SLR grammar ('' is ε):

```
(0)  START -> CODE
(1)  CODE -> VDECL CODE
(2)  CODE -> FDECL CODE
(3)  CODE -> CDECL CODE
(4)  CODE -> ''
(5)  VDECL -> vtype id semi
(6)  VDECL -> vtype ASSIGN semi
(7)  ASSIGN -> id assign RHS
(8)  RHS -> EXPR
(9)  RHS -> literal
(10) RHS -> character
(11) RHS -> boolstr
(12) EXPR -> T addsub EXPR
(13) EXPR -> T
(14) T -> F multdiv T
(15) EXPR -> lparen EXPR rparen
(16) T -> F
(17) F -> id
(18) F -> num
(19) FDECL -> vtype id lparen ARG rparen lbrace BLOCK RETURN rbrace
(20) ARG -> vtype id MOREARGS
(21) ARG -> ''
(22) MOREARGS -> comma vtype id MOREARGS
(23) MOREARGS -> ''
(24) BLOCK -> STMT BLOCK
(25) BLOCK -> ''
(26) STMT -> VDECL
(27) STMT -> ASSIGN semi
(28) STMT -> if lparen COND rparen lbrace BLOCK rbrace ELSE
(29) STMT -> while lparen COND rparen lbrace BLOCK rbrace
(30) COND -> COND comp boolstr
(31) COND -> boolstr
(32) ELSE -> else lbrace BLOCK rbrace
(33) ELSE -> ''
(34) RETURN -> return RHS semi
(35) CDECL -> class id lbrace ODECL rbrace
(36) ODECL -> VDECL ODECL
(37) ODECL -> FDECL ODECL
(38) ODECL -> ''
```

FIRST / FOLLOW table

Nonterminal	FIRST	FOLLOW
START	{'',vtype,class}	{$}
CODE	{'',vtype,class}	{$}
VDECL	{vtype}	{$,vtype,class,return,id,if,while,rbrace}
ASSIGN	{id}	{semi}
RHS	{literal,character,boolstr,lparen,id,num}	{semi}
EXPR	{lparen,id,num}	{semi,rparen}
T	{id,num}	{addsub,semi,rparen}
F	{id,num}	{multdiv,addsub,semi,rparen}
FDECL	{vtype}	{$,vtype,class,rbrace}
ARG	{vtype,''}	{rparen}
MOREARGS	{comma,''}	{rparen}
BLOCK	{'',vtype,id,if,while}	{return,rbrace}
STMT	{vtype,id,if,while}	{return,vtype,id,if,while,rbrace}
COND	{boolstr}	{rparen,comp}
ELSE	{else,''}	{return,vtype,id,if,while,rbrace}
RETURN	{return}	{rbrace}
CDECL	{class}	{$,vtype,class}
ODECL	{vtype,''}	{rbrace}

>>

SLR closure table

Goto	Kernel	State	Closure
	{START -> .CODE}	0	{START -> .CODE; CODE -> .VDECL CODE; CODE -> .FDECL CODE; CODE -> .CDECL CODE; CODE -> .; VDECL -> .vtype id semi; VDECL -> .vtype ASSIGN semi; FDECL -> .vtype id lparen ARG rparen lbrace BLOCK RETURN rbrace; CDECL -> .class id lbrace ODECL rbrace}
goto(0, CODE)	{START -> CODE.}	1	{START -> CODE.}
goto(0, VDECL)	{CODE -> VDECL.CODE}	2	{CODE -> VDECL.CODE; CODE -> .VDECL CODE; CODE -> .FDECL CODE; CODE -> .CDECL CODE; CODE -> .; VDECL -> .vtype id semi; VDECL -> .vtype ASSIGN semi; FDECL -> .vtype id lparen ARG rparen lbrace BLOCK RETURN rbrace; CDECL -> .class id lbrace ODECL rbrace}
goto(0, FDECL)	{CODE -> FDECL.CODE}	3	{CODE -> FDECL.CODE; CODE -> .VDECL CODE; CODE -> .FDECL CODE; CODE -> .CDECL CODE; CODE -> .; VDECL -> .vtype id semi; VDECL -> .vtype ASSIGN semi; FDECL -> .vtype id lparen ARG rparen lbrace BLOCK RETURN rbrace; CDECL -> .class id lbrace ODECL rbrace}
goto(0, CDECL)	{CODE -> CDECL.CODE}	4	{CODE -> CDECL.CODE; CODE -> .VDECL CODE; CODE -> .FDECL CODE; CODE -> .CDECL CODE; CODE -> .; VDECL -> .vtype id semi; VDECL -> .vtype ASSIGN semi; FDECL -> .vtype id lparen ARG rparen lbrace BLOCK RETURN rbrace; CDECL -> .class id lbrace ODECL rbrace}
