

Doc#	Array#	Document Productions (or's removed)
0	0	PGM → program id ; DECLARATIONS SUBPGM_DCLS CMPD_STMT .
1	1	ID_LIST → id ID_LIST_END
2	2	ID_LIST_END → , ID_LIST
	3	ID_LIST_END → €
3	4	DECLARATIONS → var ID_LIST : DTYPE ; DECLARATIONS
	5	DECLARATIONS → €
4	6	TYPE → array [int_lit .. int_lit] of STD_TYPE
	7	TYPE → STD_TYPE
5	8	STD_TYPE → integer
	9	STD_TYPE → real
	10	STD_TYPE → string
6	11	SUBPGM_DCLS → SUBPGM_DCL ; SUBPGM_DCLS
	12	SUBPGM_DCLS → €
7	13	SUBPGM_DCL → SUBPGM_HEAD DECLARATIONS CMPD_STMT
8	14	SUBPGM_HEAD → function id ARGS : STD_TYPE ;
	15	SUBPGM_HEAD → procedure id ARGS ;
9	16	ARGS → (PARM_LIST)
	17	ARGS → €
10	18	PARM_LIST → ID_LIST : TYPE PARM_LIST_END
11	19	PARM_LIST_END → ; PARM_LIST
	20	PARM_LIST_END → €
12	21	CMPD_STMT → begin OPT_STMTS end
13	22	OPT_STMTS → €
	23	OPT_STMTS → STMT_LIST
14	24	STMT_LIST → STMT STMT_LIST_END
15	25	STMT_LIST_END → ; STMT_LIST
	26	STMT_LIST_END → €
16	27	STMT → id PROC_OR_ASSIGN
	28	STMT → CMPD_STMT
	29	STMT → if EXPR then STMT else STMT
	30	STMT → while EXPR do STMT
	31	STMT → for VARIABLE := EXPR END_FOR
17	32	PROC_OR_ASSIGN → PROC_STMT_END
	33	PROC_OR_ASSIGN → VAR_END := EXPR
18	34	END_FOR → to EXPR do
	35	END_FOR → downto EXPR do
19	36	VARIABLE → id VAR_END
20	37	VAR_END → [EXPR]
	38	VAR_END → €
21	39	PROC_STMT_END → (EXPR_LIST)
	40	PROC_STMT_END → €
22	41	EXPR_LIST → EXPR EXPR_LIST_END
23	42	EXPR_LIST_END → , EXPR_LIST
	43	EXPR_LIST_END → €
24	44	EXPR → SIMPLE_EXPR EXPR_END
25	45	EXPR_END → RELOP SIMPLE_EXPR

	46	EXPR_END \rightarrow RELOP SIMPLE_EXPR
	47	EXPR_END $\rightarrow \epsilon$
26	48	RELOP $\rightarrow <$
	49	RELOP $\rightarrow <=$
	50	RELOP $\rightarrow >$
	51	RELOP $\rightarrow >=$
	52	RELOP $\rightarrow <>$
	53	RELOP $\rightarrow =$
27	54	SIMPLE_EXPR \rightarrow SIGN TERM SIMPLE_EXPR_END
	55	SIMPLE_EXPR \rightarrow TERM SIMPLE_EXPR_END
28	56	SIMPLE_EXPR_END \rightarrow ADD_OP SIMPLE_EXPR
	57	SIMPLE_EXPR_END $\rightarrow \epsilon$
29	58	ADD_OP $\rightarrow +$
	59	ADD_OP $\rightarrow -$
30	60	TERM \rightarrow FACTOR TERM_END
31	61	TERM_END \rightarrow MUL_OP TERM
	62	TERM_END $\rightarrow \epsilon$
32	63	MUL_OP $\rightarrow /$
	64	MUL_OP $\rightarrow *$
33	65	FACTOR \rightarrow id FACTOR_END
	66	FACTOR \rightarrow int_lit
	67	FACTOR \rightarrow real_lit
	68	FACTOR $\rightarrow ($ EXPR $)$
	69	FACTOR \rightarrow not FACTOR
34	70	FACTOR_END $\rightarrow ($ EXPR_LIST $)$
	71	FACTOR_END $\rightarrow \epsilon$
35	72	SIGN $\rightarrow +$
	73	SIGN $\rightarrow -$

