FIRST / FOLLOW table		
Nonterminal	**FIRST**	**FOLLOW**
Axioma_	{,let,id,alert,input,return,function,if}	{$}
Axioma	{,let,id,alert,input,return,function,if}	{$}
Sentencia	{let,id,alert,input,return,if}	{,let,id,alert,input,return,function,if,else}
IF_	{if}	{,let,id,alert,input,return,function,if,else}
IF1	{if}	{else,}
IF2	{else,}	{,let,id,alert,input,return,function,if,else}
Senten	{let,{,id,alert,input,return,if}	{else,,let,id,alert,input,return,function,if}
S	{id,alert,input,return}	{,let,id,alert,input,return,function,if,else}
Parametros	{,},id,(,ent,cadena}	{)}
K2	{,,}	{)}
K	{,},id,(,ent,cadena}	{;}
Funcion	{function}	{,let,id,alert,input,return,function,if}
F1	{function}	{(}
F2	{(}	{(}
F3	{[}	{,let,id,alert,input,return,function,if}
Tipo	{boolean,number,string}	{id}
Tipo_B	{boolean,number,string,}	{id}
Cabecera	{boolean,number,string,}	{)}
K	{,,}	{)}
Lista_Sentencias	{let,,id,alert,input,return,if}	{)}
E	{!,id,(,ent,cadena}	{),;,,,}
R	{id,(,ent,cadena}	{),;,,,,>,<}
U	{id,(,ent,cadena}	{),;,,,,>,<,+,-}
V	{id,(,ent,cadena}	{),;,,,,>,<,+,-}

