SLR grammar ('' is ε):

```
(0)   CODE -> VDECL CODE
(1)   CODE -> FDECL CODE
(2)   CODE -> CDECL CODE
(3)   CODE -> ' '
(4)   VDECL -> vtype id semi
(5)   VDECL -> vtype ASSIGN semi
(6)   ASSIGN -> id assign RHS
(7)   RHS -> EXPR
(8)   RHS -> literal
(9)   RHS -> character
(10)  RHS -> boolstr
(11)  EXPR -> lparen EXPR addsub EXPR rparen
(12)  EXPR -> lparen EXPR multdiv EXPR rparen
(13)  EXPR -> lparen EXPR rparen
(14)  EXPR -> id
(15)  EXPR -> num
(16)  FDECL -> vtype id lparen ARG rparen lbrace BLOCK RETURN
(17)  rbrace
(18)  ARG -> vtype id MOREARGS
(19)  ARG -> ' '
(20)  MOREARGS -> comma vtype id MOREARGS
(21)  MOREARGS -> ' '
(22)  BLOCK -> STMT BLOCK
(23)  BLOCK -> ' '
(24)  STMT ->  VDECL
(25)  STMT -> ASSIGN semi
(26)  STMT -> if lparen COND rparen lbrace BLOCK rbrace ELSE
(27)  STMT -> while lparen COND rparen lbrace BLOCK rbrace
(28)  COND -> lparen COND comp COND rparen
(29)  COND -> boolstr
(30)  ELSE -> else lbrace BLOCK rbrace
(31)  ELSE -> ' '
(32)  RETURN -> return RHS semi
(33)  CDECL -> class id lbrace ODECL rbrace
(34)  ODECL -> VDECL ODECL
(35)  ODECL -> FDECL ODECL
      ODECL -> ' '
```

FIRST / FOLLOW table

Nonterminal	FIRST	FOLLOW
CODE	{',vtype,class}	{$}
VDECL	{vtype}	{',vtype,class,id,if,while}
ASSIGN	{id}	{semi}
RHS	{literal,character,boolstr,lparen,id,num}	{semi}
EXPR	{lparen,id,num}	{semi,addsub,rparen,multdiv}
FDECL	{vtype}	{',vtype,class}
ARG	{vtype,'}	{rparen}
MOREARGS	{comma,'}	{rparen}
BLOCK	{',vtype,id,if,while}	{return,rbrace}
STMT	{vtype,id,if,while}	{',vtype,id,if,while}
COND	{lparen,boolstr}	{rparen,comp}
ELSE	{else,'}	{',vtype,id,if,while}
RETURN	{return}	{rbrace}
CDECL	{class}	{',vtype,class}
ODECL	{vtype,'}	{rbrace}

>>

SLR closure table

Goto	Kernel	State	Closure
	{CODE -> .VDECL CODE}	0	{CODE -> .VDECL CODE; VDECL -> .vtype id semi; VDECL -> .vtype ASSIGN semi}
goto(0, VDECL)	{CODE -> VDECL.CODE}	1	{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, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi}	2	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi; ASSIGN -> .id assign RHS}
goto(1, CODE)	{CODE -> VDECL CODE.}	3	{CODE -> VDECL CODE.}
goto(1, VDECL)	{CODE -> VDECL.CODE}	1	
goto(1, FDECL)	{CODE -> FDECL.CODE}	4	{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(1, CDECL)	{CODE -> CDECL.CODE}	5	{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(1, ')	{CODE -> '.'}	6	{CODE -> '.'}
goto(1, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi; FDECL -> vtype.id lparen ARG rparen lbrace BLOCK RETURN rbrace}	7	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi; FDECL -> vtype.id lparen ARG rparen lbrace BLOCK RETURN rbrace; ASSIGN -> .id assign RHS}
goto(1, class)	{CDECL -> class.id lbrace ODECL rbrace}	8	{CDECL -> class.id lbrace ODECL rbrace}
goto(2, id)	{VDECL -> vtype id.semi; ASSIGN -> id.assign RHS}	9	{VDECL -> vtype id.semi; ASSIGN -> id.assign RHS}
goto(2, ASSIGN)	{VDECL -> vtype ASSIGN.semi}	10	{VDECL -> vtype ASSIGN.semi}
goto(4, CODE)	{CODE -> FDECL CODE.}	11	{CODE -> FDECL CODE.}
goto(4, VDECL)	{CODE -> VDECL.CODE}	1	
goto(4, FDECL)	{CODE -> FDECL.CODE}	4	
goto(4, CDECL)	{CODE -> CDECL.CODE}	5	
goto(4, ')	{CODE -> '.'}	6	
goto(4, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi; FDECL -> vtype.id lparen ARG rparen lbrace BLOCK RETURN rbrace}	7	