Excel Parse (epsilon c				
PGM	→	program	id	;
ID_LIST	→	id	ID_LIST_END	
ID_LIST_END	→	,	ID_LIST	
ID_LIST_END	→			
DECLARATIONS	→	var	ID_LIST	:
DECLARATIONS	→	E		
DTYPE	→	array	[int_lit
DTYPE	→	STD_TYPE		
STD_TYPE	→	integer		
STD_TYPE	→	real		
STD_TYPE	→	string		
SUBPGM_DCLS	→	SUBPGM_DCL	;	SUBPGM_DCLS
SUBPGM_DCLS	→	E		
SUBPGM_DCL	→	SUBPGM_HEAD	DECLARATIONS	CMPD_STMT
SUBPGM_HEAD	→	function	id	ARGS
SUBPGM_HEAD	→	procedure	id	ARGS
ARGS	→	(PARAM_LIST)
ARGS	→	E		
PARAM_LIST	→	ID_LIST	:	DTYPE
PARAM_LIST_END	→	;	PARAM_LIST	
PARAM_LIST_END	→	E		
CMPD_STMT	→	begin	OPT_STMTS	end
OPT_STMTS	→	E		
OPT_STMTS	→	STMT_LIST		
STMT_LIST	→	STMT	STMT_LIST_END	
STMT_LIST_END	→	;	STMT_LIST	
STMT_LIST_END	→	E		
STMT	→	id	PROC_OR_ASSIGN	
STMT	→	CMPD_STMT		
STMT	→	if	EXPR	then
STMT	→	while	EXPR	do
STMT	→	for	VARIABLE	:=
PROC_OR_ASSIGN	→	PROC_STMT_END		
PROC_OR_ASSIGN	→	VAR_END	:=	EXPR
END_FOR	→	to	EXPR	do
END_FOR	→	downto	EXPR	do
VARIABLE	→	id	VAR_END	
VAR_END	→	[EXPR]
VAR_END	→	E		
PROC_STMT_END	→	(EXPR_LIST)
PROC_STMT_END	→	E		
EXPR_LIST	→	EXPR	EXPR_LIST_END	
EXPR_LIST_END	→	,	EXPR_LIST	
EXPR_LIST_END	→	E		
EXPR	→	SIMPLE_EXPR	EXPR_END	
EXPR_END	→	RELOP	SIMPLE_EXPR	

EXPR_END	→	RELOP	SIMPLE_EXPR	
EXPR_END	→	E		
RELOP	→	<		
RELOP	→	<=		
RELOP	→	>		
RELOP	→	>=		
RELOP	→	<>		
RELOP	→	=		
SIMPLE_EXPR	→	SIGN	TERM	SIMPLE_EXPR_END
SIMPLE_EXPR	→	TERM	SIMPLE_EXPR_END	
SIMPLE_EXPR_END	→	ADD_OP	SIMPLE_EXPR	
SIMPLE_EXPR_END	→	E		
ADD_OP	→	+		
ADD_OP	→	-		
TERM	→	FACTOR	TERM_END	
TERM_END	→	MUL_OP	TERM	
TERM_END	→	E		
MUL_OP	→	/		
MUL_OP	→	*		
FACTOR	→	id	FACTOR_END	
FACTOR	→	int_lit		
FACTOR	→	real_lit		
FACTOR	→	(EXPR)
FACTOR	→	not	FACTOR	
FACTOR_END	→	(EXPR_LIST)
FACTOR_END	→	E		
SIGN	→	+		
SIGN	→	-		

changed to E)					
DECLARATIONS	SUBPGM_DCLS	CMPD_STMT	.		program
					id
					,
					E
DTYPE	;	DECLARATIONS			var
					E
..	int_lit]	of	STD_TYPE	array
					E
					integer
					real
					string
					function
					E
					function
:	STD_TYPE	;			function
;					procedure
					(
					E
PARM_LIST_END					E
					;
					E
					E
					end
					E
					E
					;
					E
					id
					begin
STMT	else	STMT			if
STMT					while
EXPR	END_FOR				for
					(
					E
					to
					downto
					E
					[
					E
					(
					E
					E
					,
					E
					E
					<

					>
					E
					<
					<=
					>
					>=
					<>
					=
					+
					E
					+
					E
					+
					-
					E
					*
					E
					/
					*
					id
					int_lit
					real_lit
					(
					not
					(
					E
					+
					-

