

# LL(1)

---

```

S          = 'typeInt' SR
           | NotIntType Decl S
           | 'typeVoid' 'identifier' DeclFun S
           | EPSILON

SR         = 'fnDecl' SAux
           | TypeArrOpt 'fnDecl' 'identifier' DeclFun S
           | 'identifier' DeclVar S

SAux       = 'main' Main
           | 'identifier' DeclFun S

Main      = '(' ')' Body

Body      = '{' Content '}'

Content   = Command Content
           | VarType 'identifier' DeclVar Content
           | 'identifier' Content
           | 'fnRtn' Rtn ';'
           | EPSILON

Content'  = '(' ParamsCall ')' ';' Content
           | LAttr ';' Content

Command   = 'cmdIf' '(' Eb ')' Body LElsif CmdElse
           | 'cmdWhile' '(' Eb ')' Body
           | 'cmdFor' 'identifier' 'in' '(' Ea ',' Ea ',' Ea ')' Body
           | 'fnRead' '(' LIdentifier ')' ';'
           | 'fnWrite' '(' 'stringVal' PrintParams ')' ';'

LElsif    = 'cmdElsif' '(' Eb ')' Body LElsif
           | EPSILON

CmdElse   = 'cmdElse' Body
           | EPSILON

LIdentifier = 'identifier' LIdentifier
           | EPSILON

LIdentifier' = ',' 'identifier' LIdentifier
           | EPSILON

PrintParams = ',' Eb PrintParams
           | EPSILON

VarType    = 'typeInt'
           | 'typeFloat'

```

```

| 'typeString'
| 'typeChar'
| 'typeBool'

DeclVar      = LVar ';'

LVar         = Var' LVar'
              | EPSILON

LVar'        = ',' Var LVar'
              | EPSILON

Var          = 'identifier' Var'

Var'         = '=' Ec
              | ArrOpt

ParamsCall   = Ec ParamsCall'
              | EPSILON

ParamsCall'  = ',' Ec ParamsCall'
              | EPSILON

LAttr        = '=' Ec LAttr'
              | '[' Ea ']' '=' Ec LAttr'

LAttr'       = ',' LAttr''
              | EPSILON

LAttr''      = 'identifier' '=' Ec LAttr'
              | '[' Ea ']' '=' Ec LAttr'

Rtn          = Ec
              | EPSILON

DeclFun      = '(' Params ')' Body

Params       = VarType 'identifier' ArrOpt Paramsx
              | EPSILON

Paramsx      = ',' VarType 'identifier' ArrOpt Paramsx
              | EPSILON

ArrOpt       = '[' ArrSize ']'
              | EPSILON

ArrSize      = 'identifier'
              | 'intVal'

TypeArrOpt   = '[' ']'
              | EPSILON

NotIntType   = 'typeFloat'
              | 'typeString'

```

```

      | 'typeChar'
      | 'typeBool'

Decl
  = TypeArrOpt Declx

Declx
  = 'fnDecl' 'identifier' DeclFun
  | 'identifier' DeclVar

Ec
  = Eb Ec'

Ec'
  = 'opConcat' Eb Ec'
  | EPSILON

Eb
  = Tb Eb'

Eb'
  = 'opOr' Tb Eb'
  | EPSILON

Tb
  = Fb Tb'

Tb'
  = 'opAnd' Fb Tb'
  | EPSILON

Fb
  = 'opNot' Fb
  | Ra Fb'

Fb'
  = 'opGtrThan' Fb Fb'
  | 'opLessThan' Fb Fb'
  | 'opGtrEqual' Fb Fb'
  | 'opLessEq' Fb Fb'
  | EPSILON

Ra
  = Ea Ra'

Ra'
  = 'opEqual' Ea Ra'
  | 'opNotEqual' Ea Ra'
  | EPSILON

Ea
  = Ta Ea'

Ea'
  = 'opAdd' Ta Ea'
  | 'opSub' Ta Ea'
  | EPSILON

Ta
  = Fa Ta'

Ta'
  = 'opMult' Fa Ta'
  | 'opDiv' Fa Ta'
  | 'opMod' Fa Ta'
  | EPSILON

Fa
  = '(' Ec ')'
  | 'opSub' Fa
  | VarOrFn

