

1 Fichier lex polish.lex

```
%{
/* pas de fichier inclus */
}%
chiffre [0-9]
lettre [a-zA-Z]
ident {lettre}{{lettre}|{chiffre}}*
%%
"_"|\t|\n      { /* sauter les espaces */}
{chiffre}+" \." {chiffre}* { yylval = atof(yytext); return DOUBLE ; }
{chiffre}+      { yylval = (double) atoi(yytext) ;
return INT ; }
"\-"           { return OPP ; }
"/"           { return INV ; }
"+"           { return PLUS ; }
"*"           { return TIMES ; }
"\^"          { return PUISS ; }
"("           { return LPAREN ; }
")"           { return RPAREN ; }
";"           { return PV ; }
"."           { return ERREUR ; }
```

2 Fichier yacc polish.y

```
%{
#define YYSTYPE double
#include <math.h>
#include <stdio.h>
int ylex();
int yerror( char *);
int yyparse();
}%
%token DOUBLE INT PLUS PUISS TIMES INV PV OPP LPAREN RPAREN ERREUR
%%
input : {} | input line ;
line : expr PV { printf ("\t=%.10g\n", $1); } ;
expr : DOUBLE { $$ = $1; }
    | INT { $$ = $1; }
    | expr expr PLUS { $$ = $1 + $2; }
    | expr expr TIMES { $$ = $1 * $2; }
    | expr expr PUISS { $$ = pow ($1, $2); }
    | expr OPP { $$ = -$1; }
```

```

    | expr INV          { $$ = 1. / $1; }
;
%%
#include "lex.yy.c"
int yyerror (char *s) { /* Called by yyparse on error */
printf ("Ici yyerror: %s\n", s); return 0; }
int main(int argc, char **argv) { yyparse(); return 0; }

```

3 Fichier makefile

```

ok : polish.y polish.lex
    lex polish.lex
    yacc -o polish.cpp polish.y
    g++ -o polish polish.cpp -lfl -lc

```

4 Session

```

polish
123 ;
    =123
1 2 + ;
    =3
1 2 5 * + ;
    =11
1 - ;
    =-1
99 - - ;
    =99
3 / ;
    =0.3333333333
3 2 ^ ;
    =9
16 0.5 ^ ;
    =4
2 0.5 ^ ;
    =1.414213562
2 0.5 ^ - ;
    =-1.414213562

```