Select1 Set		Produ			
		NT_PROGRAM	TOK_PROGRAM	TOK_IDENT	TOK_SEMIC
		NT_ID_LIST	TOK_IDENT	NT_ID_LIST_END	E
		NT_ID_LIST_END	TOK_COMMA	NT_ID_LIST	E
		NT_ID_LIST_END	E	E	E
		NT_DECLARATION	TOK_VAR	NT_ID_LIST	TOK_COLON
		NT_DECLARATION	E	E	E
		NT_TYPE	TOK_ARRAY	TOK_LBRKT	TOK_INT_LIT
		NT_TYPE	NT_STD_TYPE	E	E
		NT_STD_TYPE	TOK_INTEGER	E	E
		NT_STD_TYPE	TOK_REAL	E	E
		NT_STD_TYPE	TOK_STRING	E	E
procedure		NT_SUBPGM_DCL	NT_SUBPGM_DCL	TOK_SEMIC	NT_SUBPGM_DCLS
		NT_SUBPGM_DCL	E	E	E
procedure		NT_SUBPGM_DCL	NT_SUBPGM_HEAD	NT_DECLARATIONS	NT_CMPD_STMT
		NT_SUBPGM_HEAD	TOK_FUNCTION	TOK_IDENT	NT_ARGS
		NT_SUBPGM_HEAD	TOK_PROCEDURE	TOK_IDENT	NT_ARGS
		NT_ARGS	TOK_LEFTPAR	NT_PARM_LIST	TOK_RIGHTPAR
		NT_ARGS	E	E	E
		NT_PARM_LIST	NT_ID_LIST	TOK_COLON	NT_TYPE
		NT_PARM_LIST_E	TOK_SEMIC	NT_PARM_LIST	E
		NT_PARM_LIST_E	E	E	E
		NT_CMPD_STMT	TOK_BEGIN	NT_OPT_STMTS	TOK_END
		NT_OPT_STMTS	E	E	E
		NT_OPT_STMTS	NT_STMT_LIST	E	E
		NT_STMT_LIST	NT_STMT	NT_STMT_LIST_END	E
		NT_STMT_LIST_END	TOK_SEMIC	NT_STMT_LIST	E
		NT_STMT_LIST_END	E	E	E
		NT_STMT	TOK_IDENT	NT_PROC_OR_ASSIGN	E
		NT_STMT	NT_CMPD_STMT	E	E
		NT_STMT	TOK_IF	NT_EXPR	TOK_THEN
		NT_STMT	TOK_WHILE	NT_EXPR	TOK_DO
		NT_STMT	TOK_FOR	NT_VARIABLE	TOK_ASSIGN
;		NT_PROC_OR_ASSIGN	NT_PROC_STMT_END	E	E
		NT_PROC_OR_ASSIGN	NT_VAR_END	TOK_ASSIGN	NT_EXPR
		NT_END_FOR	TOK_TO	NT_EXPR	TOK_DO
		NT_END_FOR	TOK_DOWNT0	NT_EXPR	TOK_DO
		NT_VARIABLE	TOK_IDENT	NT_VAR_END	E
		NT_VAR_END	TOK_LBRKT	NT_EXPR	TOK_RBRKT
		NT_VAR_END	E	E	E
		NT_PROC_STMT	TOK_LEFTPAR	NT_EXPR_LIST	TOK_RIGHTPAR
		NT_PROC_STMT	E	E	E
		NT_EXPR_LIST	NT_EXPR	NT_EXPR_LIST_END	E
		NT_EXPR_LIST_END	TOK_COMMA	NT_EXPR_LIST	E
		NT_EXPR_LIST_END	E	E	E
		NT_EXPR	NT_SIMPLE_EXPR	NT_EXPR_END	E
<=	=	NT_EXPR_END	NT_RELOP	NT_SIMPLE_EXPR	E