goto(0, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi; FDECL -> vtype.id lparen ARG rparen lbrace BLOCK RETURN rbrace}	5	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi; FDECL -> vtype.id lparen ARG rparen lbrace BLOCK RETURN rbrace; ASSIGN -> .id assign RHS}
goto(0, class)	{CDECL -> class.id lbrace ODECL rbrace}	6	{CDECL -> class.id lbrace ODECL rbrace}
goto(2, CODE)	{CODE -> VDECL CODE.}	7	{CODE -> VDECL CODE.}
goto(2, VDECL)	{CODE -> VDECL.CODE}	2	
goto(2, FDECL)	{CODE -> FDECL.CODE}	3	
goto(2, CDECL)	{CODE -> CDECL.CODE}	4	
goto(2, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi; FDECL -> vtype.id lparen ARG rparen lbrace BLOCK RETURN rbrace}	5	
goto(2, class)	{CDECL -> class.id lbrace ODECL rbrace}	6	
goto(3, CODE)	{CODE -> FDECL CODE.}	8	{CODE -> FDECL CODE.}
goto(3, VDECL)	{CODE -> VDECL.CODE}	2	
goto(3, FDECL)	{CODE -> FDECL.CODE}	3	
goto(3, CDECL)	{CODE -> CDECL.CODE}	4	
goto(3, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi; FDECL -> vtype.id lparen ARG rparen lbrace BLOCK RETURN rbrace}	5	
goto(3, class)	{CDECL -> class.id lbrace ODECL rbrace}	6	
goto(4, CODE)	{CODE -> CDECL CODE.}	9	{CODE -> CDECL CODE.}
goto(4, VDECL)	{CODE -> VDECL.CODE}	2	
goto(4, FDECL)	{CODE -> FDECL.CODE}	3	
goto(4, CDECL)	{CODE -> CDECL.CODE}	4	
goto(4, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi; FDECL -> vtype.id lparen ARG rparen lbrace BLOCK RETURN rbrace}	5	
goto(4, class)	{CDECL -> class.id lbrace ODECL rbrace}	6	
goto(5, id)	{VDECL -> vtype id.semi; FDECL -> vtype id.lparen ARG rparen lbrace BLOCK RETURN rbrace; ASSIGN -> id.assign RHS}	10	{VDECL -> vtype id.semi; FDECL -> vtype id.lparen ARG rparen lbrace BLOCK RETURN rbrace; ASSIGN -> id.assign RHS}
goto(5, ASSIGN)	{VDECL -> vtype ASSIGN.semi}	11	{VDECL -> vtype ASSIGN.semi}
goto(6, id)	{CDECL -> class id.lbrace ODECL rbrace}	12	{CDECL -> class id.lbrace ODECL rbrace}
goto(10, semi)	{VDECL -> vtype id semi.}	13	{VDECL -> vtype id semi.}
goto(10, lparen)	{FDECL -> vtype id lparen.ARG rparen lbrace BLOCK RETURN rbrace}	14	{FDECL -> vtype id lparen.ARG rparen lbrace BLOCK RETURN rbrace; ARG -> .vtype id MOREARGS; ARG -> .}
goto(10, assign)	{ASSIGN -> id assign.RHS}	15	{ASSIGN -> id assign.RHS; RHS -> .EXPR; RHS -> .literal; RHS -> .character; RHS -> .boolstr; EXPR -> .T addsub EXPR; EXPR -> .T; EXPR -> .lparen EXPR rparen; T -> .F multdiv T; T -> .F; F -> .id; F -> .num}
goto(11, semi)	{VDECL -> vtype ASSIGN semi.}	16	{VDECL -> vtype ASSIGN semi.}
goto(12, lbrace)	{CDECL -> class id lbrace.ODECL rbrace}	17	{CDECL -> class id lbrace.ODECL rbrace; ODECL -> .VDECL ODECL; ODECL -> .FDECL ODECL; ODECL -> .; VDECL -> .vtype id semi; VDECL -> .vtype ASSIGN semi; FDECL -> .vtype id lparen ARG rparen lbrace BLOCK RETURN rbrace}
