Token	#token	Lexema
const	1	(c)(o)(n)(s)(t)
integer	2	(i)(n)(t)(e)(g)(e)(r)
byte	3	(b)(y)(t)(e)
string	4	(s)(t)(r)(i)(n)(g)
while	5	(w)(h)(i)(l)(e)
if	6	(i)(f)
else	7	(e)(l)(s)(e)
and	8	(a)(n)(d)
or	9	(o)(r)
not	10	(n)(o)(t)
atribuicao	11	(=)
comparacao	12	(==)
(	13	(
)	14	)
<	15	<
>	16	>
!=	17	!=
>=	18	>=
<=	19	<=
,	20	,
op.soma	21	(+)
op.subtração	22	(-)
*	23	*
/	24	/
;	25	;
begin	26	(b)(e)(g)(i)(n)
end	27	(e)(n)(d)
then	28	(t)(h)(e)(n)
readln	29	(r)(e)(a)(d)(l)(n)
write	30	(w)(r)(i)(t)(e)
writeln	31	(w)(r)(i)(t)(e)(l)(n)
main	32	(m)(a)(i)(n)
boolean	33	(b)(o)(o)(l)(e)(a)(n)
variavel_bool	34	valor
identificador	35	((_+ (D U L)) U L) (L U D U_)*
valor		(- U lambda)((digito)+ U (atribuição byte) U (logico) U (string_const)
permitidos		(letras U digitos U espaço U '_' U '.' U ',' U ';' U '&' U ':' U '(' U ')' U '[' U ']' U '{' U '}' U '+' U '-' U ' ' U '"' U '/' U '!' U '?' U '>' U '<' U '=')
		hexa = ('A' U 'a' U 'B' U 'b' U 'C' U 'c' U 'D' U 'd' U 'E' U 'e' U 'F' U 'f' U '0' U '1' U '2' U '3' U '4' U '5' U '6' U '7' U '8' U '9')
		logico = ((t)(r)(u)(e) U (f)(a)(l)(s)(e))
		digito = ('0' U '1' U '2' U '3' U '4' U '5' U '6' U '7' U '8' U '9')
		letras = ('A' U 'B' U 'C' U 'D' U 'E' U 'F' U 'G' U 'H' U 'I' U 'J' U 'K' U 'L' U 'M' U 'N' U 'O' U 'P' U 'Q' U 'R' U 'S' U 'T' U 'U' U 'V' U 'X' U 'W' U 'Y' U 'Z' U 'a' U 'b' U 'c' U 'd' U 'e' U 'f' U 'g' U 'h' U 'i' U 'j' U 'k' U 'l' U 'm' U 'n' U 'o' U 'p' U 'q' U 'r' U 's' U 't' U 'u
		string_const = (')(permitidos)(')
		atribuição byte = (0) (h) (hexa) (hexa U lambda)