>=	<>	NT_EXPR_END	NT_RELOP	NT_SIMPLE_EXPR	E
		NT_EXPR_END	E	E	E
		NT_RELOP	TOK_LTHAN	E	E
		NT_RELOP	TOK_LT_EQ	E	E
		NT_RELOP	TOK_GTHAN	E	E
		NT_RELOP	TOK_GT_EQ	E	E
		NT_RELOP	TOK_NOT_EQ	E	E
		NT_RELOP	TOK_EQUAL	E	E
-		NT_SIMPLE_EXPR	NT_SIGN	NT_TERM	NT_SIMPLE_EXPR_END
		NT_SIMPLE_EXPR	NT_TERM	NT_SIMPLE_EXPR_END	E
-		NT_SIMPLE_EXPR	NT_ADD_OP	NT_SIMPLE_EXPR	E
		NT_SIMPLE_EXPR	E	E	E
		NT_ADD_OP	TOK_PLUS	E	E
		NT_ADD_OP	TOK_MINUS	E	E
		NT_TERM	NT_FACTOR	NT_TERM_END	E
/		NT_TERM_END	NT_MUL_OP	NT_TERM	E
		NT_TERM_END	E	E	E
		NT_MUL_OP	TOK_SLASH	E	E
		NT_MUL_OP	TOK_CHAR_LIT	E	E
		NT_FACTOR	TOK_IDENT	NT_FACTOR_END	E
		NT_FACTOR	TOK_INT_LIT	E	E
		NT_FACTOR	TOK_REAL_LIT	E	E
		NT_FACTOR	TOK_LEFTPAR	NT_EXPR	TOK_RIGHTPAR
		NT_FACTOR	TOK_NOT	NT_FACTOR	E
		NT_FACTOR_END	TOK_LEFTPAR	NT_EXPR_LIST	TOK_RIGHTPAR
		NT_FACTOR_END	E	E	E
		NT_SIGN	TOK_PLUS	E	E
		NT_SIGN	TOK_MINUS	E	E

Functions with Constant Translations

[illegible]

[illegible]

	Select Set with Constant Translation		
E	TOK_PROGRAM	E	E
E	TOK_IDENT	E	E
E	TOK_COMMA	E	E
E	E	E	E
E	TOK_VAR	E	E
E	E	E	E
NT_STD_TYPE	TOK_ARRAY	E	E
E	E	E	E
E	TOK_INTEGER	E	E
E	TOK_REAL	E	E
E	TOK_STRING	E	E
E	TOK_FUNCTION	TOK_PROCEDURE	E
E	E	E	E
E	TOK_FUNCTION	TOK_PROCEDURE	E
E	TOK_FUNCTION	E	E
E	TOK_PROCEDURE	E	E
E	TOK_LEFTPAR	E	E
E	E	E	E
E	E	E	E
E	TOK_SEMIC	E	E
E	E	E	E
E	E	E	E
E	TOK_END	E	E
E	E	E	E
E	E	E	E
E	TOK_SEMIC	E	E
E	E	E	E
E	TOK_IDENT	E	E
E	TOK_BEGIN	E	E
E	TOK_IF	E	E
E	TOK_WHILE	E	E
E	TOK_FOR	E	E
E	TOK_LEFTPAR	TOK_SEMIC	E
E	E	E	E
E	TOK_TO	E	E
E	TOK_DOWNT0	E	E
E	E	E	E
E	TOK_LBRKT	E	E
E	E	E	E
E	TOK_LEFTPAR	E	E
E	E	E	E
E	E	E	E
E	TOK_COMMA	E	E
E	E	E	E
E	E	E	E
E	TOK_LTHAN	TOK_LT_EQ	TOK_EQUAL

E	TOK_GTHAN	TOK_GT_EQ	TOK_NOT_EQ
E	E	E	E
E	TOK_LTHAN	E	E
E	TOK_LT_EQ	E	E
E	TOK_GTHAN	E	E
E	TOK_GT_EQ	E	E
E	TOK_NOT_EQ	E	E
E	TOK_EQUAL	E	E
E	TOK_PLUS	TOK_MINUS	E
E	E	E	E
E	TOK_PLUS	TOK_MINUS	E
E	E	E	E
E	TOK_PLUS	E	E
E	TOK_MINUS	E	E
E	E	E	E
E	TOK_CHAR_LIT	TOK_SLASH	E
E	E	E	E
E	TOK_SLASH	E	E
E	TOK_CHAR_LIT	E	E
E	TOK_IDENT	E	E
E	TOK_INT_LIT	E	E
E	TOK_REAL_LIT	E	E
E	TOK_LEFTPAR	E	E
E	TOK_NOT	E	E
E	TOK_LEFTPAR	E	E
E	E	E	E
E	TOK_PLUS	E	E
E	TOK_MINUS	E	E