```

```

| 'intVal'
| 'floatVal'
| 'stringVal'
| 'charVal'
| 'boolVal'

VarOrFn      = 'identifier' VarOrFn'

VarOrFn'     = '[' Ea ']'
              | '(' ParamsCall ')'
              | EPSILON

```

## GLC

---

```

S              = DeclFn S
              | DeclVar S
              | 'int' Main
              | EPSILON

DeclFn         = FnType DeclArrOpt 'fnDecl' 'identifier' Params Body

FnType         = VarType
              | 'typeVoid'

VarType        = 'typeBool'
              | 'typeInt'
              | 'typeFloat'
              | 'typeString'
              | 'typeChar'

DeclArrOpt     = '[' ArrSizeOpt ']'
              | EPSILON

ArrSizeOpt     = ArrSize
              | EPSILON

ArrSize        = 'identifier'
              | 'intVal'

Params         = '(' Param ')'

Params         = Param ',' VarType 'identifier'
              | Param ',' VarType 'identifier' DeclArr
              | VarType 'identifier'
              | VarType 'identifier' DeclArr
              | EPSILON

DeclArr        = '[' ArrSize ']'

```

```

| EPSILON

Body      = '{' Content '}'

Content   = Command Content
          | DeclVar Content
          | 'identifier' '(' ParamsCall ')' ';' Content
          | 'fnRtn' Rtn ';'
          | LAttr ';' Content
          | EPSILON

Command   = 'cmdIf' '(' Eb ')' Body LElsif
          | 'cmdIf' '(' Eb ')' Body LElsif 'cmdElse' Body
          | 'cmdWhile' '(' Eb ')' Body
          | 'cmdFor' 'identifier' '=' '(' Ea ',' Ea ',' Ea ')'

Body      | 'fnRead' '(' LIdentifier ')' ';'
          | 'fnWrite' '(' 'stringVal' PrintParams ')' ';'

LElsif     = LElsif 'cmdElsif' Body
          | EPSILON

LIdentifier = LIdentifier ',' 'identifier'
          | 'identifier'

PrintParams = ',' Eb PrintParams
          | EPSILON

DeclVar    = VarType LVar ';'

LVar       = LVar ',' Var
          | Var

Var        = 'identifier' '=' Ec
          | 'identifier'
          | 'identifier' DeclArr

ParamsCall = ParamsCall ',' Ec
          | Ec
          | EPSILON

Rtn        = Ec
          | EPSILON

LAttr      = LAttr ',' 'identifier' '=' Ec
          | LAttr ',' '[' Ea ']' '=' Ec
          | 'identifier' '=' Ec
          | 'identifier' '[' Ea ']' '=' Ec

Main       = 'fnDecl' 'main' Params Body

Ec         = Ec 'opConcat' Eb
          | Eb

```

```

Eb          = Eb 'opOr' Tb
             | Tb

Tb          = Tb 'opAnd' Fb
             | Fb

Fb          = Fb OpRel Ra
             | 'opNot' Fb
             | Ra

Ra          = Ra 'opEqual' Ea
             | Ra 'opNotEqual' Ea
             | Ea

Ea          = Ea 'opAdd' Ta
             | Ea 'opSub' Ta
             | Ta

Ta          = Ta 'opMult' Fa
             | Ta 'opDiv' Fa
             | Ta 'opMod' Fa
             | Fa

Fa          = '(' Ec ')'
             | 'opUnaryNeg' Fa
             | VarOrFn
             | 'intVal'
             | 'floatVal'
             | 'stringVal'
             | 'charVal'
             | 'boolVal'

OpRel       = 'opGtrThan'
             | 'opLessThan'
             | 'opGtrEqual'
             | 'opLessEq'

VarOrFn     = 'identifier' '[' Ea ']'
             | 'identifier' Params
             | 'identifier'

```