

In Grammar	Constant Name	Value	C++ declaration
char_lit	TOK_CHAR_LIT	-6	const int TOK_CHAR_LIT = -6;
real_lit	TOK_REAL_LIT	-5	const int TOK_REAL_LIT = -5;
int_lit	TOK_INT_LIT	-4	const int TOK_INT_LIT = -4;
id	TOK_IDENT	-3	const int TOK_IDENT = -3;
	TOK_ENDSRC	-2	const int TOK_ENDSRC = -2;
	TOK_SCANERR	-1	const int TOK_SCANERR = -1;
and	TOK_AND	0	const int TOK_AND = 0;
array	TOK_ARRAY	1	const int TOK_ARRAY = 1;
begin	TOK_BEGIN	2	const int TOK_BEGIN = 2;
boolean	TOK_BOOLEAN	3	const int TOK_BOOLEAN = 3;
case	TOK_CASE	4	const int TOK_CASE = 4;
char	TOK_CHAR	5	const int TOK_CHAR = 5;
const	TOK_CONST	6	const int TOK_CONST = 6;
div	TOK_DIV	7	const int TOK_DIV = 7;
do	TOK_DO	8	const int TOK_DO = 8;
downto	TOK_DOWNT0	9	const int TOK_DOWNT0 = 9;
else	TOK_ELSE	10	const int TOK_ELSE = 10;
end	TOK_END	11	const int TOK_END = 11;
eof	TOK_EOF	12	const int TOK_EOF = 12;
false	TOK_FALSE	13	const int TOK_FALSE = 13;
file	TOK_FILE	14	const int TOK_FILE = 14;
for	TOK_FOR	15	const int TOK_FOR = 15;
function	TOK_FUNCTION	16	const int TOK_FUNCTION = 16;
goto	TOK_GOTO	17	const int TOK_GOTO = 17;
if	TOK_IF	18	const int TOK_IF = 18;
in	TOK_IN	19	const int TOK_IN = 19;
integer	TOK_INTEGER	20	const int TOK_INTEGER = 20;
label	TOK_LABEL	21	const int TOK_LABEL = 21;
maxint	TOK_MAXINT	22	const int TOK_MAXINT = 22;
mod	TOK_MOD	23	const int TOK_MOD = 23;
nil	TOK_NIL	24	const int TOK_NIL = 24;
not	TOK_NOT	25	const int TOK_NOT = 25;
of	TOK_OF	26	const int TOK_OF = 26;
or	TOK_OR	27	const int TOK_OR = 27;
output	TOK_OUTPUT	28	const int TOK_OUTPUT = 28;
pack	TOK_PACK	29	const int TOK_PACK = 29;
packed	TOK_PACKED	30	const int TOK_PACKED = 30;
procedure	TOK_PROCEDURE	31	const int TOK_PROCEDURE = 31;
program	TOK_PROGRAM	32	const int TOK_PROGRAM = 32;
read	TOK_READ	33	const int TOK_READ = 33;
readln	TOK_READLN	34	const int TOK_READLN = 34;
real	TOK_REAL	35	const int TOK_REAL = 35;
record	TOK_RECORD	36	const int TOK_RECORD = 36;
repeat	TOK_REPEAT	37	const int TOK_REPEAT = 37;
set	TOK_SET	38	const int TOK_SET = 38;
string	TOK_STRING	39	const int TOK_STRING = 39;

