**Expr:** term (( PLUS | MINUS) term)\* Example:**term:** factor (( MUL | DIV) factor)\* 17+8\*2-10/5**factor:** INTEGER

**Expr:** term (( PLUS | MINUS) term)\* Example:**term:** factor (( MUL | DIV) factor)\* 7+3\*(10/(12/(3+1)-1))**factor:** INTEGER | LEFTPAREN expr RIGHTPAREN

**Expr:** term (( PLUS | MINUS) term)\* Example:**term:** factor (( MUL | DIV) factor)\* 5-(-(2))**factor:** (PLUS | MINUS) factor | INTEGER | LEFTPAREN expr RIGHTPAREN

**Expr:** term (( PLUS | MINUS) term)\* Example:**term:** factor (( MUL | DIV) factor)\* 7+3^2**factor:** atom ( POW factor)\*  
**atom:** INTEGER | LEFTPAREN expr RIGHTPAREN

**expr:** term ((PLUS | MINUS) term)\*  
**term:** factor ((MUL | INTEGER\_DIV | FLOAT\_DIV) factor)\*  
**factor:** PLUS factor  
 | MINUS factor  
 | INTEGER\_CONST  
 | REAL\_CONST  
 | LPAREN expr RPAREN  
 | variable

**program:** PROGRAM variable SEMI block DOT

**block:** declarations compound\_statement

**declarartions:** (VAR (variable\_declarations SEMI)+)? procedure\_declarations

**formal\_parameters:** ID (COMMA ID)\* COLON type\_spec

**formal\_parameters\_list:** formal\_parameters | formal\_parameters SEMI formal\_parameter\_list

**variable\_declaration** : ID (COMMA ID)\* COLON type\_spec

**procedure\_declaration:** PROCEDURE ID (LPAREN formal\_parameter\_list RPAREN)? SEMI block SEMI

**type\_spec:** INTEGER | REAL

**compound\_statement:** BEGIN statement\_list END

**statement\_list:** statement | statement SEMI statement\_list

**statement:** compound\_statement |proccall\_statement| assignment\_statement | read\_statement | empty

**proccall\_statement:** ID LPAREN (expr (COMMA expr)\*)? RPAREN

**assignment\_statement:** variable ASSIGN expr

**read\_statement:** READ variable

**variable:** ID

**empty:**

**factor:** PLUS factor | MINUS factor | INTEGER | LEFTPAREN expr RIGHTPAREN | variable

**Example:**

PROGRAM Main;

VAR y: INTEGER;

z: INTEGER;

PROCEDURE Alpha(a : INTEGER; b : INTEGER);

VAR x : INTEGER;

BEGIN

x := (a + b ) \* 2;

PRINT x;

PRINT a;

END;

BEGIN { Main }

y := 3;

PRINT z;

PRINT y;

Alpha(z+1, 1);

END. { Main }