goto(4, class)	{CDECL -> class.id lbrace ODECL rbrace}	8	
goto(5, CODE)	{CODE -> CDECL CODE.}	12	{CODE -> CDECL CODE.}
goto(5, VDECL)	{CODE -> VDECL.CODE}	1	
goto(5, FDECL)	{CODE -> FDECL.CODE}	4	
goto(5, CDECL)	{CODE -> CDECL.CODE}	5	
goto(5, ')	{CODE -> '.'}	6	
goto(5, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi; FDECL -> vtype.id lparen ARG rparen lbrace BLOCK RETURN rbrace}	7	
goto(5, class)	{CDECL -> class.id lbrace ODECL rbrace}	8	
goto(6, ')	{CODE -> ' '.}	13	{CODE -> ' '.}
goto(7, id)	{VDECL -> vtype id.semi; FDECL -> vtype id.lparen ARG rparen lbrace BLOCK RETURN rbrace; ASSIGN -> id.assign RHS}	14	{VDECL -> vtype id.semi; FDECL -> vtype id.lparen ARG rparen lbrace BLOCK RETURN rbrace; ASSIGN -> id.assign RHS}
goto(7, ASSIGN)	{VDECL -> vtype ASSIGN.semi}	10	
goto(8, id)	{CDECL -> class id.lbrace ODECL rbrace}	15	{CDECL -> class id.lbrace ODECL rbrace}
goto(9, semi)	{VDECL -> vtype id semi.}	16	{VDECL -> vtype id semi.}
goto(9, assign)	{ASSIGN -> id assign.RHS}	17	{ASSIGN -> id assign.RHS; RHS -> .EXPR; RHS -> .literal; RHS -> .character; RHS -> .boolstr; EXPR -> .lparen EXPR addsub EXPR rparen; EXPR -> .lparen EXPR multdiv EXPR rparen; EXPR -> .lparen EXPR rparen; EXPR -> .id; EXPR -> .num}
goto(10, semi)	{VDECL -> vtype ASSIGN semi.}	18	{VDECL -> vtype ASSIGN semi.}
goto(14, semi)	{VDECL -> vtype id semi.}	16	
goto(14, lparen)	{FDECL -> vtype id lparen.ARG rparen lbrace BLOCK RETURN rbrace}	19	{FDECL -> vtype id lparen.ARG rparen lbrace BLOCK RETURN rbrace; ARG -> .vtype id MOREARGS; ARG -> .' '}
goto(14, assign)	{ASSIGN -> id assign.RHS}	17	
goto(15, lbrace)	{CDECL -> class id lbrace.ODECL rbrace}	20	{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(17, RHS)	{ASSIGN -> id assign RHS.}	21	{ASSIGN -> id assign RHS.}
goto(17, EXPR)	{RHS -> EXPR.}	22	{RHS -> EXPR.}
goto(17, literal)	{RHS -> literal.}	23	{RHS -> literal.}
goto(17, character)	{RHS -> character.}	24	{RHS -> character.}
goto(17, boolstr)	{RHS -> boolstr.}	25	{RHS -> boolstr.}
goto(17, lparen)	{EXPR -> lparen.EXPR addsub EXPR rparen; EXPR -> lparen.EXPR multdiv EXPR rparen; EXPR -> lparen.EXPR rparen}	26	{EXPR -> lparen.EXPR addsub EXPR rparen; EXPR -> lparen.EXPR multdiv EXPR rparen; EXPR -> lparen.EXPR rparen; EXPR -> .lparen EXPR addsub EXPR rparen; EXPR -> .lparen EXPR multdiv EXPR rparen; EXPR -> .lparen EXPR rparen; EXPR -> .id; EXPR -> .num}
goto(17, id)	{EXPR -> id.}	27	{EXPR -> id.}
goto(17, num)	{EXPR -> num.}	28	{EXPR -> num.}
goto(19, ARG)	{FDECL -> vtype id lparen ARG.rparen lbrace BLOCK RETURN rbrace}	29	{FDECL -> vtype id lparen ARG.rparen lbrace BLOCK RETURN rbrace}
goto(19, vtype)	{ARG -> vtype.id MOREARGS}	30	{ARG -> vtype.id MOREARGS}
goto(19, ')	{ARG -> '.'}	31	{ARG -> '.'}
goto(20, ODECL)	{CDECL -> class id lbrace ODECL.rbrace}	32	{CDECL -> class id lbrace ODECL.rbrace}
goto(20, VDECL)	{ODECL -> VDECL.ODECL}	33	{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(20, FDECL)	{ODECL -> FDECL.ODECL}	34	{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(20, ')	{ODECL -> '.'}	35	{ODECL -> '.'}
goto(20, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi; FDECL -> vtype.id lparen ARG rparen lbrace BLOCK RETURN rbrace}	7	