then	TOK_THEN	40	const int TOK_THEN = 40;
to	TOK_TO	41	const int TOK_TO = 41;
true	TOK_TRUE	42	const int TOK_TRUE = 42;
type	TOK_TYPE	43	const int TOK_TYPE = 43;
until	TOK_UNTIL	44	const int TOK_UNTIL = 44;
var	TOK_VAR	45	const int TOK_VAR = 45;
while	TOK_WHILE	46	const int TOK_WHILE = 46;
with	TOK_WITH	47	const int TOK_WITH = 47;
write	TOK_WRITE	48	const int TOK_WRITE = 48;
writeln	TOK_WRITELN	49	const int TOK_WRITELN = 49;
.	TOK_PERIOD	50	const int TOK_PERIOD = 50;
(TOK_LEFTPAR	51	const int TOK_LEFTPAR = 51;
)	TOK_RIGHTPAR	52	const int TOK_RIGHTPAR = 52;
*	TOK_STAR	53	const int TOK_STAR = 53;
:	TOK_COLON	54	const int TOK_COLON = 54;
<	TOK_LTHAN	55	const int TOK_LTHAN = 55;
>	TOK_GTHAN	56	const int TOK_GTHAN = 56;
=	TOK_EQUAL	57	const int TOK_EQUAL = 57;
+	TOK_PLUS	58	const int TOK_PLUS = 58;
-	TOK_MINUS	59	const int TOK_MINUS = 59;
/	TOK_SLASH	60	const int TOK_SLASH = 60;
[TOK_LBRKT	61	const int TOK_LBRKT = 61;
]	TOK_RBRKT	62	const int TOK_RBRKT = 62;
,	TOK_COMMA	63	const int TOK_COMMA = 63;
;	TOK_SEMIC	64	const int TOK_SEMIC = 64;
^	TOK_HAT	65	const int TOK_HAT = 65;
<>	TOK_NOT_EQ	66	const int TOK_NOT_EQ = 66;
<=	TOK_LT_EQ	67	const int TOK_LT_EQ = 67;
>=	TOK_GT_EQ	68	const int TOK_GT_EQ = 68;
:=	TOK_ASSIGN	69	const int TOK_ASSIGN = 69;
..	TOK_DOTDOT	70	const int TOK_DOTDOT = 70;
E	E	199	const int E = 199;
PGM	NT_PROGRAM	200	const int NT_PROGRAM = 200;
DECLARATIONS	NT_DECLARATIONS	201	const int NT_DECLARATIONS = 201;
SUBPGM_DCLS	NT_SUBPGM_DCLS	202	const int NT_SUBPGM_DCLS = 202;
CMPD_STMT	NT_CMPD_STMT	203	const int NT_CMPD_STMT = 203;
ID_LIST	NT_ID_LIST	204	const int NT_ID_LIST = 204;
ID_LIST_END	NT_ID_LIST_END	205	const int NT_ID_LIST_END = 205;
DTYPE	NT_TYPE	206	const int NT_TYPE = 206;
STD_TYPE	NT_STD_TYPE	207	const int NT_STD_TYPE = 207;
SUBPGM_DCL	NT_SUBPGM_DCL	208	const int NT_SUBPGM_DCL = 208;
SUBPGM_HEAD	NT_SUBPGM_HEAD	209	const int NT_SUBPGM_HEAD = 209;
ARGS	NT_ARGS	210	const int NT_ARGS = 210;
PARM_LIST	NT_PARM_LIST	211	const int NT_PARM_LIST = 211;
PARM_LIST_END	NT_PARM_LIST_END	212	const int NT_PARM_LIST_END = 212;
OPT_STMTS	NT_OPT_STMTS	213	const int NT_OPT_STMTS = 213;
STMT_LIST	NT_STMT_LIST	214	const int NT_STMT_LIST = 214;

