

**BNF Grammar****LEGEND:**

- **Bold Words** = non-terminals
- "Words In Quotes" = terminals

// PROGRAM

**Program** → **FunctionList**|  $\epsilon$ **FunctionList** → **FunctionList** **Function**| **Function**

// FUNCTION

**Function** → "function" "identifier" " ," **FunctionParams** **FunctionLocals** **FunctionBody****FunctionParams** → "beginparams" **DeclarationList** "endparams"

| "beginparams" "endparams"

**FunctionLocals** → "beginlocals" **DeclarationList** "endlocals"

| "beginlocals" "endlocals"

**FunctionBody** → "beginbody" **StatementList** "endbody"

| "beginbody" "endbody"

**DeclarationList** → **DeclarationList** **Declaration** " ,"| **Declaration** " ,"

// DECLARATION

**Declaration** → **IdentifierList** " ." "integer"| **IdentifierList** " ." "array" "[" "number" "]" "of" "integer"**IdentifierList** → "identifier"| **IdentifierList** " ," "identifier"

// STATEMENT

**Statement** → **Var** “:=” **Expression**

- | “if” **BoolExpr** “then” **StatementList** “endif”
- | “if” **BoolExpr** “then” **StatementList** “else” **StatementList** “endif”
- | “while” **BoolExpr** “beginloop” **StatementList** “endloop”
- | “do” “beginloop” **StatementList** “endloop” “while” **BoolExpr**
- | “for” **Var** “:=” “number” “,” **BoolExpr** “,” **Var** “:=” **Expression** “beginloop”  
    **StatementList** “endloop”
- | “read” **VarList**
- | “write” **VarList**
- | “continue”
- | “return” **Expression**

**StatementList** → **Statement**

- | **StatementList** **Statement** “,”

// BOOLEAN EXPRESSION

**BoolExpr** → **BoolExpr** “or” **RelationAndExpr**

- | **RelationAndExpr**

// RELATION AND EXPRESSION

**RelationAndExpr** → **RelationAndExpr** “and” **RelationExpr**

- | **RelationExpr**

// RELATION EXPRESSION

**RelationExpr** → **Relations**

- | “not” **Relations**

**Relations** → **Expression** **Comp** **Expression**

- | “true”
- | “false”
- | “(“ **BoolExpr** “)”

// COMP

**Comp** → “==”  
| “<”  
| “<”  
| “>”  
| “<=”  
| “>=”

// EXPRESSIONS

**Expression** → **Expression** “+” **MultiplicativeExpr**  
| **Expression** “-” **MultiplicativeExpr**  
| **MultiplicativeExpr**

**ExpressionList** → **ExpressionList** **Expression** “,”  
| **Expression**  
| ε

// MULTIPLICATION EXPRESSION

**MultiplicativeExpr** → **MultiplicativeExpr** “\*” **Term**  
| **MultiplicativeExpr** “/” **Term**  
| **MultiplicativeExpr** “%” **Term**  
| **Term**

// TERMS

**Term** → “-” **TermInner**  
| “identifier” “(” **ExpressionList** “)”

**TermInner** → **Var**  
| “number”  
| “(” **Expression** “)”

// VARIABLES

**Var** → "identifier"  
| "identifier" "[" **Expression** "]"

**VarList** → **Var**  
| **VarList Var** ","