

```

%{
#include<string.h>
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>

enum KindToken { INT, FLOAT, STRING, LP, RP, LB, RB, PLUS,MOINS,MULT,DIV,PV,EGAL
};
struct Token { KindToken kind;
    union { int the_int; float the_float; char * the_string; } Value; };
int nblex= 0;
Token lexemes[1000];
}%
digit [0-9]
lettre [a-zA-Z]
%%
" |\t|\n {;}
{lettre}({lettre}|{digit})* {
    lexemes[nblex].kind=STRING;
    int l=strlen( yytext) + 1;
    char *copy = new char[l];
    strcpy( copy, yytext);
    lexemes[nblex].Value.the_string=copy;
    /*printf("coucou ici yytext=%s\n", yytext);*/
    nblex++; }
{digit}+
{ ext) ; nblex++; }
{digit}+\.{digit}* { lexemes[nblex].kind=FLOAT; lexemes[nblex].Value.the_float =
atof(yytext) ; nblex++; }
"\-" { lexemes[nblex].kind=MOINS; nblex++; }
"\/" { lexemes[nblex].kind=DIV; nblex++; }
"=" { lexemes[nblex].kind=EGAL; nblex++; }
"+" { lexemes[nblex].kind=PLUS; nblex++; }
"*" { lexemes[nblex].kind=MULT; nblex++; }
"(" { lexemes[nblex].kind=LP; nblex++; }
")" { lexemes[nblex].kind=RP; nblex++; }
"." { lexemes[nblex].kind=PV; nblex++; }
{ printf( " non reconnu: %s\n", yytext); }
%%
main( void )
{
    yylex();
    for( int i=0; i<nblex; i++)
        switch (lexemes[i].kind)
        {
            case STRING: printf( "STRING %s\n", lexemes[i].Value.the_string); break;
            case INT: printf( "INT %d\n", lexemes[i].Value.the_int); break;
            case FLOAT: printf( "FLOAT %f\n", lexemes[i].Value.the_float); break;
            case PLUS: printf( "PLUS +\n"); break;
            case MOINS: printf( "MOINS -\n"); break;
            case MULT: printf( "MULT *\n"); break;
            case DIV: printf( "DIV /\n"); break;
            case PV: printf( "PV ;\n"); break;
            case LP: printf( "LP (\n"); break;
            case RP: printf( "RP )\n"); break;
            default : printf( "heu\n"); break;
        }
    return 0;
}

```