

PROBLÈME :

Considérer le langage de programmation Mini-C dont les productions de la grammaire dite syntaxique sont données ci-après. On désire produire un code intermédiaire sous forme de quadruplets après analyse des programmes écrits en langage Mini-C.

```

Function → Type identifier (ArgList ) CompoundStmt
ArgList → Arg | ArgList , Arg
Arg → Type identifier
Declaration → Type IdentList ;
Type → int | float
IdentList → identifier , IdentList | identifier
Stmt → ForStmt
      | WhileStmt
      | Expr ;
      | IfStmt
      | CompoundStmt
      | Declaration
      | ;
ForStmt → for ( Expr ; OptExpr ; OptExpr ) Stmt
OptExpr → Expr | ε
WhileStmt → while ( Expr ) Stmt
IfStmt → if ( Expr ) Stmt ElsePart
ElsePart → else Stmt | ε
CompoundStmt → { StmtList }
StmtList → StmtList Stmt | ε
Expr → identifier = Expr | Rvalue
Rvalue → Rvalue Compare Mag | Mag
Compare → == | < | > | <= | >= | !=
Mag → Mag + Term | Mag . Term | Term
Term → Term * Factor | Term / Factor | Factor
Factor → ( Expr ) | - Factor | + Factor | identifier | number

```

Travail demandé :

Concevoir et implémenter qui produit

- Un analyseur lexico-syntaxique du langage Mini-C
- Producteur du code intermédiaire sous forme de quadruplets

ÉTAPE :

N°	ACTIVITÉ	N°	Actions
1	Conception	1	Conception des analyseurs lexical et syntaxique LL (1)
		2	Conception de l'analyseur sémantique qui génère les quadruplets du programme
2	Implémentation	1	Utilisation de l'outil de génération des analyseurs JAVACC
		2	Programmer en de Java toutes les actions de l'analyseur sémantique