Funw@p Grammar

The grammar is not completely been transformed into a LL1 grammar in order to maintain a certain legibility of it, and of the Top-Down Parsing as well.
Therefore, we are going to manage these cases with the lookahead strategy.

Program	→	Main | DeclList Main
Main	→	**func Main ( )** Block
DeclList	→	Decl DeclList | **ε**
Decl	→	**var** VarDeclList Type **;** | **func IDE** FunDecl
TypeList	→	Type**,** TypeList | Type
RType	→	Type | **ε**
Type	→	**int** | **bool** | **char** | **url** | **string** | **fun (** TypeList **)** RType
VarDeclList	→	VarDecl**,** VarDeclList | VarDecl
VarDecl	→	**IDE** | **IDE =** Exp
FunDecl	→	**(** Params **)** RType Block
Params	→	ParamList | **ε**
ParamList	→	**IDE** Type **,** ParamList | **IDE** Type
Block	→	**{** DeclList StmtList **}**
StmtList	→	Stmt StmtList | **ε**
Stmt	→	**IDE** StmtIDE | **IDE = readIn ( ) ;** | Call **;** | **IDE = async (** Exp **) ;** | **IDE = dasync (IDE ,** Call **) ;** | **if (** Exp **)** Block ElseStmt | **while (** Exp **)** Block | **for (IDE =** Exp**;** Exp **; IDE** StmtIDE **)** Block | **return** Exp **;** | **return func** FunDecl **;** | **println (** PrintList **);**
StmtIDE	→	**=** Exp **;** | **+=** Exp **;** | **-=** Exp **;** | **++ ;** | **-- ;**
PrintList	→	Exp**,** PrintList | Exp
ElseStmt	→	**else** Block | **ε**

Exp	→	AndExp MoreAndExps
MoreAndExps	→	**||** AndExp **|** **ε**
AndExp	→	UnaryRelExp MoreUnaryRelExps
MoreUnary-RelExps	→	**&&** UnaryRelExp **|** **ε**
UnaryRelExp	→	**!** UnaryRelExp **|** RelExp
RelExp	→	SumExp MoreSumExps
MoreSumExps	→	RelOp SumExp **|** **ε**
RelOp	→	**<=** **|** **<** **|** **>** **|** **>=** **|** **==** **|** **!=**
SumExp	→	Term MoreTerms
MoreTerms	→	**+** Term **|** **-** Term **|** **ε**
Term	→	UnaryExp MoreUnaryExps
MoreUnaryExps	→	***** UnaryExp **|** Term **/** UnaryExp **|** **ε**
UnaryExp	→	**-** UnaryExp **|** Factor
Factor	→	**IDE** **|** **(** Exp **)** **|** Call **|** Const
Call	→	**IDE** **(** Args **)**
Args	→	ArgList **|** **ε**
ArgList	→	Exp **,** ArgList **|** Exp
Const	→	**NUMBERC** **|** **CHARC** **|** **STRINGC** **|** **URLC** **|** **true** **|** **false**