goto(14, ARG)	{FDECL -> vtype id lparen ARG.rparen lbrace BLOCK RETURN rbrace}	18	{FDECL -> vtype id lparen ARG.rparen lbrace BLOCK RETURN rbrace}
goto(14, vtype)	{ARG -> vtype.id MOREARGS}	19	{ARG -> vtype.id MOREARGS}
goto(15, RHS)	{ASSIGN -> id assign RHS.}	20	{ASSIGN -> id assign RHS.}
goto(15, EXPR)	{RHS -> EXPR.}	21	{RHS -> EXPR.}
goto(15, literal)	{RHS -> literal.}	22	{RHS -> literal.}
goto(15, character)	{RHS -> character.}	23	{RHS -> character.}
goto(15, boolstr)	{RHS -> boolstr.}	24	{RHS -> boolstr.}
goto(15, T)	{EXPR -> T.addsub EXPR; EXPR -> T.}	25	{EXPR -> T.addsub EXPR; EXPR -> T.}
goto(15, lparen)	{EXPR -> lparen.EXPR rparen}	26	{EXPR -> lparen.EXPR rparen; EXPR -> .T addsub EXPR; EXPR -> .T; EXPR -> .lparen EXPR rparen; T -> .F multdiv T; T -> .F; F -> .id; F -> .num}
goto(15, F)	{T -> F.multdiv T; T -> F.}	27	{T -> F.multdiv T; T -> F.}
goto(15, id)	{F -> id.}	28	{F -> id.}
goto(15, num)	{F -> num.}	29	{F -> num.}
goto(17, ODECL)	{CDECL -> class id lbrace ODECL.rbrace}	30	{CDECL -> class id lbrace ODECL.rbrace}
goto(17, VDECL)	{ODECL -> VDECL.ODECL}	31	{ODECL -> VDECL.ODECL; ODECL -> .VDECL ODECL; ODECL -> .FDECL ODECL; ODECL -> .; VDECL -> .vtype id semi; VDECL -> .vtype ASSIGN semi; FDECL -> .vtype id lparen ARG rparen lbrace BLOCK RETURN rbrace}
goto(17, FDECL)	{ODECL -> FDECL.ODECL}	32	{ODECL -> FDECL.ODECL; ODECL -> .VDECL ODECL; ODECL -> .FDECL ODECL; ODECL -> .; VDECL -> .vtype id semi; VDECL -> .vtype ASSIGN semi; FDECL -> .vtype id lparen ARG rparen lbrace BLOCK RETURN rbrace}
goto(17, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi; FDECL -> vtype.id lparen ARG rparen lbrace BLOCK RETURN rbrace}	5	
goto(18, rparen)	{FDECL -> vtype id lparen ARG rparen.lbrace BLOCK RETURN rbrace}	33	{FDECL -> vtype id lparen ARG rparen.lbrace BLOCK RETURN rbrace}
goto(19, id)	{ARG -> vtype id.MOREARGS}	34	{ARG -> vtype id.MOREARGS; MOREARGS -> .comma vtype id MOREARGS; MOREARGS -> .}
goto(25, addsub)	{EXPR -> T addsub.EXPR}	35	{EXPR -> T addsub.EXPR; EXPR -> .T addsub EXPR; EXPR -> .T; EXPR -> .lparen EXPR rparen; T -> .F multdiv T; T -> .F; F -> .id; F -> .num}
goto(26, EXPR)	{EXPR -> lparen EXPR.rparen}	36	{EXPR -> lparen EXPR.rparen}
goto(26, T)	{EXPR -> T.addsub EXPR; EXPR -> T.}	25	
goto(26, lparen)	{EXPR -> lparen.EXPR rparen}	26	
goto(26, F)	{T -> F.multdiv T; T -> F.}	27	
goto(26, id)	{F -> id.}	28	
goto(26, num)	{F -> num.}	29	
goto(27, multdiv)	{T -> F multdiv.T}	37	{T -> F multdiv.T; T -> .F multdiv T; T -> .F; F -> .id; F -> .num}
goto(30, rbrace)	{CDECL -> class id lbrace ODECL rbrace.}	38	{CDECL -> class id lbrace ODECL rbrace.}
goto(31, ODECL)	{ODECL -> VDECL ODECL.}	39	{ODECL -> VDECL ODECL.}
goto(31, VDECL)	{ODECL -> VDECL.ODECL}	31	
goto(31, FDECL)	{ODECL -> FDECL.ODECL}	32	
goto(31, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi; FDECL -> vtype.id lparen ARG rparen lbrace BLOCK RETURN rbrace}	5	
