First e Follow da Linguagem PasC

	First	Follow
prog	program	$
body	num, char	$
decl-list	num, char , ε	{
decl	num, char	;
type	num, char	id
id-list	id	;
id-list'	, , ε	;
stmt-list	id, if, while, read, write, ε	}
stmt	id, if, while, read, write	;
assign-stmt	id	;
if-stmt	if	;
if-stmt'	else, ε	;
condition	id, num_const, char_const, (, not	)
while-stmt	while	;
stmt-prefix	while	{
read-stmt	read	;
write-stmt	write	;
writable	id, num_const, char_const, (, not, literal	;
expression	id, num_const, char_const, (, not	)
expression'	==, >, >=, <, <=, !=, ε	)
simple-expr	id, num_const, char_const, (, not	; , ==, >, >=, <, <=, !=,)
simple-expr'	+, -, or, ε	; , ==, >, >=, <, <=, !=,)
term	id, num_const, char_const, (, not	+, -, or, ; , ==, >, >=, <, <=, !=,)
term'	*, /, and, ε	+, -, or, ; , ==, >, >=, <, <=, !=,)
factor-a	id, num_const, char_const, (, not	*, /, and, +, -, or, ; , ==, >, >=, <, <=, !=,)
factor	id, num_const, char_const, (	*, /, and, +, -, or, ; , ==, >, >=, <, <=, !=,)
relop	==, >, >=, <, <=, !=	id, num_const, char_const, (, not
addop	+, -, or	id, num_const, char_const, (, not
mulop	*, /, and	id, num_const, char_const, (, not
constant	num_const, char_const	*, /, and, +, -, or, ; , ==, >, >=, <, <=, !=,)