<CompUnit> → [<ConstDecl> | <VarDecl> | <FuncDef>]<MainFuncDef>

<ConstDecl> → ‘const’ <BType> <ConstDef> {‘,’ <ConstDef>};

<ConstDef> → <Ident> { '[' <ConstExp> ']' } '=' <ConstInitVal>

<ConstInitVal> → <ConstExp> |

'{' [ <ConstInitVal> { ',' <ConstInitVal> } ] '}'

<ConstExp> → <AddExp>

<VarDecl> → <BType> <VarDef> { ',' <VarDef> } ';'

<VarDef> → <Ident> { '[' <ConstExp> ']' } |

<Ident> { '[' <ConstExp> ']' } '=' <InitVal>

<InitVal> → <Exp> |

'{' [ <InitVal> { ',' InitVal } ] '}'

<FuncDef> → 'func' <Ident> '(' [FuncFParams] ')' <FuncType> <Block>

<FuncFParams> → <FuncFParam> { ',' <FuncFParam> }

<FuncFParam> → <Ident> <FuncFVarParam> |

<Ident> <FuncFArrayParam> |

<Ident> <FuncFFuncParam>

<FuncFVarParam> → <BType>

<FuncFArrayParam> → '[' ']' { '[' ConstExp ']' } <BType>

<FuncType> → [<BType>]

<Block> → '{' { <BlockItem> } '}'

<BlockItem> → <ConstDecl> |

<VarDecl> |

<Stmt>

<Stmt> → <AssignStmt> |

[Exp] ';' |

<Block> |

<IfStmt> |

<WhileStmt> |

<BreakStmt> |

<ContinueStmt> |

<ReturnStmt> |

<PrintfStmt> |

<ScanfStmt>

<IfStmt> → 'if' '(' <Cond> ')' <Stmt> [ 'else' <Stmt> ]

<WhileStmt> → 'while' '(' <Cond> ')' <Stmt>

<BreakStmt> → 'break' ';'

<ContinueStmt> → 'continue' ';'

<ReturnStmt> → [Exp] ';'

<AssignStmt> → <LVal> '=' <Exp> ';'

<Exp> → <AddExp>

<AddExp> → <MulExp> |

<AddExp> ( <AddOp> ) <MulExp>

<MulExp> → <UnaryExp> |

<MulExp> ( <MulOp> ) <UnaryExp>

<UnaryExp> → <PrimaryExp> |

<Ident> ['(' [<FuncRParams>] ')'] |

<UnaryOp> <UnaryExp>

<PrimaryExp> → '(' <Exp> ')' |

<LVal> |

<Number>

<LVal> → <Ident> {'[' <Exp> ']'}

<UnaryOp> → '+' |

'−' |

'!'

<AddOp> → '+' |

'−'

<MulOp> → '\*' |

'/' |

'%'

<BType> → ‘int’ |

‘char’