BNF

```
start -> program id program_begin stmt* program_end

stmt -> ifStmt | whileStmt | assignStmt | defineStmt | atomicStmt

ifStmt -> ifStmtTmp | ifStmtTmp else begin stmt* end
ifStmtTmp -> ifStmtTmp elseif ( exp ) begin stmt* end | if ( exp ) begin stmt* end

whileStmt -> while ( exp ) begin stmt* end

assignStmt -> id = exp ;

defineStmt -> type defineStmtTmp ;
defineStmtTmp -> defineStmtTmp , id | defineStmtTmp , id = exp | id | id = exp
type -> integer

atomicStmt -> continue; | break; | display( ); | display ( stringLiteral ) ;

exp -> exp compop simple_exp | simple_exp
simple_exp -> simple_exp addop term | term
term -> term mulop factor | factor
factor -> ( exp ) | number | id

compop -> < | > | <= | >= | ==
addop -> + | -      mulop -> * | /
```

EBNF

```
start -> program id program_begin {stmt} program_end

stmt -> ifStmt | whileStmt | assignStmt | defineStmt | atomicStmt

ifStmt -> if ( exp ) begin {stmt} end { elseif ( exp ) begin {stmt} end } [ else begin {stmt} end]

whileStmt -> while ( exp ) begin {stmt} end

assignStmt -> id = exp ;

defineStmt -> type id [ = exp ] { , id [ = exp ] } ;
type -> integer

atomicStmt -> continue; | break; | display( [ stingLiteral ] );

exp -> simple_exp { compop simple_exp }
simple_exp -> term { addop term }
term -> factor { mulop factor }
factor -> ( exp ) | number | id

compop -> < | > | <= | >= | ==
addop -> + | -
mulop -> * | /
```