SLR closure table			
Goto	**Kernel**	**State**	**Closure**
	{Axioma_ -> .Axioma}	0	{Axioma_ -> .Axioma; Axioma -> .Sentencia Axioma; Axioma -> .Funcion Axioma; Axioma -> .; Sentencia -> .S; Sentencia -> .IF_; Sentencia -> .let Tipo id ;; Funcion -> .F1 F2 F3; S -> .id = E ;; S -> .id %= E ;; S -> .id (Parametros) ;; S -> .alert (E) ;; S -> .input (id) ;; S -> .return X ;; IF_ -> .IF1 IF2; F1 -> .function Tipo_B id; IF1 -> .if (E) Senten}
goto(0, Axioma)	{Axioma_ -> Axioma.}	1	{Axioma_ -> Axioma.}
goto(0, Sentencia)	{Axioma -> Sentencia.Axioma}	2	{Axioma -> Sentencia.Axioma; Axioma -> .Sentencia Axioma; Axioma -> .Funcion Axioma; Axioma -> .; Sentencia -> .S; Sentencia -> .IF_; Sentencia -> .let Tipo id ;; Funcion -> .F1 F2 F3; S -> .id = E ;; S -> .id %= E ;; S -> .id (Parametros) ;; S -> .alert (E) ;; S -> .input (id) ;; S -> .return X ;; IF_ -> .IF1 IF2; F1 -> .function Tipo_B id; IF1 -> .if (E) Senten}
goto(0, Funcion)	{Axioma -> Funcion.Axioma}	3	{Axioma -> Funcion.Axioma; Axioma -> .Sentencia Axioma; Axioma -> .Funcion Axioma; Axioma -> .; Sentencia -> .S; Sentencia -> .IF_; Sentencia -> .let Tipo id ;; Funcion -> .F1 F2 F3; S -> .id = E ;; S -> .id %= E ;; S -> .id (Parametros) ;; S -> .alert (E) ;; S -> .input (id) ;; S -> .return X ;; IF_ -> .IF1 IF2; F1 -> .function Tipo_B id; IF1 -> .if (E) Senten}
goto(0,)	{Axioma -> .}	4	{Axioma -> .}
goto(0, S)	{Sentencia -> S.}	5	{Sentencia -> S.}
goto(0, IF_)	{Sentencia -> IF_.}	6	{Sentencia -> IF_.}
goto(0, let)	{Sentencia -> let.Tipo id ;}	7	{Sentencia -> let.Tipo id ;; Tipo -> .boolean; Tipo -> .number; Tipo -> .string}
goto(0, F1)	{Funcion -> F1.F2 F3}	8	{Funcion -> F1.F2 F3; F2 -> .(Cabecera)}
goto(0, id)	{S -> id.= E ;; S -> id.%= E ;; S -> id.(Parametros) ;}	9	{S -> id.= E ;; S -> id.%= E ;; S -> id.(Parametros) ;}
goto(0, alert)	{S -> alert.(E) ;}	10	{S -> alert.(E) ;}
goto(0, input)	{S -> input.(id) ;}	11	{S -> input.(id) ;}
goto(0, return)	{S -> return.X ;}	12	{S -> return.X ;; X -> .E; X -> .; E -> .! V; E -> .R; R -> .R > U; R -> .R < U; R -> .U; U -> .U + V; U -> .U - V; U -> .V; V -> .id; V -> .(E); V -> .id (Parametros); V -> .ent; V -> .cadena}
goto(0, IF1)	{IF_ -> IF1.IF2}	13	{IF_ -> IF1.IF2; IF2 -> .else Senten; IF2 -> .}
goto(0, function)	{F1 -> function.Tipo_B id}	14	{F1 -> function.Tipo_B id; Tipo_B -> .Tipo; Tipo_B -> .; Tipo -> .boolean; Tipo -> .number; Tipo -> .string}
goto(0, if)	{IF1 -> if.(E) Senten}	15	{IF1 -> if.(E) Senten}
goto(2, Axioma)	{Axioma -> Sentencia Axioma.}	16	{Axioma -> Sentencia Axioma.}
goto(2, Sentencia)	{Axioma -> Sentencia.Axioma}	2	
goto(2, Funcion)	{Axioma -> Funcion.Axioma}	3	
goto(2,)	{Axioma -> .}	4	
goto(2, S)	{Sentencia -> S.}	5	
goto(2, IF_)	{Sentencia -> IF_.}	6	
goto(2, let)	{Sentencia -> let.Tipo id ;}	7	
goto(2, F1)	{Funcion -> F1.F2 F3}	8	
goto(2, id)	{S -> id.= E ;; S -> id.%= E ;; S -> id.(Parametros) ;}	9	
goto(2, alert)	{S -> alert.(E) ;}	10	
goto(2, input)	{S -> input.(id) ;}	11	
goto(2, return)	{S -> return.X ;}	12	
goto(2, IF1)	{IF_ -> IF1.IF2}	13	
goto(2, function)	{F1 -> function.Tipo_B id}	14	
goto(2, if)	{IF1 -> if.(E) Senten}	15	
goto(3, Axioma)	{Axioma -> Funcion Axioma.}	17	{Axioma -> Funcion Axioma.}
goto(3, Sentencia)	{Axioma -> Sentencia.Axioma}	2	
goto(3, Funcion)	{Axioma -> Funcion.Axioma}	3	
goto(3,)	{Axioma -> .}	4	
goto(3, S)	{Sentencia -> S.}	5	
goto(3, IF_)	{Sentencia -> IF_.}	6	
goto(3, let)	{Sentencia -> let.Tipo id ;}	7	
goto(3, F1)	{Funcion -> F1.F2 F3}	8	
goto(3, id)	{S -> id.= E ;; S -> id.%= E ;; S -> id.(Parametros) ;}	9	
goto(3, alert)	{S -> alert.(E) ;}	10	
goto(3, input)	{S -> input.(id) ;}	11	
goto(3, return)	{S -> return.X ;}	12	
goto(3, IF1)	{IF_ -> IF1.IF2}	13	
goto(3, function)	{F1 -> function.Tipo_B id}	14	
goto(3, if)	{IF1 -> if.(E) Senten}	15	
goto(7, Tipo)	{Sentencia -> let Tipo.id ;}	18	{Sentencia -> let Tipo.id ;}
goto(7, boolean)	{Tipo -> boolean.}	19	{Tipo -> boolean.}
goto(7, number)	{Tipo -> number.}	20	{Tipo -> number.}
goto(7, string)	{Tipo -> string.}	21	{Tipo -> string.}
goto(8, F2)	{Funcion -> F1 F2.F3}	22	{Funcion -> F1 F2.F3; F3 -> .{ Lista_Sentencias }}