goto(32, ODECL)	{ODECL -> FDECL ODECL.}	40	{ODECL -> FDECL ODECL.}
goto(32, VDECL)	{ODECL -> VDECL.ODECL}	31	
goto(32, FDECL)	{ODECL -> FDECL.ODECL}	32	
goto(32, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi; FDECL -> vtype.id lparen ARG rparen lbrace BLOCK RETURN rbrace}	5	
goto(33, lbrace)	{FDECL -> vtype id lparen ARG rparen lbrace.BLOCK RETURN rbrace}	41	{FDECL -> vtype id lparen ARG rparen lbrace.BLOCK RETURN rbrace; BLOCK -> .STMT BLOCK; BLOCK -> .; STMT -> .VDECL; STMT -> .ASSIGN semi; STMT -> .if lparen COND rparen lbrace BLOCK rbrace ELSE; STMT -> .while lparen COND rparen lbrace BLOCK rbrace; VDECL -> .vtype id semi; VDECL -> .vtype ASSIGN semi; ASSIGN -> .id assign RHS}
goto(34, MOREARGS)	{ARG -> vtype id MOREARGS.}	42	{ARG -> vtype id MOREARGS.}
goto(34, comma)	{MOREARGS -> comma.vtype id MOREARGS}	43	{MOREARGS -> comma.vtype id MOREARGS}
goto(35, EXPR)	{EXPR -> T addsub EXPR.}	44	{EXPR -> T addsub EXPR.}
goto(35, T)	{EXPR -> T.addsub EXPR; EXPR -> T.}	25	
goto(35, lparen)	{EXPR -> lparen.EXPR rparen}	26	
goto(35, F)	{T -> F.multdiv T; T -> F.}	27	
goto(35, id)	{F -> id.}	28	
goto(35, num)	{F -> num.}	29	
goto(36, rparen)	{EXPR -> lparen EXPR rparen.}	45	{EXPR -> lparen EXPR rparen.}
goto(37, T)	{T -> F multdiv T.}	46	{T -> F multdiv T.}
goto(37, F)	{T -> F.multdiv T; T -> F.}	27	
goto(37, id)	{F -> id.}	28	
goto(37, num)	{F -> num.}	29	
goto(41, BLOCK)	{FDECL -> vtype id lparen ARG rparen lbrace BLOCK.RETURN rbrace}	47	{FDECL -> vtype id lparen ARG rparen lbrace BLOCK.RETURN rbrace; RETURN -> .return RHS semi}
goto(41, STMT)	{BLOCK -> STMT.BLOCK}	48	{BLOCK -> STMT.BLOCK; BLOCK -> .STMT BLOCK; BLOCK -> .; STMT -> .VDECL; STMT -> .ASSIGN semi; STMT -> .if lparen COND rparen lbrace BLOCK rbrace ELSE; STMT -> .while lparen COND rparen lbrace BLOCK rbrace; VDECL -> .vtype id semi; VDECL -> .vtype ASSIGN semi; ASSIGN -> .id assign RHS}
goto(41, VDECL)	{STMT -> VDECL.}	49	{STMT -> VDECL.}
goto(41, ASSIGN)	{STMT -> ASSIGN.semi}	50	{STMT -> ASSIGN.semi}
goto(41, if)	{STMT -> if.lparen COND rparen lbrace BLOCK rbrace ELSE}	51	{STMT -> if.lparen COND rparen lbrace BLOCK rbrace ELSE}
goto(41, while)	{STMT -> while.lparen COND rparen lbrace BLOCK rbrace}	52	{STMT -> while.lparen COND rparen lbrace BLOCK rbrace}
goto(41, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi}	53	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi; ASSIGN -> .id assign RHS}
goto(41, id)	{ASSIGN -> id.assign RHS}	54	{ASSIGN -> id.assign RHS}
goto(43, vtype)	{MOREARGS -> comma vtype.id MOREARGS}	55	{MOREARGS -> comma vtype.id MOREARGS}
goto(47, RETURN)	{FDECL -> vtype id lparen ARG rparen lbrace BLOCK RETURN.rbrace}	56	{FDECL -> vtype id lparen ARG rparen lbrace BLOCK RETURN.rbrace}
goto(47, return)	{RETURN -> return.RHS semi}	57	{RETURN -> return.RHS semi; RHS -> .EXPR; RHS -> .literal; RHS -> .character; RHS -> .boolstr; EXPR -> .T addsub EXPR; EXPR -> .T; EXPR -> .lparen EXPR rparen; T -> .F multdiv T; T -> .F; F -> .id; F -> .num}
