SYM {CNST}{SYM}

DECLARATION\_KEYWORDS "thus"|"is"|"known"|"as"|"shall"|"be"

TYPE\_KEYWORDS "numeric"|"text"|"complex"

CONDITIONAL\_KEYWORDS "when"|"then"|"when"|"not"

LOOP\_KEYWORDS "while"

ARITHMETIC\_OPERATION\_KEYWORDS "added"|"with"|"subtracted"|"by"|"multipliy"|"divide"

RELATIONAL\_OPERATION\_KEYWORDS "equals"|"greater"|"than"|"lesser"|"different"|"to"

BOOLEAN\_CNST\_KEYWORDS "true"|"false"

IO\_KEYWORDS "read"|"written"

KEYWORD {DECLARATION\_KEYWORDS}{1}|{TYPE\_KEYWORDS}{1}|{CONDITIONAL\_KEYWORDS}{1}|{LOOP\_KEYWORDS}{1}|{ARITHMETIC\_OPERATION\_KEYWORDS}{1}|{RELATIONAL\_OPERATION\_KEYWORDS}{1}|{BOOLEAN\_CNST\_KEYWORDS}{1}|{IO\_KEYWORDS}{1}

DELIMITERS \.|:|;|,

%%

{INCORRECT\_NUMBER} printf("Incorrect number detected: %s\n", yytext);

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

{KEYWORD} printf("A keyword: %s\n", yytext);

{INCORRECT\_SYM} printf("An Incorrect Identifier: %s\n", yytext);

{SYM} printf("An identifier: %s\n", yytext);

{DELIMITERS} printf("A delimiter: %s\n", yytext);

\(.\*\) /\* Comments \*/

[ \t\r\n]+ /\* eat up whitespace \*/

. printf("Error, unrecognized char: %s \n", yytext);

%%

main( argc, argv )

int argc;

char \*\*argv;

{

++argv, --argc; /\* skip over program name \*/

if ( argc > 0 ) {

printf("Using file\n");

yyin = fopen( argv[0], "r" );

}

else {

printf("Using stdin");

yyin = stdin;

}

yylex();

return 0;

}