

Gramática LL(1)

Ponto de Início de Execução:

```
S = DeclId S
S = FunDecl S
S = ProcDecl S
S = &
```

Declaração de Variáveis:

```
DeclId = Type LId
DeclId = 'const' Type LId
LId = 'id' IdAttr LIdr ';'
LIdr = ',' 'id' IdAttr LIdr
LIdr = &
Type = 'int' | 'float' | 'bool' | 'char' | 'string'
```

Id:

```
Id = 'id' IdOpt
IdOpt = ArrayOpt
IdOpt = FunCall
IdAttr = ArrayOpt AttrOpt
IdAttr = FunCall
```

Array:

```
ArrayOpt = '[' Ea ']'
ArrayOpt = &
```

Atribuição:

```
AttrOpt = 'opAttrib' Ec
AttrOpt = &
```

Declaração de Funções:

```
FunDecl = 'funDef' Type FunName Param Body
FunName = 'id' | 'main'
Param = '(' LParam ')'
LParam = Type 'id' ArrayOpt LParamr
LParam = &
LParamr = ',' Type 'id' ArrayOpt LParamr
LParamr = &
LParamCall = Ec LParamCallr
```

```

LParamCallr = ',' Ec LParamCallr
LParamCallr = &
FunCall = '(' LParamCall ')'
Return = Ec
Return = &

```

Declaração de Procedimento:

```

ProcDecl = 'procDef' 'id' Param Body

```

Corpo da Função / Procedimento:

```

Body = '{' BodyPart '}'
BodyPart = DeclId BodyPart
BodyPart = LId BodyPart
BodyPart = Command BodyPart
BodyPart = 'funRet' Return ';'
BodyPart = &

```

Comandos:

```

PrintParam = ',' LParamCall
PrintParam = &
Command = 'print' '(' Ec PrintParam ')' ';'
Command = 'scan' '(' LParamCall ')' ';'
Command = 'whileLoop' '(' Eb ')' Body
Command = 'forLoop' '(' 'typeInt' 'id' ':' '(' Ec ',' Ec ','
Ec ')' ')' Body
Command = 'condIf' '(' Eb ')' Body Ifr
Ifr = 'condElse' Body
Ifr = 'condElseIf' '(' Eb ')' Body Ifr
Ifr = &

```

Expressões:

```

Ec = Fc Ecr
Ecr = 'opConcat' Fc Ecr
Ecr = &
Fc = 'constStr'
Fc = 'constChar'
Fc = Eb
Eb = Tb Ebr
Ebr = 'opOr' Tb Ebr
Ebr = &
Tb = Fb Tbr

```

```
Tbr = 'opAnd' Fb Tbr
Tbr = &
Fb = 'opNot' Fb
Fb = 'cteBool'
Fb = Ra Fbr
Fbr = 'opGreater' Ra Fbr
Fbr = 'opLesser' Ra Fbr
Fbr = 'opGreq' Ra Fbr
Fbr = 'opLeq' Ra Fbr
Fbr = &
Ra = Ea Rar
Rar = 'opEquals' Ea Rar
Rar = 'opNotEqual' Ea Rar
Rar = &
Ea = Ta Ear
Ear = 'opAdd' Ta Ear
Ear = 'opSub' Ta Ear
Ear = &
Ta = Pa Tar
Tar = 'opMult' Pa Tar
Tar = 'opDiv' Pa Tar
Tar = &
Pa = Fa Par
Par = 'opPow' Fa Par
Par = &
Fa = '(' Ec ') '
Fa = 'opSub' Fc
Fa = Id | 'cteInt' | 'cteFloat'
```