goto(26, EXPR)	{EXPR -> lparen EXPR.addsub EXPR rparen; EXPR -> lparen EXPR.multdiv EXPR rparen; EXPR -> lparen EXPR.rparen}	36	{EXPR -> lparen EXPR.addsub EXPR rparen; EXPR -> lparen EXPR.multdiv EXPR rparen; EXPR -> lparen EXPR.rparen}
goto(26, lparen)	{EXPR -> lparen.EXPR addsub EXPR rparen; EXPR -> lparen.EXPR multdiv EXPR rparen; EXPR -> lparen.EXPR rparen}	26	
goto(26, id)	{EXPR -> id.}	27	
goto(26, num)	{EXPR -> num.}	28	
goto(29, rparen)	{FDECL -> vtype id lparen ARG rparen.lbrace BLOCK RETURN rbrace}	37	{FDECL -> vtype id lparen ARG rparen.lbrace BLOCK RETURN rbrace}
goto(30, id)	{ARG -> vtype id.MOREARGS}	38	{ARG -> vtype id.MOREARGS; MOREARGS -> .comma vtype id MOREARGS; MOREARGS -> .' '}
goto(31, ')	{ARG -> ' '.}	39	{ARG -> ' '.}
goto(32, rbrace)	{CDECL -> class id lbrace ODECL rbrace.}	40	{CDECL -> class id lbrace ODECL rbrace.}
goto(33, ODECL)	{ODECL -> VDECL ODECL.}	41	{ODECL -> VDECL ODECL.}
goto(33, VDECL)	{ODECL -> VDECL.ODECL}	33	
goto(33, FDECL)	{ODECL -> FDECL.ODECL}	34	
goto(33, ')	{ODECL -> '.'}	35	
goto(33, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi; FDECL -> vtype.id lparen ARG rparen lbrace BLOCK RETURN rbrace}	7	
goto(34, ODECL)	{ODECL -> FDECL ODECL.}	42	{ODECL -> FDECL ODECL.}
goto(34, VDECL)	{ODECL -> VDECL.ODECL}	33	
goto(34, FDECL)	{ODECL -> FDECL.ODECL}	34	
goto(34, ')	{ODECL -> '.'}	35	
goto(34, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi; FDECL -> vtype.id lparen ARG rparen lbrace BLOCK RETURN rbrace}	7	
goto(35, ')	{ODECL -> ' '.}	43	{ODECL -> ' '.}
goto(36, addsub)	{EXPR -> lparen EXPR addsub.EXPR rparen}	44	{EXPR -> lparen EXPR addsub.EXPR rparen; EXPR -> .lparen EXPR addsub EXPR rparen; EXPR -> .lparen EXPR multdiv EXPR rparen; EXPR -> .lparen EXPR rparen; EXPR -> .id; EXPR -> .num}
goto(36, multdiv)	{EXPR -> lparen EXPR multdiv.EXPR rparen}	45	{EXPR -> lparen EXPR multdiv.EXPR rparen; EXPR -> .lparen EXPR addsub EXPR rparen; EXPR -> .lparen EXPR multdiv EXPR rparen; EXPR -> .lparen EXPR rparen; EXPR -> .id; EXPR -> .num}
goto(36, rparen)	{EXPR -> lparen EXPR rparen.}	46	{EXPR -> lparen EXPR rparen.}
goto(37, lbrace)	{FDECL -> vtype id lparen ARG rparen lbrace.BLOCK RETURN rbrace}	47	{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(38, MOREARGS)	{ARG -> vtype id MOREARGS.}	48	{ARG -> vtype id MOREARGS.}
goto(38, comma)	{MOREARGS -> comma.vtype id MOREARGS}	49	{MOREARGS -> comma.vtype id MOREARGS}
goto(38, ')	{MOREARGS -> '.'}	50	{MOREARGS -> '.'}
goto(44, EXPR)	{EXPR -> lparen EXPR addsub EXPR.rparen}	51	{EXPR -> lparen EXPR addsub EXPR.rparen}
goto(44, lparen)	{EXPR -> lparen.EXPR addsub EXPR rparen; EXPR -> lparen.EXPR multdiv EXPR rparen; EXPR -> lparen.EXPR rparen}	26	
goto(44, id)	{EXPR -> id.}	27	
goto(44, num)	{EXPR -> num.}	28	
goto(45, EXPR)	{EXPR -> lparen EXPR multdiv EXPR.rparen}	52	{EXPR -> lparen EXPR multdiv EXPR.rparen}
goto(45, lparen)	{EXPR -> lparen.EXPR addsub EXPR rparen; EXPR -> lparen.EXPR multdiv EXPR rparen; EXPR -> lparen.EXPR rparen}	26	
goto(45, id)	{EXPR -> id.}	27	
goto(45, num)	{EXPR -> num.}	28	
goto(47, BLOCK)	{FDECL -> vtype id lparen ARG rparen lbrace BLOCK.RETURN rbrace}	53	{FDECL -> vtype id lparen ARG rparen lbrace BLOCK.RETURN rbrace; RETURN -> .return RHS semi}
