



		{ }
STATELIST	{int, float, string, ident, print, read, return, if, for, '{', break, ';'}	{ ' ' }
STATELIST2	{int, float, string, ident, print, read, return, if, for, '{', break, ';', &}	{ ' ' }
ALLOCEXPRESSION	{new}	{',', ')')}
ALLOCEXPRESSION2	{int, float, string}	{',', ')')}
B	{ '[', & }	{ '=', '*', '\', '%', '+', '-', '<', '>', '<=', '>=', '==', '!=', ',', ')', ']', '[' }
T2	{ '[' }	{ '=', '*', '\', '%', '+', '-', '<', '>', '<=', '>=', '==', '!=', ',', ')', ']', '[' }
EXPRESSION	{ '+', '-', int_constant, float_constant, string_constant, null, ident, '(' }	{ ',', ')')}
EXPRESSION2	{ '<', '>', '<=', '>=', '==', '!=', & }	{ ',', ')')}
NUMEXPRESSION	{ '+', '-', int_constant, float_constant, string_constant, null, ident, '(' }	{ ',', ')', ']', '<', '>', '<=', '>=', '==', '!= ' }
C	{ '+', '-', & }	{ '<', '>', '<=', '>=', '==', '!=', ',', ')', ']' }
T3	{ '+', '-' }	{ '+', '-', '<', '>', '<=', '>=', '==', '!=', ',', ')', ']' }
TERM	{ '+', '-', int_constant, float_constant, string_constant, null, ident, '(' }	{ '+', '-', '<', '>', '<=', '>=', '==', '!=', ',', ')', ']' }
D	{ '*', '\', '%', & }	{ '+', '-', '<', '>', '<=', '>=', '==', '!=', ',', ')', ']' }
T4	{ '*', '\', '%' }	{ '*', '\', '%', '+', '-', '<', '>', '<=', '>=', '==', '!=', ',', ')', ']' }
UNARYEXPR	{ '+', '-', int_constant, float_constant, string_constant, null, ident, '(' }	{ '*', '\', '%', '+', '-', '<', '>', '<=', '>=', '==', '!=', ',', ')', ']' }
FACTOR	{int_constant, float_constant, string_constant, null, ident, '('}	{ '*', '\', '%', '+', '-', '<', '>', '<=', '>=', '==', '!=', ',', ')', ']' }

LVALUE	{ident}	{ '=', '*', '\', '%', '+', '-', '<', '>', '<=', '>=', '==', '!=', ':', ')', ']' }
--------	---------	---