

```

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'
```