goto(47, STMT)	{BLOCK -> STMT.BLOCK}	54	{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(47, ')	{BLOCK -> '.'}	55	{BLOCK -> '.'}
goto(47, VDECL)	{STMT -> VDECL.}	56	{STMT -> VDECL.}
goto(47, ASSIGN)	{STMT -> ASSIGN.semi}	57	{STMT -> ASSIGN.semi}
goto(47, if)	{STMT -> if.lparen COND rparen lbrace BLOCK rbrace ELSE}	58	{STMT -> if.lparen COND rparen lbrace BLOCK rbrace ELSE}
goto(47, while)	{STMT -> while.lparen COND rparen lbrace BLOCK rbrace}	59	{STMT -> while.lparen COND rparen lbrace BLOCK rbrace}
goto(47, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi}	2	
goto(47, id)	{ASSIGN -> id.assign RHS}	60	{ASSIGN -> id.assign RHS}
goto(49, vtype)	{MOREARGS -> comma vtype.id MOREARGS}	61	{MOREARGS -> comma vtype.id MOREARGS}
goto(50, ')	{MOREARGS -> ' '.}	62	{MOREARGS -> ' '.}
goto(51, rparen)	{EXPR -> lparen EXPR addsub EXPR rparen.}	63	{EXPR -> lparen EXPR addsub EXPR rparen.}
goto(52, rparen)	{EXPR -> lparen EXPR multdiv EXPR rparen.}	64	{EXPR -> lparen EXPR multdiv EXPR rparen.}
goto(53, RETURN)	{FDECL -> vtype id lparen ARG rparen lbrace BLOCK RETURN.rbrace}	65	{FDECL -> vtype id lparen ARG rparen lbrace BLOCK RETURN.rbrace}
goto(53, return)	{RETURN -> return.RHS semi}	66	{RETURN -> return.RHS semi; RHS -> .EXPR; RHS -> .literal; RHS -> .character; RHS -> .boolstr; EXPR -> .lparen EXPR addsub EXPR rparen; EXPR -> .lparen EXPR multdiv EXPR rparen; EXPR -> .lparen EXPR rparen; EXPR -> .id; EXPR -> .num}
goto(54, BLOCK)	{BLOCK -> STMT BLOCK.}	67	{BLOCK -> STMT BLOCK.}
goto(54, STMT)	{BLOCK -> STMT.BLOCK}	54	
goto(54, ')	{BLOCK -> '.'}	55	
goto(54, VDECL)	{STMT -> VDECL.}	56	
goto(54, ASSIGN)	{STMT -> ASSIGN.semi}	57	
goto(54, if)	{STMT -> if.lparen COND rparen lbrace BLOCK rbrace ELSE}	58	
goto(54, while)	{STMT -> while.lparen COND rparen lbrace BLOCK rbrace}	59	
goto(54, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi}	2	
goto(54, id)	{ASSIGN -> id.assign RHS}	60	
goto(55, ')	{BLOCK -> ' '.}	68	{BLOCK -> ' '.}
goto(57, semi)	{STMT -> ASSIGN semi.}	69	{STMT -> ASSIGN semi.}
goto(58, lparen)	{STMT -> if lparen.COND rparen lbrace BLOCK rbrace ELSE}	70	{STMT -> if lparen.COND rparen lbrace BLOCK rbrace ELSE; COND -> .lparen COND comp COND rparen; COND -> .boolstr}
goto(59, lparen)	{STMT -> while lparen.COND rparen lbrace BLOCK rbrace}	71	{STMT -> while lparen.COND rparen lbrace BLOCK rbrace; COND -> .lparen COND comp COND rparen; COND -> .boolstr}
goto(60, assign)	{ASSIGN -> id assign.RHS}	17	
goto(61, id)	{MOREARGS -> comma vtype id.MOREARGS}	72	{MOREARGS -> comma vtype id.MOREARGS; MOREARGS -> .comma vtype id MOREARGS; MOREARGS -> .' '}
goto(65, rbrace)	{FDECL -> vtype id lparen ARG rparen lbrace BLOCK RETURN rbrace.}	73	{FDECL -> vtype id lparen ARG rparen lbrace BLOCK RETURN rbrace.}
goto(66, RHS)	{RETURN -> return RHS.semi}	74	{RETURN -> return RHS.semi}
goto(66, EXPR)	{RHS -> EXPR.}	22	
goto(66, literal)	{RHS -> literal.}	23	
goto(66, character)	{RHS -> character.}	24	
goto(66, boolstr)	{RHS -> boolstr.}	25	
goto(66, lparen)	{EXPR -> lparen.EXPR addsub EXPR rparen; EXPR -> lparen.EXPR multdiv EXPR rparen; EXPR -> lparen.EXPR rparen}	26	
goto(66, id)	{EXPR -> id.}	27	
goto(66, num)	{EXPR -> num.}	28	
goto(70, COND)	{STMT -> if lparen COND.rparen lbrace BLOCK rbrace ELSE}	75	{STMT -> if lparen COND.rparen lbrace BLOCK rbrace ELSE}