goto(48, BLOCK)	{BLOCK -> STMT BLOCK.}	58	{BLOCK -> STMT BLOCK.}
goto(48, STMT)	{BLOCK -> STMT.BLOCK}	48	
goto(48, VDECL)	{STMT -> VDECL.}	49	
goto(48, ASSIGN)	{STMT -> ASSIGN.semi}	50	
goto(48, if)	{STMT -> if.lparen COND rparen lbrace BLOCK rbrace ELSE}	51	
goto(48, while)	{STMT -> while.lparen COND rparen lbrace BLOCK rbrace}	52	
goto(48, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi}	53	
goto(48, id)	{ASSIGN -> id.assign RHS}	54	
goto(50, semi)	{STMT -> ASSIGN semi.}	59	{STMT -> ASSIGN semi.}
goto(51, lparen)	{STMT -> if lparen.COND rparen lbrace BLOCK rbrace ELSE}	60	{STMT -> if lparen.COND rparen lbrace BLOCK rbrace ELSE; COND -> .COND comp boolstr; COND -> .boolstr}
goto(52, lparen)	{STMT -> while lparen.COND rparen lbrace BLOCK rbrace}	61	{STMT -> while lparen.COND rparen lbrace BLOCK rbrace; COND -> .COND comp boolstr; COND -> .boolstr}
goto(53, id)	{VDECL -> vtype id.semi; ASSIGN -> id.assign RHS}	62	{VDECL -> vtype id.semi; ASSIGN -> id.assign RHS}
goto(53, ASSIGN)	{VDECL -> vtype ASSIGN.semi}	11	
goto(54, assign)	{ASSIGN -> id assign.RHS}	15	
goto(55, id)	{MOREARGS -> comma vtype id.MOREARGS}	63	{MOREARGS -> comma vtype id.MOREARGS; MOREARGS -> .comma vtype id MOREARGS; MOREARGS -> .}
goto(56, rbrace)	{FDECL -> vtype id lparen ARG rparen lbrace BLOCK RETURN rbrace.}	64	{FDECL -> vtype id lparen ARG rparen lbrace BLOCK RETURN rbrace.}
goto(57, RHS)	{RETURN -> return RHS.semi}	65	{RETURN -> return RHS.semi}
goto(57, EXPR)	{RHS -> EXPR.}	21	
goto(57, literal)	{RHS -> literal.}	22	
goto(57, character)	{RHS -> character.}	23	
goto(57, boolstr)	{RHS -> boolstr.}	24	
goto(57, T)	{EXPR -> T.addsub EXPR; EXPR -> T.}	25	
goto(57, lparen)	{EXPR -> lparen.EXPR rparen}	26	
goto(57, F)	{T -> F.multdiv T; T -> F.}	27	
goto(57, id)	{F -> id.}	28	
goto(57, num)	{F -> num.}	29	
goto(60, COND)	{STMT -> if lparen COND.rparen lbrace BLOCK rbrace ELSE; COND -> COND.comp boolstr}	66	{STMT -> if lparen COND.rparen lbrace BLOCK rbrace ELSE; COND -> COND.comp boolstr}
goto(60, boolstr)	{COND -> boolstr.}	67	{COND -> boolstr.}
goto(61, COND)	{STMT -> while lparen COND.rparen lbrace BLOCK rbrace; COND -> COND.comp boolstr}	68	{STMT -> while lparen COND.rparen lbrace BLOCK rbrace; COND -> COND.comp boolstr}
goto(61, boolstr)	{COND -> boolstr.}	67	
goto(62, semi)	{VDECL -> vtype id semi.}	13	
goto(62, assign)	{ASSIGN -> id assign.RHS}	15	
goto(63, MOREARGS)	{MOREARGS -> comma vtype id MOREARGS.}	69	{MOREARGS -> comma vtype id MOREARGS.}
goto(63, comma)	{MOREARGS -> comma.vtype id MOREARGS}	43	
goto(65, semi)	{RETURN -> return RHS semi.}	70	{RETURN -> return RHS semi.}
goto(66, rparen)	{STMT -> if lparen COND rparen.lbrace BLOCK rbrace ELSE}	71	{STMT -> if lparen COND rparen.lbrace BLOCK rbrace ELSE}
goto(66, comp)	{COND -> COND comp.boolstr}	72	{COND -> COND comp.boolstr}