goto(8, ()	{F2 -> (.Cabecera)}	23	{F2 -> (.Cabecera); Cabecera -> .Tipo id K; Cabecera -> .; Tipo -> .boolean; Tipo -> .number; Tipo -> .string}
goto(9, =)	{S -> id =.E ;}	24	{S -> id =.E ;; E -> .! V; E -> .R; R -> .R > U; R -> .R < U; R -> .U; U -> .U + V; U -> .U - V; U -> .V; V -> .id; V -> .(E); V -> .id (Parametros); V -> .ent; V -> .cadena}
goto(9, %=)	{S -> id %=.E ;}	25	{S -> id %=.E ;; E -> .! V; E -> .R; R -> .R > U; R -> .R < U; R -> .U; U -> .U + V; U -> .U - V; U -> .V; V -> .id; V -> .(E); V -> .id (Parametros); V -> .ent; V -> .cadena}
goto(9, ()	{S -> id (.Parametros) ;; Parametros -> .E K2; Parametros -> .; E -> .! V; E -> .R; R -> .R > U; R -> .R < U; R -> .U; U -> .U + V; U -> .U - V; U -> .V; V -> .id; V -> .(E); V -> .id (Parametros); V -> .ent; V -> .cadena}
goto(10, ()	{S -> alert (.E) ;}	27	{S -> alert (.E) ;; E -> .! V; E -> .R; R -> .R > U; R -> .R < U; R -> .U; U -> .U + V; U -> .U - V; U -> .V; V -> .id; V -> .(E); V -> .id (Parametros); V -> .ent; V -> .cadena}
goto(11, ()	{S -> input (.id) ;}	28	{S -> input (.id) ;}
goto(12, X)	{S -> return X.;}	29	{S -> return X.;}
goto(12, E)	{X -> E.}	30	{X -> E.}
goto(12,)	{X -> .}	31	{X -> .}
goto(12, !)	{E -> !.V}	32	{E -> !.V; V -> .id; V -> .(E); V -> .id (Parametros); V -> .ent; V -> .cadena}
goto(12, R)	{E -> R.; R -> R.> U; R -> R.< U}	33	{E -> R.; R -> R.> U; R -> R.< U}
goto(12, U)	{R -> U.; U -> U.+ V; U -> U.- V}	34	{R -> U.; U -> U.+ V; U -> U.- V}
goto(12, V)	{U -> V.}	35	{U -> V.}
goto(12, id)	{V -> id.; V -> id.(Parametros)}	36	{V -> id.; V -> id.(Parametros)}
goto(12, ()	{V -> (.E)}	37	{V -> (.E); E -> .! V; E -> .R; R -> .R > U; R -> .R < U; R -> .U; U -> .U + V; U -> .U - V; U -> .V; V -> .id; V -> .(E); V -> .id (Parametros); V -> .ent; V -> .cadena}
goto(12, ent)	{V -> ent.}	38	{V -> ent.}
goto(12, cadena)	{V -> cadena.}	39	{V -> cadena.}
goto(13, IF2)	{IF_ -> IF1 IF2.}	40	{IF_ -> IF1 IF2.}
goto(13, else)	{IF2 -> else.Senten}	41	{IF2 -> else.Senten; Senten -> .Sentencia; Senten -> .{ Lista_Sentencias }; Sentencia -> .S; Sentencia -> .IF_; Sentencia -> .let Tipo id ;; S -> .id = E ;; S -> .id %= E ;; S -> .id (Parametros) ;; S -> .alert (E) ;; S -> .input (id) ;; S -> .return X ;; IF_ -> .IF1 IF2; IF1 -> .if (E) Senten}
goto(13,)	{IF2 -> .}	42	{IF2 -> .}
goto(14, Tipo_B)	{F1 -> function Tipo_B.id}	43	{F1 -> function Tipo_B.id}
goto(14, Tipo)	{Tipo_B -> Tipo.}	44	{Tipo_B -> Tipo.}
goto(14,)	{Tipo_B -> .}	45	{Tipo_B -> .}
goto(14, boolean)	{Tipo -> boolean.}	19	
goto(14, number)	{Tipo -> number.}	20	
goto(14, string)	{Tipo -> string.}	21	
goto(15, ()	{IF1 -> if (.E) Senten}	46	{IF1 -> if (.E) Senten; E -> .! V; E -> .R; R -> .R > U; R -> .R < U; R -> .U; U -> .U + V; U -> .U - V; U -> .V; V -> .id; V -> .(E); V -> .id (Parametros); V -> .ent; V -> .cadena}
goto(18, id)	{Sentencia -> let Tipo id.;}	47	{Sentencia -> let Tipo id.;}
goto(22, F3)	{Funcion -> F1 F2 F3.}	48	{Funcion -> F1 F2 F3.}
goto(22, {)	{F3 -> {.Lista_Sentencias }}	49	{F3 -> {.Lista_Sentencias }; Lista_Sentencias -> .Sentencia Lista_Sentencias; Lista_Sentencias -> .; Sentencia -> .S; Sentencia -> .IF_; Sentencia -> .let Tipo id ;; S -> .id = E ;; S -> .id %= E ;; S -> .id (Parametros) ;; S -> .alert (E) ;; S -> .input (id) ;; S -> .return X ;; IF_ -> .IF1 IF2; IF1 -> .if (E) Senten}
goto(23, Cabecera)	{F2 -> (Cabecera.)}	50	{F2 -> (Cabecera.)}
goto(23, Tipo)	{Cabecera -> Tipo.id K}	51	{Cabecera -> Tipo.id K}
goto(23,)	{Cabecera -> .}	52	{Cabecera -> .}
goto(23, boolean)	{Tipo -> boolean.}	19	
goto(23, number)	{Tipo -> number.}	20	
goto(23, string)	{Tipo -> string.}	21	
goto(24, E)	{S -> id = E.;}	53	{S -> id = E.;}
goto(24, !)	{E -> !.V}	32	
goto(24, R)	{E -> R.; R -> R.> U; R -> R.< U}	33	
goto(24, U)	{R -> U.; U -> U.+ V; U -> U.- V}	34	
goto(24, V)	{U -> V.}	35	
goto(24, id)	{V -> id.; V -> id.(Parametros)}	36	
goto(24, ()	{V -> (.E)}	37	
goto(24, ent)	{V -> ent.}	38	
goto(24, cadena)	{V -> cadena.}	39	
goto(25, E)	{S -> id %= E.;}	54	{S -> id %= E.;}
goto(25, !)	{E -> !.V}	32	
goto(25, R)	{E -> R.; R -> R.> U; R -> R.< U}	33	
goto(25, U)	{R -> U.; U -> U.+ V; U -> U.- V}	34	
goto(25, V)	{U -> V.}	35	