goto(70, lparen)	{COND -> lparen.COND comp COND rparen}	76	{COND -> lparen.COND comp COND rparen; COND -> .lparen COND comp COND rparen; COND -> .boolstr}
goto(70, boolstr)	{COND -> boolstr.}	77	{COND -> boolstr.}
goto(71, COND)	{STMT -> while lparen COND.rparen lbrace BLOCK rbrace}	78	{STMT -> while lparen COND.rparen lbrace BLOCK rbrace}
goto(71, lparen)	{COND -> lparen.COND comp COND rparen}	76	
goto(71, boolstr)	{COND -> boolstr.}	77	
goto(72, MOREARGS)	{MOREARGS -> comma vtype id MOREARGS.}	79	{MOREARGS -> comma vtype id MOREARGS.}
goto(72, comma)	{MOREARGS -> comma.vtype id MOREARGS}	49	
goto(72, ')	{MOREARGS -> '.'}	50	
goto(74, semi)	{RETURN -> return RHS semi.}	80	{RETURN -> return RHS semi.}
goto(75, rparen)	{STMT -> if lparen COND rparen.lbrace BLOCK rbrace ELSE}	81	{STMT -> if lparen COND rparen.lbrace BLOCK rbrace ELSE}
goto(76, COND)	{COND -> lparen COND.comp COND rparen}	82	{COND -> lparen COND.comp COND rparen}
goto(76, lparen)	{COND -> lparen.COND comp COND rparen}	76	
goto(76, boolstr)	{COND -> boolstr.}	77	
goto(78, rparen)	{STMT -> while lparen COND rparen.lbrace BLOCK rbrace}	83	{STMT -> while lparen COND rparen.lbrace BLOCK rbrace}
goto(81, lbrace)	{STMT -> if lparen COND rparen lbrace.BLOCK rbrace ELSE}	84	{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(82, comp)	{COND -> lparen COND comp.COND rparen}	85	{COND -> lparen COND comp.COND rparen; COND -> .lparen COND comp COND rparen; COND -> .boolstr}
goto(83, lbrace)	{STMT -> while lparen COND rparen lbrace.BLOCK rbrace}	86	{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(84, BLOCK)	{STMT -> if lparen COND rparen lbrace BLOCK.rbrace ELSE}	87	{STMT -> if lparen COND rparen lbrace BLOCK.rbrace ELSE}
goto(84, STMT)	{BLOCK -> STMT.BLOCK}	54	
goto(84, ')	{BLOCK -> '.'}	55	
goto(84, VDECL)	{STMT -> VDECL.}	56	
goto(84, ASSIGN)	{STMT -> ASSIGN.semi}	57	
goto(84, if)	{STMT -> if.lparen COND rparen lbrace BLOCK rbrace ELSE}	58	
goto(84, while)	{STMT -> while.lparen COND rparen lbrace BLOCK rbrace}	59	
goto(84, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi}	2	
goto(84, id)	{ASSIGN -> id.assign RHS}	60	
goto(85, COND)	{COND -> lparen COND comp COND.rparen}	88	{COND -> lparen COND comp COND.rparen}
goto(85, lparen)	{COND -> lparen.COND comp COND rparen}	76	
goto(85, boolstr)	{COND -> boolstr.}	77	
goto(86, BLOCK)	{STMT -> while lparen COND rparen lbrace BLOCK.rbrace}	89	{STMT -> while lparen COND rparen lbrace BLOCK.rbrace}
goto(86, STMT)	{BLOCK -> STMT.BLOCK}	54	
goto(86, ')	{BLOCK -> '.'}	55	
goto(86, VDECL)	{STMT -> VDECL.}	56	
goto(86, ASSIGN)	{STMT -> ASSIGN.semi}	57	
goto(86, if)	{STMT -> if.lparen COND rparen lbrace BLOCK rbrace ELSE}	58	
goto(86, while)	{STMT -> while.lparen COND rparen lbrace BLOCK rbrace}	59	
goto(86, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi}	2	
goto(86, id)	{ASSIGN -> id.assign RHS}	60	
goto(87, rbrace)	{STMT -> if lparen COND rparen lbrace BLOCK rbrace.ELSE}	90	{STMT -> if lparen COND rparen lbrace BLOCK rbrace.ELSE; ELSE -> .else lbrace BLOCK rbrace; ELSE -> .' '}
goto(88, rparen)	{COND -> lparen COND comp COND rparen.}	91	{COND -> lparen COND comp COND rparen.}
goto(89, rbrace)	{STMT -> while lparen COND rparen lbrace BLOCK rbrace.}	92	{STMT -> while lparen COND rparen lbrace BLOCK rbrace.}
