https://github.com/saltylex/milanguage/tree/main

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

%option noyywrap

/* Definitions */

letter          [A-Za-z_]
space [ ]
digit           [0-9]
symbol [-]
string_c (\"({letter}|{digit}|{space}|{symbol})*\")
identifier ({letter})({letter}|{digit})*
wrong_id ({digit})({letter}|{digit})*


/* Rules */
%%
{digit}+            { printf("An integer: %s\n", yytext); }
{digit}+"."{digit}*  { printf("A float: %s\n", yytext); }
"milreturnezi"|"milnumar"|"milsir"|"milintrebi"|"milaltfel"|"miltimp"|"milpentru"|"milafisezi"|"milcitesti"     { printf("Reserved Word:
%s\n", yytext); }
{string_c}   { printf("String: %s\n", yytext); }
{identifier}  { printf("Identifier: %s\n", yytext); }
{wrong_id} { printf("Wrong identifier!: %s\n", yytext); }
"+"|"-"|"*"|"/"|"%"      { printf("An operator: %s\n", yytext); }
"=="|"<"|"<="|">"|">=" { printf("Comparison operator: %s\n", yytext); }
"="|"+="|"-="|"*="|"/=" { printf("Assignment operator: %s\n", yytext); }

"{"                 { printf("Left Brace\n"); }
"}"                 { printf("Right Brace\n"); }
"("                 { printf("Left Parenthesis\n"); }
")"                 { printf("Right Parenthesis\n"); }
";"                 { printf("Semicolon\n"); }
","                 { printf("Separator\n"); }
[ \t]+              /* eat up whitespace */
\n                  { printf("Newline\n"); }
"<<"                { printf("Write\n"); }
">>"                { printf("Read\n"); }
.                  { printf("Lexical Error!: %s\n", yytext); }

%%

int main(int argc, char **argv) {
    ++argv,  - argc;
    if (argc > 0)
        yyin = fopen(argv[0], "r");
    else
        yyin = stdin;
    yylex();
    return 0;
}
```