Rows of C++ 2D Array 0 = LHS, 1-8 = RHS, 9-11 = Select Set

{NT_PROGRAM,TOK_PROGRAM, TOK_IDENT, TOK_SEMIC, NT_DECLARATIONS, NT_SUBPGM_DCLS, NT_CMPD_STMT,
{NT_ID_LIST,TOK_IDENT, NT_ID_LIST_END, E, E, E, E, E, E, TOK_IDENT, E, E},
{NT_ID_LIST_END,TOK_COMMA, NT_ID_LIST, E, E, E, E, E, E, TOK_COMMA, E, E},
{NT_ID_LIST_END,E, E, E, E, E, E, E, E, E, E},
{NT_DECLARATIONS,TOK_VAR, NT_ID_LIST, TOK_COLON, NT_TYPE, TOK_SEMIC, NT_DECLARATIONS, E, E, TOK_VAR,
{NT_DECLARATIONS,E, E, E, E, E, E, E, E, E, E},
{NT_TYPE,TOK_ARRAY, TOK_LBRKT, TOK_INT_LIT, TOK_DOTDOT, TOK_INT_LIT, TOK_RBRKT, TOK_OF, NT_STD_TYPE,
{NT_TYPE,NT_STD_TYPE, E, E, E, E, E, E, E, E, E},
{NT_STD_TYPE,TOK_INTEGER, E, E, E, E, E, E, E, TOK_INTEGER, E, E},
{NT_STD_TYPE,TOK_REAL, E, E, E, E, E, E, E, TOK_REAL, E, E},
{NT_STD_TYPE,TOK_STRING, E, E, E, E, E, E, E, TOK_STRING, E, E},
{NT_SUBPGM_DCLS,NT_SUBPGM_DCL, TOK_SEMIC, NT_SUBPGM_DCLS, E, E, E, E, E, TOK_FUNCTION, TOK_PROC,
{NT_SUBPGM_DCLS,E, E, E, E, E, E, E, E, E, E},
{NT_SUBPGM_DCL,NT_SUBPGM_HEAD, NT_DECLARATIONS, NT_CMPD_STMT, E, E, E, E, E, TOK_FUNCTION, TOK,
{NT_SUBPGM_HEAD,TOK_FUNCTION, TOK_IDENT, NT_ARGS, TOK_COLON, NT_STD_TYPE, TOK_SEMIC, E, E, TOK,
{NT_SUBPGM_HEAD,TOK_PROCEDURE, TOK_IDENT, NT_ARGS, TOK_SEMIC, E, E, E, E, TOK_PROCEDURE, E, E},
{NT_ARGS,TOK_LEFTPAR, NT_PARM_LIST, TOK_RIGHTPAR, E, E, E, E, E, TOK_LEFTPAR, E, E},
{NT_ARGS,E, E, E, E, E, E, E, E, E, E},
{NT_PARM_LIST,NT_ID_LIST, TOK_COLON, NT_TYPE, NT_PARM_LIST_END, E, E, E, E, E, E},
{NT_PARM_LIST_END,TOK_SEMIC, NT_PARM_LIST, E, E, E, E, E, E, TOK_SEMIC, E, E},
{NT_PARM_LIST_END,E, E, E, E, E, E, E, E, E, E},
{NT_CMPD_STMT,TOK_BEGIN, NT_OPT_STMTS, TOK_END, E, E, E, E, E, E, E},
{NT_OPT_STMTS,E, E, E, E, E, E, E, E, TOK_END, E, E},
{NT_OPT_STMTS,NT_STMT_LIST, E, E, E, E, E, E, E, E, E},
{NT_STMT_LIST,NT_STMT, NT_STMT_LIST_END, E, E, E, E, E, E, E, E},
{NT_STMT_LIST_END,TOK_SEMIC, NT_STMT_LIST, E, E, E, E, E, E, TOK_SEMIC, E, E},
{NT_STMT_LIST_END,E, E, E, E, E, E, E, E, E, E},
{NT_STMT,TOK_IDENT, NT_PROC_OR_ASSIGN, E, E, E, E, E, E, TOK_IDENT, E, E},
{NT_STMT,NT_CMPD_STMT, E, E, E, E, E, E, E, TOK_BEGIN, E, E},
{NT_STMT,TOK_IF, NT_EXPR, TOK_THEN, NT_STMT, TOK_ELSE, NT_STMT, E, E, TOK_IF, E, E},
{NT_STMT,TOK_WHILE, NT_EXPR, TOK_DO, NT_STMT, E, E, E, E, TOK_WHILE, E, E},
{NT_STMT,TOK_FOR, NT_VARIABLE, TOK_ASSIGN, NT_EXPR, NT_END_FOR, E, E, E, TOK_FOR, E, E},
{NT_PROC_OR_ASSIGN,NT_PROC_STMT_END, E, E, E, E, E, E, E, TOK_LEFTPAR, TOK_SEMIC, E},
{NT_PROC_OR_ASSIGN,NT_VAR_END, TOK_ASSIGN, NT_EXPR, E, E, E, E, E, E, E},
{NT_END_FOR,TOK_TO, NT_EXPR, TOK_DO, E, E, E, E, E, TOK_TO, E, E},
{NT_END_FOR,TOK_DOWNT, NT_EXPR, TOK_DO, E, E, E, E, E, TOK_DOWNT, E, E},
{NT_VARIABLE,TOK_IDENT, NT_VAR_END, E, E, E, E, E, E, E, E},
{NT_VAR_END,TOK_LBRKT, NT_EXPR, TOK_RBRKT, E, E, E, E, E, TOK_LBRKT, E, E},
{NT_VAR_END,E, E, E, E, E, E, E, E, E, E},
{NT_PROC_STMT_END,TOK_LEFTPAR, NT_EXPR_LIST, TOK_RIGHTPAR, E, E, E, E, E, TOK_LEFTPAR, E, E},
{NT_PROC_STMT_END,E, E, E, E, E, E, E, E, E, E},
{NT_EXPR_LIST,NT_EXPR, NT_EXPR_LIST_END, E, E, E, E, E, E, E, E},
{NT_EXPR_LIST_END,TOK_COMMA, NT_EXPR_LIST, E, E, E, E, E, E, TOK_COMMA, E, E},
{NT_EXPR_LIST_END,E, E, E, E, E, E, E, E, E, E},
{NT_EXPR,NT_SIMPLE_EXPR, NT_EXPR_END, E, E, E, E, E, E, E, E},
{NT_EXPR_END,NT_RELOP, NT_SIMPLE_EXPR, E, E, E, E, E, E, TOK_LTHAN, TOK_LT_EQ, TOK_EQUAL},