SLR closure table

Goto	Kernel	State	Closure
goto(25, id)	{V -> id.; V -> id.(Parametros)}	36	
goto(25, ()	{V -> (.E)}	37	
goto(25, ent)	{V -> ent.}	38	
goto(25, cadena)	{V -> cadena.}	39	
goto(26, Parametros)	{S -> id (Parametros.) ;}	55	{S -> id (Parametros.) ;}
goto(26, E)	{Parametros -> E.K2}	56	{Parametros -> E.K2; K2 -> ., E K2; K2 -> .}
goto(26,)	{Parametros -> .}	57	{Parametros -> .}
goto(26, !)	{E -> !.V}	32	
goto(26, R)	{E -> R.; R -> R.> U; R -> R.< U}	33	
goto(26, U)	{R -> U.; U -> U.+ V; U -> U.- V}	34	
goto(26, V)	{U -> V.}	35	
goto(26, id)	{V -> id.; V -> id.(Parametros)}	36	
goto(26, ()	{V -> (.E)}	37	
goto(26, ent)	{V -> ent.}	38	
goto(26, cadena)	{V -> cadena.}	39	
goto(27, E)	{S -> alert (E.) ;}	58	{S -> alert (E.) ;}
goto(27, !)	{E -> !.V}	32	
goto(27, R)	{E -> R.; R -> R.> U; R -> R.< U}	33	
goto(27, U)	{R -> U.; U -> U.+ V; U -> U.- V}	34	
goto(27, V)	{U -> V.}	35	
goto(27, id)	{V -> id.; V -> id.(Parametros)}	36	
goto(27, ()	{V -> (.E)}	37	
goto(27, ent)	{V -> ent.}	38	
goto(27, cadena)	{V -> cadena.}	39	
goto(28, id)	{S -> input (id.) ;}	59	{S -> input (id.) ;}
goto(29, ;)	{S -> return X ;.}	60	{S -> return X ;.}
goto(32, V)	{E -> ! V.}	61	{E -> ! V.}
goto(32, id)	{V -> id.; V -> id.(Parametros)}	36	
goto(32, ()	{V -> (.E)}	37	
goto(32, ent)	{V -> ent.}	38	
goto(32, cadena)	{V -> cadena.}	39	
goto(33, >)	{R -> R >.U}	62	{R -> R >.U; U -> .U + V; U -> .U - V; U -> .V; V -> .id; V -> .(E); V -> .id (Parametros); V -> .ent; V -> .cadena}
goto(33, <)	{R -> R <.U}	63	{R -> R <.U; U -> .U + V; U -> .U - V; U -> .V; V -> .id; V -> .(E); V -> .id (Parametros); V -> .ent; V -> .cadena}
goto(34, +)	{U -> U +.V}	64	{U -> U +.V; V -> .id; V -> .(E); V -> .id (Parametros); V -> .ent; V -> .cadena}
goto(34, -)	{U -> U -.V}	65	{U -> U -.V; V -> .id; V -> .(E); V -> .id (Parametros); V -> .ent; V -> .cadena}
goto(36, ()	{V -> id (.Parametros)}	66	{V -> id (.Parametros); Parametros -> .E K2; Parametros -> ., E -> .! V; E -> .R; R -> .R > U; R -> .R < U; R -> .U; U -> .U + V; U -> .U - V; U -> .V; V -> .id; V -> .(E); V -> .id (Parametros); V -> .ent; V -> .cadena}
goto(37, E)	{V -> (E.)}	67	{V -> (E.)}
goto(37, !)	{E -> !.V}	32	
goto(37, R)	{E -> R.; R -> R.> U; R -> R.< U}	33	
goto(37, U)	{R -> U.; U -> U.+ V; U -> U.- V}	34	
goto(37, V)	{U -> V.}	35	
goto(37, id)	{V -> id.; V -> id.(Parametros)}	36	
goto(37, ()	{V -> (.E)}	37	
goto(37, ent)	{V -> ent.}	38	
goto(37, cadena)	{V -> cadena.}	39	
goto(41, Senten)	{IF2 -> else Senten.}	68	{IF2 -> else Senten.}
goto(41, Sentencia)	{Senten -> Sentencia.}	69	{Senten -> Sentencia.}
goto(41, ()	{Senten -> (.Lista_Sentencias)}	70	{Senten -> (.Lista_Sentencias); Lista_Sentencias -> .Sentencia Lista_Sentencias; Lista_Sentencias -> .; Sentencia -> .S; Sentencia -> .IF_; Sentencia -> .let Tipo id ;; S -> .id = E ;; S -> .id %= E ;; S -> .id (Parametros) ;; S -> .alert (E) ;; S -> .input (id) ;; S -> .return X ;; IF_ -> .IF1 IF2; IF1 -> .if (E) Senten}
goto(41, S)	{Sentencia -> S.}	5	
goto(41, IF_)	{Sentencia -> IF_.}	6	
goto(41, let)	{Sentencia -> let.Tipo id ;}	7	
goto(41, id)	{S -> id.= E ;; S -> id.%= E ;; S -> id.(Parametros) ;}	9	
goto(41, alert)	{S -> alert.(E) ;}	10	
goto(41, input)	{S -> input.(id) ;}	11	
goto(41, return)	{S -> return.X ;}	12	
goto(41, IF1)	{IF_ -> IF1.IF2}	13	
goto(41, if)	{IF1 -> if.(E) Senten}	15	
goto(43, id)	{F1 -> function Tipo_B id.}	71	{F1 -> function Tipo_B id.}
goto(46, E)	{IF1 -> if (E.) Senten}	72	{IF1 -> if (E.) Senten}
goto(46, !)	{E -> !.V}	32	
goto(46, R)	{E -> R.; R -> R.> U; R -> R.< U}	33	
goto(46, U)	{R -> U.; U -> U.+ V; U -> U.- V}	34	