goto(68, rparen)	{STMT -> while lparen COND rparen.lbrace BLOCK rbrace}	73	{STMT -> while lparen COND rparen.lbrace BLOCK rbrace}
goto(68, comp)	{COND -> COND comp.boolstr}	72	
goto(71, lbrace)	{STMT -> if lparen COND rparen lbrace.BLOCK rbrace ELSE}	74	{STMT -> if lparen COND rparen lbrace.BLOCK rbrace ELSE; BLOCK -> .STMT BLOCK; BLOCK -> .; STMT -> .VDECL; STMT -> .ASSIGN semi; STMT -> .if lparen COND rparen lbrace BLOCK rbrace ELSE; STMT -> .while lparen COND rparen lbrace BLOCK rbrace; VDECL -> .vtype id semi; VDECL -> .vtype ASSIGN semi; ASSIGN -> .id assign RHS}
goto(72, boolstr)	{COND -> COND comp boolstr.}	75	{COND -> COND comp boolstr.}
goto(73, lbrace)	{STMT -> while lparen COND rparen lbrace.BLOCK rbrace}	76	{STMT -> while lparen COND rparen lbrace.BLOCK rbrace; BLOCK -> .STMT BLOCK; BLOCK -> .; STMT -> .VDECL; STMT -> .ASSIGN semi; STMT -> .if lparen COND rparen lbrace BLOCK rbrace ELSE; STMT -> .while lparen COND rparen lbrace BLOCK rbrace; VDECL -> .vtype id semi; VDECL -> .vtype ASSIGN semi; ASSIGN -> .id assign RHS}
goto(74, BLOCK)	{STMT -> if lparen COND rparen lbrace BLOCK.rbrace ELSE}	77	{STMT -> if lparen COND rparen lbrace BLOCK.rbrace ELSE}
goto(74, STMT)	{BLOCK -> STMT.BLOCK}	48	
goto(74, VDECL)	{STMT -> VDECL.}	49	
goto(74, ASSIGN)	{STMT -> ASSIGN.semi}	50	
goto(74, if)	{STMT -> if.lparen COND rparen lbrace BLOCK rbrace ELSE}	51	
goto(74, while)	{STMT -> while.lparen COND rparen lbrace BLOCK rbrace}	52	
goto(74, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi}	53	
goto(74, id)	{ASSIGN -> id.assign RHS}	54	
goto(76, BLOCK)	{STMT -> while lparen COND rparen lbrace BLOCK.rbrace}	78	{STMT -> while lparen COND rparen lbrace BLOCK.rbrace}
goto(76, STMT)	{BLOCK -> STMT.BLOCK}	48	
goto(76, VDECL)	{STMT -> VDECL.}	49	
goto(76, ASSIGN)	{STMT -> ASSIGN.semi}	50	
goto(76, if)	{STMT -> if.lparen COND rparen lbrace BLOCK rbrace ELSE}	51	
goto(76, while)	{STMT -> while.lparen COND rparen lbrace BLOCK rbrace}	52	
goto(76, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi}	53	
goto(76, id)	{ASSIGN -> id.assign RHS}	54	
goto(77, rbrace)	{STMT -> if lparen COND rparen lbrace BLOCK rbrace.ELSE}	79	{STMT -> if lparen COND rparen lbrace BLOCK rbrace.ELSE; ELSE -> .else lbrace BLOCK rbrace; ELSE -> .}
goto(78, rbrace)	{STMT -> while lparen COND rparen lbrace BLOCK rbrace.}	80	{STMT -> while lparen COND rparen lbrace BLOCK rbrace.}
goto(79, ELSE)	{STMT -> if lparen COND rparen lbrace BLOCK rbrace ELSE.}	81	{STMT -> if lparen COND rparen lbrace BLOCK rbrace ELSE.}
goto(79, else)	{ELSE -> else.lbrace BLOCK rbrace}	82	{ELSE -> else.lbrace BLOCK rbrace}
goto(82, lbrace)	{ELSE -> else lbrace.BLOCK rbrace}	83	{ELSE -> else lbrace.BLOCK rbrace; BLOCK -> .STMT BLOCK; BLOCK -> .; STMT -> .VDECL; STMT -> .ASSIGN semi; STMT -> .if lparen COND rparen lbrace BLOCK rbrace ELSE; STMT -> .while lparen COND rparen lbrace BLOCK rbrace; VDECL -> .vtype id semi; VDECL -> .vtype ASSIGN semi; ASSIGN -> .id assign RHS}
