Program → **{** Global **}** 'int' 'main' '(' ')' Statements

Global → FunctionDeclaration **|** Declarations

FunctionDeclaration → Type Id '(' ParamDeclarations ’)’ Statements

ParamDeclarations → ParamDeclaration **{** ',' ParamDeclaration **} |** ε

ParamDeclaration → Type Id **[** ‘[‘ ‘]’ **]**

Statements → ‘{‘ **{** Declarations **}** **{** Statement **}** ‘}’

Declarations → Type Init **{** ',' Init **}** ';'

Init → ArrayInit **|** NoArrayInit

ArrayInit → Id '[' Integer ']' **[** '=' '{' Expression **{** ',' Expression **}** '}' **]**

NoArrayInit → Id **[** '=' Expression **]**

Statement → Skip **|** IfStatement **|** Block **|** WhileStatement **|** SwitchStatement **|** ForStatement **|** Return **|** Expression **|** Break **|** Continue

Block → '{' **{** Statement **}** '}'

IfStatement → 'if' '(' Expression ')' Block **{** 'else' 'if' '(' Expression ')' Block **}** **[** 'else' Block **]**

WhileStatement → 'while' '(' Expression ')' Block

SwitchStatement → 'switch' '(' Expression ')' '{' **{** 'case' Literal ':' **{** Statement **}** **}** **[** 'default' ':' **{** Statement **}** **]** '}'

ForStatement → 'for' '(' InnerForStatement ';' Expression ';' InnerForStatement ')' Block

InnerForStatement → Expression **{** ',' Expression **} |** ε

Return → 'return' **[** Expression **]** ';'

Function → Id ‘(‘ Params ‘)’  
Params → Expression **{** ‘,’ Expression **} |** ε

Break → 'break' ';'

Continue → 'continue' ';'

Skip → ';'

Expression → **[** Disjunction AssignmentOp **]** Disjunction

AssignmentOp → ‘+=’ **|** ‘-=’ **| ‘**\*=’ **|** ‘/=’ **| ‘**%=’ **|** ‘=’

Disjunction → Conjunction **{** ‘||’ Conjunction **}**

Conjunction → Equality **{** ‘&&’ Equality **}**

Equality → Relation **[** EquOp Relation **]z**

EquOp → ‘==’ **|** ‘!=’

Relation → Addition **[** RelOp Addition **]**

RelOp → ‘<’ **| ‘**<=’ **|** ‘>’ **|** ‘>=’

Addition → Term **{** AddOp Term }

AddOp → ‘+’ **|** ‘-‘

Term → Double **{** MulOp Double **}**

MulOp → ‘\*’ **|** ‘/’ **|** ‘%’

Double → Factor **[** DouOp **]**

DouOp → ‘++’ **|** ‘—‘

Factor→ **[** UnaryOp **]** Primary

UnaryOp → ‘-‘ **|** ‘!’

Primary → Id **[** ‘[’ Expression ’]’ **] |** Literal **|** ‘(‘ Expression ‘)’ **|** Function **|** ‘(‘ Type ’)’ Expression

Type → 'int' **|** 'float' **|** 'char' **|** 'bool' **|** 'time' **|** 'date' **|** ‘void’

Id → Letter **{** Letter **|** Digit **}**

Letter → a **|** b **|** .. **|** z **|** A **|** B **|** .. **|** Z

Digit → 0 **|** 1 **|** .. **|** 9

Literal → Integer **|** Boolean **|** Float **|** Char **|** Date **|** Time

Integer → Digit **{** Digit **}**

Boolean → True **|** False

Float → Integer ‘.’ Integer

Char → ‘ ASCII Char ‘

Time → Integer ‘:’ Integer ‘:’ Integer

Date → Integer ‘.’ Integer ‘.’ Integer