

**Exercice 1.**

- Compléter et tester le programme YACC suivant permettant d'évaluer une expression arithmétique ayant des nombres à un chiffre comme opérandes :

```
% {
#include <ctype.h>
% }
%token chiffre
%%
Ligne:  Expr '\n'      { printf("%d \n", $1); }
      ;
Expr :  Expr '+' Terme  { $$ = $1 + $3; }
      |  Terme
      ;
Terme :  Terme '*' Facteur { $$ = $1 * $3; }
      |  Facteur
      ;
Facteur :  '(' Expr ')'  { $$ = $2; }
      |  chiffre
      ;
%%
yylex()
{ int c;
  c = getchar();
  if (isdigit(c))
    { yylval = c - '0';
      return chiffre ;
    }
  return (c);
}
```

**Exercice 2.**

- Compléter votre programme YACC en y ajoutant les opérandes – (unaire) – (entre deux opérandes) et /. Prendre comme opérandes des nombres entiers ou réels quelconques.

**Exercice 3.**

- Utiliser maintenant la grammaire ambiguë et priorités et associativités pour effectuer le même traitement que l'exercice 2.