X	NULL(X)
prog	
statilst	
statlistp	x
stat	
whenlist	
whenlistp	x
whenitem	
bexpr	
expr	
exprlist	
exprlistp	x

X	FIRST(X)
prog	{'=', print, read, cond, while, '{'}
statilst	{'=', print, read, cond, while, '{'}
statlistp	{';'}
stat	{'=', print, read, cond, while, '{'}
whenlist	{when}
whenlistp	{when}
whenitem	{when}
bexpr	{RELOP}
expr	{'+', '-', '*', '/', NUM, ID}
exprlist	{'+', '-', '*', '/', NUM, ID}
exprlistp	{'+', '-', '*', '/', NUM, ID}

X	FOLLOW(X)
prog	{$}
statilst	{'EOF', '}'}
statlistp	{'EOF', '}'}
stat	{';', 'EOF', '}', when, else}
whenlist	{else}
whenlistp	{else}
whenitem	{when, else}
bexpr	{')'}
expr	{'+', '-', '*', '/', NUM, ID, ';', 'EOF', '}', ')', else, when}
exprlist	{')'}
exprlistp	{')'}

PRODUZIONE	GUIDA(PRODUZIONE)
<prog> -> <statlist>EOF	{'=', print, read, cond, while, '{'}
<statlist> -> <stat><statlistp>	{'=', print, read, cond, while, '{'}
<statlistp> -> ;<stat><statlistp>	{;}
<statlistp> -> *eps*	{'EOF', '}'}
<stat> -> =ID<expr>	{'='}
<stat> -> print(exprlist>)	{print}
<stat> -> read(ID)	{read}
<stat> -> cond<whenlist>else<stat>	{cond}
<stat> -> while(<bexpr>)<stat>	{while}
<stat> -> {<statlist>}	{'{'}
<whenlist> -> <whenitem><whenlistp>	{when}
<whenlistp> -> <whenitem><whenlistp>	{when}
<whenlisp> -> eps	{else}
<whenitem> -> when(<bexpr>)do<stat>	{when}
<bexpr> -> RELOP<expr><expr>	{RELOP}
<expr> -> +(<exprlist>)	{+}
<expr> -> *(<exprlist>)	{*}
<expr> -> -<expr><expr>	{-}
<expr> -> /<expr><expr>	{/}
<expr> -> NUM	NUM
<expr> -> ID	ID
<exprlist> -> <expr><exprlistp>	{'+', '-', '*', '/', NUM, ID}
<exprlistp> -> <expr><exprlistp>	{'+', '-', '*', '/', NUM, ID}
<exprlistp> -> *eps*	{')'}