goto(46, V)	{U -> V.}	35	
goto(46, id)	{V -> id.; V -> id.(Parametros)}	36	
goto(46, ()	{V -> (.E)}	37	
goto(46, ent)	{V -> ent.}	38	
goto(46, cadena)	{V -> cadena.}	39	
goto(47, ;)	{Sentencia -> let Tipo id ;.}	73	{Sentencia -> let Tipo id ;.}
goto(49, Lista_Sentencias)	{F3 -> { Lista_Sentencias.}}	74	{F3 -> { Lista_Sentencias.}}
goto(49, Sentencia)	{Lista_Sentencias -> Sentencia.Lista_Sentencias}	75	{Lista_Sentencias -> Sentencia.Lista_Sentencias; Lista_Sentencias -> .Sentencia Lista_Sentencias; Lista_Sentencias -> .; Sentencia -> .S; Sentencia -> .IF_; Sentencia -> .let Tipo id ;; S -> .id = E ;; S -> .id %= E ;; S -> .id (Parametros) ;; S -> .alert (E) ;; S -> .input (id) ;; S -> .return X ;; IF_ -> .IF1 IF2; IF1 -> .if (E) Senten}
goto(49,)	{Lista_Sentencias -> .}	76	{Lista_Sentencias -> .}
goto(49, S)	{Sentencia -> S.}	5	
goto(49, IF_)	{Sentencia -> IF_.}	6	
goto(49, let)	{Sentencia -> let.Tipo id ;}	7	
goto(49, id)	{S -> id.= E ;; S -> id.%= E ;; S -> id.(Parametros) ;}	9	
goto(49, alert)	{S -> alert.(E) ;}	10	
goto(49, input)	{S -> input.(id) ;}	11	
goto(49, return)	{S -> return.X ;}	12	
goto(49, IF1)	{IF_ -> IF1.IF2}	13	
goto(49, if)	{IF1 -> if.(E) Senten}	15	
goto(50,))	{F2 -> (Cabecera).}	77	{F2 -> (Cabecera).}
goto(51, id)	{Cabecera -> Tipo id.K}	78	{Cabecera -> Tipo id.K; K -> ., Tipo id K; K -> .}
goto(53, ;)	{S -> id = E ;.}	79	{S -> id = E ;.}
goto(54, ;)	{S -> id %= E ;.}	80	{S -> id %= E ;.}
goto(55,))	{S -> id (Parametros).;}	81	{S -> id (Parametros).;}
goto(56, K2)	{Parametros -> E K2.}	82	{Parametros -> E K2.}
goto(56, ,)	{K2 -> ,.E K2}	83	{K2 -> ,.E K2; E -> .! V; E -> .R; R -> .R > U; R -> .R < U; R -> .U; U -> .U + V; U -> .U - V; U -> .V; V -> .id; V -> .(E); V -> .id (Parametros); V -> .ent; V -> .cadena}
goto(56,)	{K2 -> .}	84	{K2 -> .}
goto(58,))	{S -> alert (E).;}	85	{S -> alert (E).;}
goto(59,))	{S -> input (id).;}	86	{S -> input (id).;}
goto(62, U)	{R -> R > U.; U -> U.+ V; U -> U.- V}	87	{R -> R > U.; U -> U.+ V; U -> U.- V}
goto(62, V)	{U -> V.}	35	
goto(62, id)	{V -> id.; V -> id.(Parametros)}	36	
goto(62, ()	{V -> (.E)}	37	
goto(62, ent)	{V -> ent.}	38	
goto(62, cadena)	{V -> cadena.}	39	
goto(63, U)	{R -> R < U.; U -> U.+ V; U -> U.- V}	88	{R -> R < U.; U -> U.+ V; U -> U.- V}
goto(63, V)	{U -> V.}	35	
goto(63, id)	{V -> id.; V -> id.(Parametros)}	36	
goto(63, ()	{V -> (.E)}	37	
goto(63, ent)	{V -> ent.}	38	
goto(63, cadena)	{V -> cadena.}	39	
goto(64, V)	{U -> U + V.}	89	{U -> U + V.}
goto(64, id)	{V -> id.; V -> id.(Parametros)}	36	
goto(64, ()	{V -> (.E)}	37	
goto(64, ent)	{V -> ent.}	38	
goto(64, cadena)	{V -> cadena.}	39	
goto(65, V)	{U -> U - V.}	90	{U -> U - V.}
goto(65, id)	{V -> id.; V -> id.(Parametros)}	36	
goto(65, ()	{V -> (.E)}	37	
goto(65, ent)	{V -> ent.}	38	
goto(65, cadena)	{V -> cadena.}	39	
goto(66, Parametros)	{V -> id (Parametros.)}	91	{V -> id (Parametros.)}
goto(66, E)	{Parametros -> E.K2}	56	
goto(66,)	{Parametros -> .}	57	
goto(66, !)	{E -> !.V}	32	
goto(66, R)	{E -> R.; R -> R.> U; R -> R.< U}	33	
goto(66, U)	{R -> U.; U -> U.+ V; U -> U.- V}	34	
goto(66, V)	{U -> V.}	35	
goto(66, id)	{V -> id.; V -> id.(Parametros)}	36	
goto(66, ()	{V -> (.E)}	37	
goto(66, ent)	{V -> ent.}	38	
goto(66, cadena)	{V -> cadena.}	39	
goto(67,))	{V -> (E).}	92	{V -> (E).}
goto(70, Lista_Sentencias)	{Senten -> { Lista_Sentencias.}}	93	{Senten -> { Lista_Sentencias.}}
goto(70, Sentencia)	{Lista_Sentencias -> Sentencia.Lista_Sentencias}	75	
goto(70,)	{Lista_Sentencias -> .}	76	
goto(70, S)	{Sentencia -> S.}	5	
goto(70, IF_)	{Sentencia -> IF_.}	6	