STMT_LIST_END	NT_STMT_LIST_END	215	const int NT_STMT_LIST_END = 215;
STMT	NT_STMT	216	const int NT_STMT = 216;
END_FOR	NT_END_FOR	217	const int NT_END_FOR = 217;
PROC_OR_ASSIGN	NT_PROC_OR_ASSIGN	218	const int NT_PROC_OR_ASSIGN = 218;
VARIABLE	NT_VARIABLE	219	const int NT_VARIABLE = 219;
VAR_END	NT_VAR_END	220	const int NT_VAR_END = 220;
PROC_STMT	NT_PROC_STMT	221	const int NT_PROC_STMT = 221;
PROC_STMT_END	NT_PROC_STMT_END	222	const int NT_PROC_STMT_END = 222;
EXPR_LIST	NT_EXPR_LIST	223	const int NT_EXPR_LIST = 223;
EXPR_LIST_END	NT_EXPR_LIST_END	224	const int NT_EXPR_LIST_END = 224;
EXPR	NT_EXPR	225	const int NT_EXPR = 225;
EXPR_END	NT_EXPR_END	226	const int NT_EXPR_END = 226;
RELOP	NT_RELOP	227	const int NT_RELOP = 227;
SIMPLE_EXPR	NT_SIMPLE_EXPR	228	const int NT_SIMPLE_EXPR = 228;
SIMPLE_EXPR_END	NT_SIMPLE_EXPR_END	229	const int NT_SIMPLE_EXPR_END = 229;
ADD_OP	NT_ADD_OP	230	const int NT_ADD_OP = 230;
TERM	NT_TERM	231	const int NT_TERM = 231;
TERM_END	NT_TERM_END	232	const int NT_TERM_END = 232;
MUL_OP	NT_MUL_OP	233	const int NT_MUL_OP = 233;
FACTOR	NT_FACTOR	234	const int NT_FACTOR = 234;
FACTOR_END	NT_FACTOR_END	235	const int NT_FACTOR_END = 235;
SIGN	NT_SIGN	236	const int NT_SIGN = 236;

[illegible]

[illegible]

_LIST>",
_LIST>","<ID_LIST_END>",
_LIST>","<ID_LIST_END>","<DTYPE>",
_LIST>","<ID_LIST_END>","<DTYPE>","<STD_TYPE>",
_LIST>","<ID_LIST_END>","<DTYPE>","<STD_TYPE>","<SUBPGM_DCL>",
_LIST>","<ID_LIST_END>","<DTYPE>","<STD_TYPE>","<SUBPGM_DCL>","<SUBPGM_HEAD>",
_LIST>","<ID_LIST_END>","<DTYPE>","<STD_TYPE>","<SUBPGM_DCL>","<SUBPGM_HEAD>","<ARGS>",
_LIST>","<ID_LIST_END>","<DTYPE>","<STD_TYPE>","<SUBPGM_DCL>","<SUBPGM_HEAD>","<ARGS>",".
_LIST>","<ID_LIST_END>","<DTYPE>","<STD_TYPE>","<SUBPGM_DCL>","<SUBPGM_HEAD>","<ARGS>",".
_LIST>","<ID_LIST_END>","<DTYPE>","<STD_TYPE>","<SUBPGM_DCL>","<SUBPGM_HEAD>","<ARGS>",".
_LIST>","<ID_LIST_END>","<DTYPE>","<STD_TYPE>","<SUBPGM_DCL>","<SUBPGM_HEAD>","<ARGS>",".

[illegible]

<PARM_LIST>","
<PARM_LIST>","<PARM_LIST_END>","
<PARM_LIST>","<PARM_LIST_END>","<OPT_STMTS>","
<PARM_LIST>","<PARM_LIST_END>","<OPT_STMTS>","<STMT_LIST>","

[illegible]

[illegible]

[illegible]

LE_EXPR_END>,"
LE_EXPR_END>","<ADD_OP>","
LE_EXPR_END>","<ADD_OP>","<TERM>","
LE_EXPR_END>","<ADD_OP>","<TERM>","<TERM_END>","
LE_EXPR_END>","<ADD_OP>","<TERM>","<TERM_END>","<MUL_OP>","
LE_EXPR_END>","<ADD_OP>","<TERM>","<TERM_END>","<MUL_OP>","<FACTOR>","
LE_EXPR_END>","<ADD_OP>","<TERM>","<TERM_END>","<MUL_OP>","<FACTOR>","<FACTOR_END>","
LE_EXPR_END>","<ADD_OP>","<TERM>","<TERM_END>","<MUL_OP>","<FACTOR>","<FACTOR_END>,"'

'<SIGN>',