goto(90, ELSE)	{STMT -> if lparen COND rparen lbrace BLOCK rbrace ELSE.}	93	{STMT -> if lparen COND rparen lbrace BLOCK rbrace ELSE.}
goto(90, else)	{ELSE -> else.lbrace BLOCK rbrace}	94	{ELSE -> else.lbrace BLOCK rbrace}
goto(90, ')	{ELSE -> '.'}	95	{ELSE -> '.'}
goto(94, lbrace)	{ELSE -> else lbrace.BLOCK rbrace}	96	{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(95, ')	{ELSE -> ' '.}	97	{ELSE -> ' '.}
goto(96, BLOCK)	{ELSE -> else lbrace BLOCK.rbrace}	98	{ELSE -> else lbrace BLOCK.rbrace}
goto(96, STMT)	{BLOCK -> STMT.BLOCK}	54	
goto(96, ')	{BLOCK -> '.'}	55	
goto(96, VDECL)	{STMT -> VDECL.}	56	
goto(96, ASSIGN)	{STMT -> ASSIGN.semi}	57	
goto(96, if)	{STMT -> if.lparen COND rparen lbrace BLOCK rbrace ELSE}	58	
goto(96, while)	{STMT -> while.lparen COND rparen lbrace BLOCK rbrace}	59	
goto(96, vtype)	{VDECL -> vtype.id semi; VDECL -> vtype.ASSIGN semi}	2	
goto(96, id)	{ASSIGN -> id.assign RHS}	60	
goto(98, rbrace)	{ELSE -> else lbrace BLOCK rbrace.}	99	{ELSE -> else lbrace BLOCK rbrace.}

LR table

State	ACTION																							GOTO														
	'	vtype	id	semi	assign	literal	character	boolstr	lparen	addsub	rparen	multdiv	num	lbrace	rbrace	comma	if	while	comp	else	return	class	$	CODE	VDECL	ASSIGN	RHS	EXPR	FDECL	ARG	MOREARGS	BLOCK	STMT	COND	ELSE	RETURN	CDECL	ODECL
0		s2																							1													
1	s6	s7																				s8		3	1				4								5	
2			s9																							10												
3																							acc															
4	s6	s7																				s8		11	1				4								5	
5	s6	s7																				s8		12	1				4								5	
6	s13																																					
7			s14																							10												
8			s15																																			
9				s16	s17																																	
10				s18																																		
11																							r_{1}															
12																							r_{2}															
13																							r_{3}															
14				s16	s17				s19																													
15														s20																								
16	r_{4}	r_{4}	r_{4}														r_{4}	r_{4}				r_{4}																
17			s27			s23	s24	s25	s26				s28														21	22										
18	r_{5}	r_{5}	r_{5}														r_{5}	r_{5}				r_{5}																
19	s31	s30																												29								
20	s35	s7																							33				34									32
21				r_{6}																																		
22				r_{7}																																		
23				r_{8}																																		
24				r_{9}																																		
25				r_{10}																																		
26			s27						s26				s28															36										
27				r_{14}						r_{14}	r_{14}	r_{14}																										
28				r_{15}						r_{15}	r_{15}	r_{15}																										
29											s37																											
30			s38																																			
31	s39																																					
32															s40																							
33	s35	s7																							33				34									41
34	s35	s7																							33				34									42
35	s43																																					
36										s44	s46	s45																										
37														s47																								
38	s50															s49															48							
39											r_{18}																											