SLR closure table

Goto	Kernel	State	Closure
goto(70, let)	{Sentencia -> let.Tipo id ;}	7	
goto(70, id)	{S -> id.= E ;; S -> id.%= E ;; S -> id.(Parametros) ;}	9	
goto(70, alert)	{S -> alert.(E) ;}	10	
goto(70, input)	{S -> input.(id) ;}	11	
goto(70, return)	{S -> return.X ;}	12	
goto(70, IF1)	{IF_ -> IF1.IF2}	13	
goto(70, if)	{IF1 -> if.(E) Senten}	15	
goto(72,))	{IF1 -> if (E).Senten}	94	{IF1 -> if (E).Senten; Senten -> .Sentencia; Senten -> .{ Lista_Sentencias }; Sentencia -> .S; Sentencia -> .IF_; Sentencia -> .let Tipo id ;; S -> .id = E ;; S -> .id %= E ;; S -> .id (Parametros) ;; S -> .alert (E) ;; S -> .input (id) ;; S -> .return X ;; IF_ -> .IF1 IF2; IF1 -> .if (E) Senten}
goto(74, })	{F3 -> { Lista_Sentencias }.}	95	{F3 -> { Lista_Sentencias }.}
goto(75, Lista_Sentencias)	{Lista_Sentencias -> Sentencia Lista_Sentencias.}	96	{Lista_Sentencias -> Sentencia Lista_Sentencias.}
goto(75, Sentencia)	{Lista_Sentencias -> Sentencia.Lista_Sentencias}	75	
goto(75,)	{Lista_Sentencias -> .}	76	
goto(75, S)	{Sentencia -> S.}	5	
goto(75, IF_)	{Sentencia -> IF_.}	6	
goto(75, let)	{Sentencia -> let.Tipo id ;}	7	
goto(75, id)	{S -> id.= E ;; S -> id.%= E ;; S -> id.(Parametros) ;}	9	
goto(75, alert)	{S -> alert.(E) ;}	10	
goto(75, input)	{S -> input.(id) ;}	11	
goto(75, return)	{S -> return.X ;}	12	
goto(75, IF1)	{IF_ -> IF1.IF2}	13	
goto(75, if)	{IF1 -> if.(E) Senten}	15	
goto(78, K)	{Cabecera -> Tipo id K.}	97	{Cabecera -> Tipo id K.}
goto(78, ,)	{K -> ,.Tipo id K}	98	{K -> ,.Tipo id K; Tipo -> .boolean; Tipo -> .number; Tipo -> .string}
goto(78,)	{K -> .}	99	{K -> .}
goto(81, ;)	{S -> id (Parametros) ;.}	100	{S -> id (Parametros) ;.}
goto(83, E)	{K2 -> , E.K2}	101	{K2 -> , E.K2; K2 -> ., E K2; K2 -> .}
goto(83, !)	{E -> !.V}	32	
goto(83, R)	{E -> R.; R -> R.> U; R -> R.< U}	33	
goto(83, U)	{R -> U.; U -> U.+ V; U -> U.- V}	34	
goto(83, V)	{U -> V.}	35	
goto(83, id)	{V -> id.; V -> id.(Parametros)}	36	
goto(83, ()	{V -> (.E)}	37	
goto(83, ent)	{V -> ent.}	38	
goto(83, cadena)	{V -> cadena.}	39	
goto(85, ;)	{S -> alert (E) ;.}	102	{S -> alert (E) ;.}
goto(86, ;)	{S -> input (id) ;.}	103	{S -> input (id) ;.}
goto(87, +)	{U -> U +.V}	64	
goto(87, -)	{U -> U -.V}	65	
goto(88, +)	{U -> U +.V}	64	
goto(88, -)	{U -> U -.V}	65	
goto(91,))	{V -> id (Parametros).}	104	{V -> id (Parametros).}
goto(93, })	{Senten -> { Lista_Sentencias }.}	105	{Senten -> { Lista_Sentencias }.}
goto(94, Senten)	{IF1 -> if (E) Senten.}	106	{IF1 -> if (E) Senten.}
goto(94, Sentencia)	{Senten -> Sentencia.}	69	
goto(94, {)	{Senten -> {.Lista_Sentencias }}	70	
goto(94, S)	{Sentencia -> S.}	5	
goto(94, IF_)	{Sentencia -> IF_.}	6	
goto(94, let)	{Sentencia -> let.Tipo id ;}	7	
goto(94, id)	{S -> id.= E ;; S -> id.%= E ;; S -> id.(Parametros) ;}	9	
goto(94, alert)	{S -> alert.(E) ;}	10	
goto(94, input)	{S -> input.(id) ;}	11	
goto(94, return)	{S -> return.X ;}	12	
goto(94, IF1)	{IF_ -> IF1.IF2}	13	
goto(94, if)	{IF1 -> if.(E) Senten}	15	
goto(98, Tipo)	{K -> , Tipo.id K}	107	{K -> , Tipo.id K}
goto(98, boolean)	{Tipo -> boolean.}	19	
goto(98, number)	{Tipo -> number.}	20	
goto(98, string)	{Tipo -> string.}	21	
goto(101, K2)	{K2 -> , E K2.}	108	{K2 -> , E K2.}
goto(101, ,)	{K2 -> ,.E K2}	83	
goto(101,)	{K2 -> .}	84	
goto(107, id)	{K -> , Tipo id.K}	109	{K -> , Tipo id.K; K -> ., Tipo id K; K -> .}
goto(109, K)	{K -> , Tipo id K.}	110	{K -> , Tipo id K.}
goto(109, ,)	{K -> ,.Tipo id K}	98	
goto(109,)	{K -> .}	99	