goto(83, BLOCK)	{ELSE -> else lbrace BLOCK.rbrace}	84	{ELSE -> else lbrace BLOCK.rbrace}
goto(83, STMT)	{BLOCK -> STMT.BLOCK}	48	
goto(83, VDECL)	{STMT -> VDECL.}	49	
goto(83, ASSIGN)	{STMT -> ASSIGN.semi}	50	
goto(83, if)	{STMT -> if.lparen COND rparen lbrace BLOCK rbrace ELSE}	51	
goto(83, while)	{STMT -> while.lparen COND rparen lbrace BLOCK rbrace}	52	
goto(83, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi}	53	
goto(83, id)	{ASSIGN -> id.assign RHS}	54	
goto(84, rbrace)	{ELSE -> else lbrace BLOCK rbrace.}	85	{ELSE -> else lbrace BLOCK rbrace.}

LR table

State	ACTION																						GOTO																	
	vtype	id	semi	assign	literal	character	boolstr	addsub	multdiv	lparen	rparen	num	lbrace	rbrace	comma	if	while	comp	else	return	class	$	START	CODE	VDECL	ASSIGN	RHS	EXPR	T	F	FDECL	ARG	MOREARGS	BLOCK	STMT	COND	ELSE	RETURN	CDECL	ODECL
0	s5																				s6	r_4		1	2						3								4	
1																						acc																		
2	s5																				s6	r_4		7	2						3								4	
3	s5																				s6	r_4		8	2						3								4	
4	s5																				s6	r_4		9	2						3								4	
5		s10																								11														
6		s12																																						
7																						r_1																		
8																						r_2																		
9																						r_3																		
10			s13	s15						s14																														
11			s16																																					
12													s17																											
13	r_5	r_5												r_5		r_5	r_5			r_5	r_5	r_5																		
14	s19										r_{21}																					18								
15		s28			s22	s23	s24			s26		s29															20	21	25	27										
16	r_6	r_6												r_6		r_6	r_6			r_6	r_6	r_6																		
17	s5													r_{38}											31						32									30
18											s33																													
19		s34																																						
20			r_7																																					
21			r_8																																					
22			r_9																																					
23			r_{10}																																					
24			r_{11}																																					
25			r_{13}					s35			r_{13}																													
26		s28								s26		s29																36	25	27										
27			r_{16}					r_{16}	s37		r_{16}																													
28			r_{17}					r_{17}	r_{17}		r_{17}																													
29			r_{18}					r_{18}	r_{18}		r_{18}																													
30														s38																										
31	s5													r_{38}											31						32									39
32	s5													r_{38}											31						32									40
33													s41																											
34											r_{23}				s43																		42							
35		s28								s26		s29																44	25	27										
36											s45																													
