**CFG**

S -> CODE

CODE -> STMT CODE

CODE -> FDECL CODE

CODE -> ''

VDECL -> vtype id semi

VDECL -> vtype ASSIGN semi

ASSIGN -> id assign RHS

RHS -> EXPR

RHS -> character

RHS -> string

RHS -> boolstr

EXPR -> CALL

EXPR -> TERM addsub EXPR

EXPR -> TERM

TERM -> FACT multdiv TERM

TERM -> FACT

FACT -> num

FACT -> lparen EXPR rparen

FACT -> id

FDECL -> vtype id lparen ARG rparen lbrace BLOCK RETURN rbrace

ARG -> vtype id MOREARGS

ARG -> ''

MOREARGS -> comma vtype id MOREARGS

MOREARGS -> ''

BLOCK -> STMT BLOCK

BLOCK -> ''

STMT -> VDECL

STMT -> ASSIGN semi

STMT -> CALL semi

STMT -> if lparen COND rparen lbrace BLOCK rbrace ELSE

STMT -> while lparen COND rparen lbrace BLOCK rbrace

CALL -> id lparen ARGVAL rparen

ARGVAL -> EXPR MOREARGVAL

ARGVAL -> ''

MOREARGVAL -> comma EXPR MOREARGVAL

MOREARGVAL -> ''

COND -> COND\_EXPR LOGIC\_EXPR

COND\_EXPR -> boolstr

COND\_EXPR -> FACT comp FACT

LOGIC\_EXPR -> ''

LOGIC\_EXPR -> LOGIC\_OP COND

LOGIC\_OP -> and

LOGIC\_OP -> or

ELSE -> else lbrace BLOCK rbrace

ELSE -> ''

RETURN -> return RHS semi

S -> CODE

CODE -> STMT CODE

CODE -> FDECL CODE

CODE -> ''

VDECL -> vtype id semi

VDECL -> vtype ASSIGN semi

ASSIGN -> id assign RHS

RHS -> EXPR

RHS -> character

RHS -> string

RHS -> boolstr

EXPR -> CALL

EXPR -> TERM addsub EXPR

EXPR -> TERM

TERM -> FACT multdiv TERM

TERM -> FACT

FACT -> num

FACT -> lparen EXPR rparen

FACT -> id

FDECL -> vtype id lparen ARG rparen lbrace BLOCK RETURN rbrace

ARG -> vtype id MOREARGS

ARG -> ''

MOREARGS -> comma vtype id MOREARGS

MOREARGS -> ''

BLOCK -> STMT BLOCK

BLOCK -> ''

STMT -> VDECL

STMT -> ASSIGN semi

STMT -> CALL semi

STMT -> if lparen COND rparen lbrace BLOCK rbrace ELSE

STMT -> while lparen COND rparen lbrace BLOCK rbrace

CALL -> id lparen ARGVAL rparen

ARGVAL -> EXPR MOREARGVAL

ARGVAL -> ''

MOREARGVAL -> comma EXPR MOREARGVAL

MOREARGVAL -> ''

COND -> COND\_EXPR LOGIC\_EXPR

COND\_EXPR -> boolstr

COND\_EXPR -> FACT comp FACT

LOGIC\_EXPR -> ''

LOGIC\_EXPR -> LOGIC\_OP COND

LOGIC\_OP -> and

LOGIC\_OP -> or

ELSE -> else lbrace BLOCK rbrace

ELSE -> ''

RETURN -> return RHS semi

**Terminals**

1. **vtype** for the types of variables and functions
2. **num** for signed integers
3. **character** for a single character
4. **boolstr** for Boolean strings
5. **string** for literal strings
6. **id** for the identifiers of variables and functions
7. **if**, **else**, **while**, and **return** for if, else, while, and return statements respectively
8. **addsub** for +, and -
9. **multidiv** for \*, and /
10. **assign** for assignment operators
11. **comp** for comparison operators
12. **semi** and **comma** for semicolons and commas respectively
13. **lparen, rparen, lbrace**, and **rbrace** for (, ), {, and } respectively
14. **and, and or for & and |**

**Non-terminals (15)**

['EXPR', 'VDECL', 'LOGIC\_EXPR', 'CODE', 'CALL', 'ELSE', 'ARGVAL', 'MOREARGVAL', 'BLOCK', 'ASSIGN', 'STMT', 'FACT', 'COND', 'MOREARGS', 'RHS', 'S', 'RETURN', 'COND\_EXPR', 'TERM', 'FDECL', 'ARG', 'LOGIC\_OP']

**Start symbol: S**