{NT_EXPR_END,NT_RELOP, NT_SIMPLE_EXPR, E, E, E, E, E, E, TOK_GTHAN, TOK_GT_EQ, TOK_NOT_EQ},
{NT_EXPR_END,E, E, E, E, E, E, E, E, E, E},
{NT_RELOP,TOK_LTHAN, E, E, E, E, E, E, E, TOK_LTHAN, E, E},
{NT_RELOP,TOK_LT_EQ, E, E, E, E, E, E, E, TOK_LT_EQ, E, E},
{NT_RELOP,TOK_GTHAN, E, E, E, E, E, E, E, TOK_GTHAN, E, E},
{NT_RELOP,TOK_GT_EQ, E, E, E, E, E, E, E, TOK_GT_EQ, E, E},
{NT_RELOP,TOK_NOT_EQ, E, E, E, E, E, E, E, TOK_NOT_EQ, E, E},
{NT_RELOP,TOK_EQUAL, E, E, E, E, E, E, E, TOK_EQUAL, E, E},
{NT_SIMPLE_EXPR,NT_SIGN, NT_TERM, NT_SIMPLE_EXPR_END, E, E, E, E, E, TOK_PLUS, TOK_MINUS, E},
{NT_SIMPLE_EXPR,NT_TERM, NT_SIMPLE_EXPR_END, E, E, E, E, E, E, E, E, E},
{NT_SIMPLE_EXPR_END,NT_ADD_OP, NT_SIMPLE_EXPR, E, E, E, E, E, E, TOK_PLUS, TOK_MINUS, E},
{NT_SIMPLE_EXPR_END,E, E, E, E, E, E, E, E, E, E},
{NT_ADD_OP,TOK_PLUS, E, E, E, E, E, E, E, TOK_PLUS, E, E},
{NT_ADD_OP,TOK_MINUS, E, E, E, E, E, E, E, TOK_MINUS, E, E},
{NT_TERM,NT_FACTOR, NT_TERM_END, E, E, E, E, E, E, E, E, E},
{NT_TERM_END,NT_MUL_OP, NT_TERM, E, E, E, E, E, E, TOK_CHAR_LIT, TOK_SLASH, E},
{NT_TERM_END,E, E, E, E, E, E, E, E, E, E},
{NT_MUL_OP,TOK_SLASH, E, E, E, E, E, E, E, TOK_SLASH, E, E},
{NT_MUL_OP,TOK_CHAR_LIT, E, E, E, E, E, E, E, TOK_CHAR_LIT, E, E},
{NT_FACTOR,TOK_IDENT, NT_FACTOR_END, E, E, E, E, E, E, TOK_IDENT, E, E},
{NT_FACTOR,TOK_INT_LIT, E, E, E, E, E, E, E, TOK_INT_LIT, E, E},
{NT_FACTOR,TOK_REAL_LIT, E, E, E, E, E, E, E, TOK_REAL_LIT, E, E},
{NT_FACTOR,TOK_LEFTPAR, NT_EXPR, TOK_RIGHTPAR, E, E, E, E, E, TOK_LEFTPAR, E, E},
{NT_FACTOR,TOK_NOT, NT_FACTOR, E, E, E, E, E, E, TOK_NOT, E, E},
{NT_FACTOR_END,TOK_LEFTPAR, NT_EXPR_LIST, TOK_RIGHTPAR, E, E, E, E, E, TOK_LEFTPAR, E, E},
{NT_FACTOR_END,E, E, E, E, E, E, E, E, E, E},
{NT_SIGN,TOK_PLUS, E, E, E, E, E, E, E, TOK_PLUS, E, E},
{NT_SIGN,TOK_MINUS, E, E, E, E, E, E, E, TOK_MINUS, E, E},