40	r_{32}	r_{32}																				r_{32}																
41															r_{33}																							
42															r_{34}																							
43															r_{35}																							
44			s27						s26				s28															51										
45			s27						s26				s28															52										
46				r_{13}						r_{13}	r_{13}	r_{13}																										
47	s55	s2	s60														s58	s59							56	57						53	54					
48											r_{17}																											
49		s61																																				
50	s62																																					
51											s63																											
52											s64																											
53																					s66															65		
54	s55	s2	s60														s58	s59							56	57						67	54					
55	s68																																					
56	r_{23}	r_{23}	r_{23}														r_{23}	r_{23}																				
57				s69																																		
58									s70																													
59									s71																													
60					s17																																	
61			s72																																			
62											r_{20}																											
63				r_{11}						r_{11}	r_{11}	r_{11}																										
64				r_{12}						r_{12}	r_{12}	r_{12}																										
65															s73																							
66			s27			s23	s24	s25	s26				s28														74	22										
67															r_{21}						r_{21}																	
68															r_{22}						r_{22}																	
69	r_{24}	r_{24}	r_{24}														r_{24}	r_{24}																				
70								s77	s76																									75				
71								s77	s76																									78				
72	s50															s49															79							
73	r_{16}	r_{16}																				r_{16}																
74				s80																																		
75											s81																											
76								s77	s76																									82				
77											r_{28}								r_{28}																			
78											s83																											
79											r_{19}																											
80															r_{31}																							
81														s84																								
82																			s85																			
83														s86																								
84	s55	s2	s60														s58	s59							56	57						87	54					
85								s77	s76																									88				
86	s55	s2	s60														s58	s59							56	57						89	54					
87															s90																							
88											s91																											
89															s92																							
90	s95																			s94															93			
91											r_{27}								r_{27}																			
92	r_{26}	r_{26}	r_{26}														r_{26}	r_{26}																				
93	r_{25}	r_{25}	r_{25}														r_{25}	r_{25}																				
94														s96																								
95	s97																																					
96	s55	s2	s60														s58	s59							56	57						98	54					
97	r_{30}	r_{30}	r_{30}														r_{30}	r_{30}																				
98															s99																							
99	r_{29}	r_{29}	r_{29}														r_{29}	r_{29}																				

Input (tokens): id + id * id

Maximum number of steps: 100

PARSE

Trace				Tree
Step	Stack	Input	Action	
1	0	id + id * id $		