

Lex et Yacc, exemple de somme ou produit de liste d'entiers

D. Michelucci

1 fichier makefile

```
ok: exemple_liste liste
liste: liste.lex liste.yacc
      lex liste.lex
      yacc -o liste.cpp liste.yacc
      g++ -o liste liste.cpp -lfl -ly -lc

exemple_liste: exemple_liste.tex
              latex exemple_liste
              dvipdf exemple_liste

clean:
      rm -fr liste.cpp liste
      rm -fr lex.yy.c
      rm -fr *.aux *.log *.dvi
```

2 fichier liste.lex

```
%{
#include<stdlib.h>
%}
%%
[ \t\n] { ; }
[0-9]+ { yylval= atoi( ytext); return ENTIER; }
produit { return PRODUIT ; }
somme { return SOMME ; }
", " {return ','; }
"." {return FIN; }
<<EOF>> {return FIN; }
. {printf(" caractere inconnu: \n", ytext[0]); }
%%
```

3 fichier liste.yacc

```
%{
#include<stdio.h>
#include<stdlib.h>
```

```

int res; /* partout , on pourrait remplacer res par $$, ou une autre valeur $ */
extern "C"
{
    int yydebug;
    int yyparse();
    void yyerror(char*);
    int yylex();
    int yywrap() { return 1; }
};
%}
%token SOMME PRODUIT
%token ENTIER
%token FIN
%%
liste: FIN { printf("Adieu ,_monde_cruel\n"); return 0; }
|     SOMME ls FIN { printf("somme=%d\n", $2); } liste
|     PRODUIT lp FIN { printf("produit=%d\n", $2); } liste ;
ls:   ENTIER { $$=$1; } |   ls ',' ENTIER { $$= $1 + $3; };
lp:   ENTIER { $$=$1; } |   lp ',' ENTIER { $$= $1 * $3; };
%%
#include "lex.yy.c"
extern int yydebug;
/* int main() { yyparse(); } */
main()
{
    yydebug=1;
    yyparse();
}

/* autres regles possibles pour ls et lp :
ls: { $$=0;} | ls ',' ENTIER { $$= $1 + $3; };
lp: { $$=1;} | lp ',' ENTIER { $$= $1 * $3; };
elles admettent :
somme .
produit .
*/

```

4 Session

produit 1,2,3,4. somme 1,2,3,4..