

GRAMÁTICA 3.0

Literal: (-?[0-9]*) U (0x[abcdef0-9]*) U ([a-z]) U (“[a-z]*[^\n\$”]”)

ID: ((_ | .)(_ | . | a-z0-9)*(a-z0-9)*([(a-z0-9)*])? | ((a-z)(a-z0-9 | _ | .)*([a-z0-9]*])?)

Programa	S -> DBF λ
Declarações	D -> VD CD B
Declaração de variáveis	V -> 'var' V1
Declaração 1	V1 -> TL;
Tipos	T -> 'char' 'integer'
Lista de IDs ou valores	L -> ID='-[0-9]*,L ID,L ID['{int>0}'],L ID ID['{int>0}'] ID='[a-z],L
Declaração Constantes	C -> 'const' ID '=' Literal ';' ;
Bloco de comandos	B -> AB RB TB ;B EB WB
Atribuição	A -> ID '=' X;
Repetição	R -> 'for' ID '=' X 'to' X R1
Step opcional	R1 -> 'step' X 'do' T2 'do' T2
Teste	T -> 'if' X 'then' T2 T1
Else	T1 -> else T2 λ
Comando ou Comandos	T2 -> '{ B }' A R T ; E W
Leitura	E -> 'readln('ID');'
Escrita	W -> 'write('X2');' 'writeln('X2');'
Expressão	X -> '('X')' ID X1 Literal X1
Expressão 1	X1 -> O X λ
Operadores	O -> '='(str) '>=' '<=' '>' '<' '<>' '='(arit) 'or' '+' '-' '%' '/' 'and' '*' 'not'
Lista de expressões	X2 -> X X3
Expressão 5	X3 -> ','X2 λ
Fim de arquivo	F -> EOF