Х	FIRST(X)	FOLLOW(X)
PROGRAM	{int, float, string, ident, print, read, return, if, for, '{', break, ';', def, &}	{\$}
FUNCLIST	{def}	{\$}
FUNCLIST2	{def, &}	{\$}
FUNCDEF	{def}	{def, \$}
PARAMLIST	{int, float, string, &}	{ ')' }
PARAMLIST2	{',', <mark>&</mark> }	{ ')' }
STATEMENT	{int, float, string, ident, print, read, return, if, for, '{', break, ';'}	{\$, int, float, string, ident, print, read, return, if, for, '{', break, ';', '}'}
VARDECL	{int, float, string}	{`;'}
Α	{'[', &}	{`;'}
T1	{T}	{ '[', ',']}
ATRIBSTAT	{ident}	{';', ')'}
ATRIBSTAT2	{ident, new, '+', '-', int_constant, float_constant, string_constant, null, '(')	{';', ')'}
ATRIBSTAT3	{'[', '*', '\', '%', '+', '-', '<', '>', '<=', '>=', '==', '!=', '(', &}	{';', ')'}
PARAMLISTCALL	{ident, &}	{ ')' }
PARAMLISTCALL2	{',', <mark>&</mark> }	{ ')' }
PRINTSTAT	{print}	{ `; '}
READSTAT	{read}	{ `; '}
RETURNSTAT	{return}	{`;'}
IFSTAT	{if}	{\$, int, float, string, ident, print, read, return, if, for, '{', break, ';', '}'}
IFSTAT2	{else, &}	{\$, int, float, string, ident, print, read, return, if, for, '{', break, ';', '}'}
FORSTAT	{for}	{\$, int, float, string, ident, print, read, return, if, for, '{', break, ';',

		1
		'}'}
STATELIST	<pre>{int, float, string, ident, print, read, return, if, for, '{', break, ';'}</pre>	{ '}' }
STATELISt2	{int, float, string, ident, print, read, return, if, for, '{', break, ';', &}	{ '}' }
ALLOCEXPRESSION	{new}	{';', ')'}
ALLOCEXPRESSION2	{int, float, string}	{';', ')'}
В	{'[', &}	{'=', '*', '\', '%', '+', '-', '<', '>', '<=', '>=', '==', '!=', ';', ')', ']'}
Т2	{T}	{'=', '*', '\', '%', '+', '-', '<', '>', '<=', '>=', '==', '!=', ';', ')', ']', '['}
EXPRESSION	{'+', '-', int_constant, float_constant, string_constant, null, ident, '('}	{';', ')'}
EXPRESSION2	{'<', '>', '<=', '>=', '==', '!=', &}	{';', ')'}
NUMEXPRESSION	{'+', '-', int_constant, float_constant, string_constant, null, ident, '('}	{',', ')', ']', '<', '>', '<=', '>'=, '==', '!='}
С	{'+', '-', &}	{'<', '>', '<=', '>=', '==', '!=', ';', ')', ']'}
Т3	{'+', '-'}	{'+', '-', '<', '>', '<=', '>=', '==', '!=', ';', ')', ']'}
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}	{'=', '*', '\', '%', '+', '-', '<', '>', '<=', '>=', '==', '!=', ';', ')', ']'}
		,, :-, ,, /, 1}