LR table

ACTION

State		let	id	;	if	(	)	else	{	}	=	%=	alert	input	return	,	function	boolean	number	string	!	>	<	+	-	ent	cadena	$
0	s4	s7	s9		s15								s10	s11	s12		s14											
1																												acc
2	s4	s7	s9		s15								s10	s11	s12		s14											
3	s4	s7	s9		s15								s10	s11	s12		s14											
4																												r_{3}
5	r_{4}	r_{4}	r_{4}		r_{4}			r_{4}					r_{4}	r_{4}	r_{4}		r_{4}											
6	r_{5}	r_{5}	r_{5}		r_{5}			r_{5}					r_{5}	r_{5}	r_{5}		r_{5}											
7																		s19	s20	s21								
8						s23																						
9						s26					s24	s25																
10						s27																						
11						s28																						
12	s31		s36			s37															s32					s38	s39	
13	s42							s41																				
14	s45																	s19	s20	s21								
15						s46																						
16																												r_{1}
17																												r_{2}
18			s47																									
19			r_{29}																									
20			r_{30}																									
21			r_{31}																									
22									s49																			
23	s52																	s19	s20	s21								
24			s36			s37															s32					s38	s39	
25			s36			s37															s32					s38	s39	
26	s57		s36			s37															s32					s38	s39	
27			s36			s37															s32					s38	s39	
28			s59																									
29				s60																								
30				r_{23}																								
31				r_{24}																								
32			s36			s37																				s38	s39	
33	r_{41}			r_{41}			r_{41}									r_{41}						s62	s63					
34	r_{44}			r_{44}			r_{44}									r_{44}						r_{44}	r_{44}	s64	s65			
35	r_{47}			r_{47}			r_{47}									r_{47}						r_{47}	r_{47}	r_{47}	r_{47}			
36	r_{48}			r_{48}		s66	r_{48}									r_{48}						r_{48}	r_{48}	r_{48}	r_{48}			
37			s36			s37															s32					s38	s39	
38	r_{51}			r_{51}			r_{51}									r_{51}						r_{51}	r_{51}	r_{51}	r_{51}			
39	r_{52}			r_{52}			r_{52}									r_{52}						r_{52}	r_{52}	r_{52}	r_{52}			
40	r_{7}	r_{7}	r_{7}		r_{7}			r_{7}					r_{7}	r_{7}	r_{7}		r_{7}											
41		s7	s9		s15				s70				s10	s11	s12													
42	r_{10}	r_{10}	r_{10}		r_{10}			r_{10}					r_{10}	r_{10}	r_{10}		r_{10}											
43			s71																									
44			r_{32}																									
45			r_{33}																									
46			s36			s37															s32					s38	s39	
47				s73																								
48	r_{25}	r_{25}	r_{25}		r_{25}								r_{25}	r_{25}	r_{25}		r_{25}											
49	s76	s7	s9		s15								s10	s11	s12													
50							s77																					
51			s78																									
52							r_{35}																					
53				s79																								
54				s80																								

GOTO

State	Axioma_	Axioma	Sentencia	IF_	IF1	IF2	Senten	S	Parametros	K2	X	Funcion	F1	F2	F3	Tipo	Tipo_B	Cabecera	K	Lista_Sentencias	E	R	U	V
0		1	2	6	13			5				3	8											
1																								
2		16	2	6	13			5				3	8											
3		17	2	6	13			5				3	8											
4																								
5																								
6																								
7																18								
8														22										
9																								
10																								
11																								
12											29										30	33	34	35
13						40																		
14																44	43							
15																								
16																								
17																								
18																								
19																								
20																								
21																								
22															48									
23																51		50						
24																					53	33	34	35
25																					54	33	34	35
26									55												56	33	34	35
27																					58	33	34	35
28																								
29																								
30																								
31																								
32																								61
33																								
34																								
35																								
36																								
37																					67	33	34	35
38																								
39																								
40																								
41			69	6	13		68	5																
42																								
43																								
44																								
45																								
46																					72	33	34	35
47																								
48																								
49			75	6	13			5												74				
50																								
51																								
52																								
53																								
54																								