37		s28										s29																	46	27										
38	r_{35}																				r_{35}	r_{35}																		
39														r_{36}																										
40														r_{37}																										
41	s53	s54												r_{25}		s51	s52			r_{25}					49	50								47	48					
42											r_{20}																													
43	s55																																							
44			r_{12}								r_{12}																													
45			r_{15}								r_{15}																													
46			r_{14}					r_{14}			r_{14}																													
47																				s57																		56		
48	s53	s54												r_{25}		s51	s52			r_{25}					49	50								58	48					
49	r_{26}	r_{26}												r_{26}		r_{26}	r_{26}			r_{26}																				
50			s59																																					
51										s60																														
52										s61																														
53		s62																								11														
54				s15																																				
55		s63																																						
56														s64																										
57		s28			s22	s23	s24			s26		s29															65	21	25	27										
58														r_{24}						r_{24}																				
59	r_{27}	r_{27}												r_{27}		r_{27}	r_{27}			r_{27}																				
60							s67																													66				
61							s67																													68				
62			s13	s15																																				
63											r_{23}				s43																		69							
64	r_{19}													r_{19}							r_{19}	r_{19}																		
65			s70																																					
66											s71							s72																						
67											r_{31}							r_{31}																						
68											s73							s72																						
69											r_{22}																													
70														r_{34}																										
71													s74																											
72							s75																																	
73													s76																											
74	s53	s54												r_{25}		s51	s52			r_{25}					49	50								77	48					
75											r_{30}							r_{30}																						
76	s53	s54												r_{25}		s51	s52			r_{25}					49	50								78	48					
77														s79																										
78														s80																										
79	r_{33}	r_{33}												r_{33}		r_{33}	r_{33}		s82	r_{33}																	81			
80	r_{29}	r_{29}												r_{29}		r_{29}	r_{29}			r_{29}																				
81	r_{28}	r_{28}												r_{28}		r_{28}	r_{28}			r_{28}																				
82													s83																											
83	s53	s54												r_{25}		s51	s52			r_{25}					49	50								84	48					
84														s85																										
85	r_{32}	r_{32}												r_{32}		r_{32}	r_{32}			r_{32}																				

Input (tokens):

Maximum number of steps: 200

PARSE

Trace				Tree
Step	Stack	Input	Action	
1	0	$		