program::= function+

function::= 'fun' name '(' args ')' locals statements\_block

statements\_block::= 'begin' statements 'end'

statements::= statement(';' statement)\*

statement::= conditional\_block

| assignment

| print

| write

| read

| return

| 'skip'

| 'break'

| statements\_block

| name '(' expression\_list ')'

conditional\_block ::= 'while' relational\_expr 'do' statement

| 'if' relational\_expr 'then' statement

| 'if' relational\_expr 'then' statement 'else' statement

assignment::= (name ':=' expression

| name '[' integer ']' ':=' expression)

print::= 'print' '(' string ')'

write::= 'write' '(' expression ')'

read::= 'read' '(' name ')'

return::= 'return' expression

expression\_list::= expression (',' expression)\*

expression::= expression '+' expression

| expression '-' expression

| expression '\*' expression

| expression '/' expression

| '-' expression

| '+' expression

| '(' expression ')'

| name '('expression\_list')'

| name

| name '[' expression ']'

| literal

| 'int' '(' expression ')'

| 'float' '(' expression ')'

literal::= integer

| float

relational\_expr::= 'not'? relation (('and'|'or') 'not'? relation)\*

relation::= '(' expression ( '<' | '<=' | '>' | '>=' | '==' | '¡=' ) expression ')'

integer::= [1-9][0-9]+ | '0'

float::= '0' '.' [0-9]+ ('e' ('+' | '-' | ) [0-9]+)?

| [1-9][0-9]\* '.' [0-9]+ ('e' ('+' | '-' | ) [0-9]+)?

| [1-9][0-9]\* 'e' ('+' | '-' | ) [0-9]+

string::= '"' .\* ('\"' |'\\' | '\n' | ) '"' string?

name::= [a-zA-Z\_] [a-zA-Z\_0-9]\*

locals::= name':'(('int'|'float')('[' integer ']')?|'string')(';' name':'(('int'|'float')('[' integer ']')?|'string'))\*

args::= name':'(('int'|'float')('[' integer ']')?|'string')(',' name':'(('int'|'float')('[' integer ']')?|'string'))\*

* Palabras reservadas

BEGIN, END, IF, THEN, ELSE, WHILE, DO,

PRINT, WRITE, READ, RETURN, SKIP, BREAK,

FUN, INT\_TYPE, FLOAT\_TYPE, STRING\_TYPE,

* comentarios

COMENT,

* Operador de asignación

ASIGN,

* Operadores aritméticos

PLUS, MINUS, TIMES, DIVIDE,

LPAREN, RPAREN, LBRACKET, RBRACKET,

* Signos de puntuación

COLON, COMMA, SEMICOLON,

* Operadores lógicos

NOT, AND, OR,

* Operadores de relación

LESS, LESS\_EQUAL, GREATER, GREATER\_EQUAL, EQUAL, NOT\_EQUAL,

* Literales

NAME, INTEGER, FLOAT, STRING,

* Ignorar espacios en blanco y tabulaciones

ignore = ' \t'

* Palabras clave y palabras reservadas

BEGIN = r'begin( |\t|\n)'

END = r'end\b'

IF = r'if\b'

THEN = r'then\b'

ELSE = r'else\b'

WHILE = r'while\b'

DO = r'do( |\t|\n)'

PRINT = r'print\b'

WRITE = r'write\b'

READ = r'read\b'

RETURN = r'return\b'

SKIP = r'skip\b'

BREAK = r'break\b'

FUN = r'fun\b'

INT\_TYPE = r'int\b'

FLOAT\_TYPE = r'float\b'

STRING\_TYPE = r'string\b'

* Operador de asignación

ASIGN = r':='

* Operadores

PLUS = r'\+'

MINUS = r'-'

TIMES = r'\\*'

DIVIDE = r'/'

LPAREN = r'\('

RPAREN = r'\)'

LBRACKET = r'\['

RBRACKET = r'\]'

* Signos de puntuación

COLON = r':'

COMMA = r','

SEMICOLON = r';'

* Operadores relacionales

NOT\_EQUAL = r'!='

LESS\_EQUAL = r'<='

GREATER\_EQUAL = r'>='

EQUAL = r'=='

LESS = r'<'

GREATER = r'>'

* Operadores lógicos

AND = r'and\b'

OR = r'or\b'

NOT = r'not\b'

* Definir patrones para tokens

NAME = r'[a-zA-Z\_][a-zA-Z\_0-9]\*'

INTEGER = r'[1-9][0-9]\*|0'

FLOAT = r'0\.[0-9]+([eE][+\-]?[0-9]+)?|[1-9][0-9]\*\.[0-9]+([eE][+\-]?[0-9]+)?|[1-9][0-9]\*[eE][+\-]?[0-9]+'

STRING = r'"(\\.|[^"\\])\*"'