Input (tokens): id + id * id

Maximum number of steps: 100

PARSE

Trace				Tree
Step	Stack	Input	Action	
1	0	id + id * id $	s9	
2	0 id 9	+ id * id $		

LR table

State	ACTION																											
		let	id	;	if	(	)	else	{	}	=	%=	alert	input	return	,	function	boolean	number	string	!	>	<	+	-	ent	cadena	$
55							s81																					
56	s84															s83												
57							r_{20}																					
58							s85																					
59							s86																					
60	r_{18}	r_{18}	r_{18}		r_{18}			r_{18}					r_{18}	r_{18}	r_{18}		r_{18}											
61	r_{40}			r_{40}			r_{40}									r_{40}												
62			s36			s37																				s38	s39	
63			s36			s37																				s38	s39	
64			s36			s37																				s38	s39	
65			s36			s37																				s38	s39	
66	s57		s36			s37															s32					s38	s39	
67							s92																					
68	r_{9}	r_{9}	r_{9}		r_{9}			r_{9}					r_{9}	r_{9}	r_{9}		r_{9}											
69	r_{11}	r_{11}	r_{11}		r_{11}			r_{11}					r_{11}	r_{11}	r_{11}		r_{11}											
70	s76	s7	s9		s15								s10	s11	s12													
71						r_{26}																						
72							s94																					
73	r_{6}	r_{6}	r_{6}		r_{6}			r_{6}					r_{6}	r_{6}	r_{6}		r_{6}											
74										s95																		
75	s76	s7	s9		s15								s10	s11	s12													
76										r_{39}																		
77									r_{27}																			
78	s99															s98												
79	r_{13}	r_{13}	r_{13}		r_{13}			r_{13}					r_{13}	r_{13}	r_{13}		r_{13}											
80	r_{14}	r_{14}	r_{14}		r_{14}			r_{14}					r_{14}	r_{14}	r_{14}		r_{14}											
81				s100																								
82							r_{19}																					
83			s36			s37															s32					s38	s39	
84							r_{22}																					
85				s102																								
86				s103																								
87	r_{42}			r_{42}			r_{42}									r_{42}						r_{42}	r_{42}	s64	s65			
88	r_{43}			r_{43}			r_{43}									r_{43}						r_{43}	r_{43}	s64	s65			
89	r_{45}			r_{45}			r_{45}									r_{45}						r_{45}	r_{45}	r_{45}	r_{45}			
90	r_{46}			r_{46}			r_{46}									r_{46}						r_{46}	r_{46}	r_{46}	r_{46}			
91							s104																					
92	r_{49}			r_{49}			r_{49}									r_{49}						r_{49}	r_{49}	r_{49}	r_{49}			
93										s105																		
94		s7	s9		s15				s70				s10	s11	s12													
95	r_{28}	r_{28}	r_{28}		r_{28}								r_{28}	r_{28}	r_{28}		r_{28}											
96										r_{38}																		
97							r_{34}																					
98																		s19	s20	s21								
99							r_{37}																					
100	r_{15}	r_{15}	r_{15}		r_{15}			r_{15}					r_{15}	r_{15}	r_{15}		r_{15}											
101	s84															s83												
102	r_{16}	r_{16}	r_{16}		r_{16}			r_{16}					r_{16}	r_{16}	r_{16}		r_{16}											
103	r_{17}	r_{17}	r_{17}		r_{17}			r_{17}					r_{17}	r_{17}	r_{17}		r_{17}											
104	r_{50}			r_{50}			r_{50}									r_{50}						r_{50}	r_{50}	r_{50}	r_{50}			
105	r_{12}	r_{12}	r_{12}		r_{12}			r_{12}					r_{12}	r_{12}	r_{12}		r_{12}											
106	r_{8}							r_{8}																				
107			s109																									
108							r_{21}																					
109	s99															s98												
110							r_{36}																					

State	GOTO																							
	Axioma_	Axioma	Sentencia	IF_	IF1	IF2	Senten	S	Parametros	K2	X	Funcion	F1	F2	F3	Tipo	Tipo_B	Cabecera	K	Lista_Sentencias	E	R	U	V
55																								
56										82														
57																								
58																								
59																								
60																								
61																								
62																							87	35
63																							88	35
64																								89
65																								90
66									91												56	33	34	35
67																								
68																								
69																								
70			75	6	13			5												93				
71																								
72																								
73																								
74																								
75			75	6	13			5												96				
76																								
77																								
78																			97					
79																								
80																								
81																								
82																								
83																					101	33	34	35
84																								
85																								
86																								
87																								
88																								
89																								
90																								
91																								
92																								
93																								
94			69	6	13		106	5																
95																								
96																								
97																								
98																107								
99																								
100																								
101										108														
102																								
103																								
104																								
105																								
106																								
107																								
108																								
109																			110					
110																								