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
"<STMT>",
"<STMT>",
"<STMT>",
"<STMT>",
"<STMT>",
"<STMT>", "<PROC_OR_ASSIGN>",
"<STMT>", "<PROC_OR_ASSIGN>",
"<STMT>", "<PROC_OR_ASSIGN>", "<END_FOR>",
"<STMT>", "<PROC_OR_ASSIGN>", "<END_FOR>",
"<STMT>", "<PROC_OR_ASSIGN>", "<END_FOR>", "<VARIABLE>",
"<STMT>", "<PROC_OR_ASSIGN>", "<END_FOR>", "<VARIABLE>", "<VAR_END>",
"<STMT>", "<PROC_OR_ASSIGN>", "<END_FOR>", "<VARIABLE>", "<VAR_END>",
"<STMT>", "<PROC_OR_ASSIGN>", "<END_FOR>", "<VARIABLE>", "<VAR_END>", "<PROC_STMT_EI",
"<STMT>", "<PROC_OR_ASSIGN>", "<END_FOR>", "<VARIABLE>", "<VAR_END>", "<PROC_STMT_EI",
"<STMT>", "<PROC_OR_ASSIGN>", "<END_FOR>", "<VARIABLE>", "<VAR_END>", "<PROC_STMT_EI",
"<STMT>", "<PROC_OR_ASSIGN>", "<END_FOR>", "<VARIABLE>", "<VAR_END>", "<PROC_STMT_EI",
"<STMT>", "<PROC_OR_ASSIGN>", "<END_FOR>", "<VARIABLE>", "<VAR_END>", "<PROC_STMT_EI",
"<STMT>", "<PROC_OR_ASSIGN>", "<END_FOR>", "<VARIABLE>", "<VAR_END>", "<PROC_STMT_EI",
"<STMT>", "<PROC OR ASSIGN>", "<END FOR>", "<VARIABLE>", "<VAR END>", "<PROC STMT EI
```

[illegible]

ND>,"
ND>,"
ND>," "<EXPR_LIST>,"
ND>," "<EXPR_LIST>," "<EXPR_LIST_END>,"
ND>," "<EXPR_LIST>," "<EXPR_LIST_END>,"
ND>," "<EXPR_LIST>," "<EXPR_LIST_END>," "<EXPR>,"
ND>," "<EXPR_LIST>," "<EXPR_LIST_END>," "<EXPR>," "<EXPR_END>,"

[illegible]

[illegible]

"<FACTOR_END>",
"<FACTOR_END>",
"<FACTOR_END>", "<SIGN>",