

BNF

$\langle \text{program} \rangle ::= \underline{\text{program}} \langle \text{identifier} \rangle \langle \text{compound statement} \rangle$

$\langle \text{compound statement} \rangle ::= \underline{\text{begin}} \langle \text{statement}' \rangle \underline{\text{end}}$

$\langle \text{statement}' \rangle ::= \langle \text{statement} \rangle \langle \text{statement}' \rangle \mid \langle \text{statement} \rangle$

$\langle \text{statement} \rangle ::= \langle \text{conditional statement} \rangle \mid \langle \text{simple statement} \rangle ;$

$\langle \text{conditional statement} \rangle ::= \underline{\text{if}} \langle \text{expression} \rangle \langle \text{compound statement} \rangle \langle \text{else-if statement} \rangle \mid$
 $\underline{\text{if}} \langle \text{expression} \rangle \langle \text{compound statement} \rangle \langle \text{else-if statement} \rangle \underline{\text{else}} \langle \text{compound statement} \rangle$

$\langle \text{else-if statement} \rangle ::= \underline{\text{else-if}} \langle \text{expression} \rangle \langle \text{compound statement} \rangle \langle \text{else-if statement} \rangle \mid \epsilon$

$\langle \text{simple statement} \rangle ::= \langle \text{assignment statement} \rangle \mid \langle \text{print statement} \rangle \mid \langle \text{declaration statement} \rangle$

$\langle \text{assignment statement} \rangle ::= \langle \text{identifier} \rangle \underline{=} \langle \text{expression} \rangle$

$\langle \text{print statement} \rangle ::= \underline{\text{print_line}} \underline{(\langle \text{string literal} \rangle)}$

$\langle \text{declaration statement} \rangle ::= \langle \text{type} \rangle \langle \text{variable declaration}' \rangle$

$\langle \text{variable declaration}' \rangle ::= \underline{,} \langle \text{variable declaration} \rangle \langle \text{variable declaration}' \rangle \mid \langle \text{variable declaration} \rangle$

$\langle \text{variable declaration} \rangle ::= \langle \text{identifier} \rangle \mid \langle \text{identifier} \rangle \underline{=} \langle \text{expression} \rangle$

$\langle \text{identifier} \rangle ::= \underline{\text{identifier}}$

$\langle \text{expression} \rangle ::= \langle \text{simple expression} \rangle \mid \langle \text{simple expression} \rangle \langle \text{relational operator} \rangle \langle \text{simple expression} \rangle$

$\langle \text{simple expression} \rangle ::= \langle \text{simple expression} \rangle \langle \text{adding operator} \rangle \langle \text{term} \rangle \mid \langle \text{term} \rangle$

$\langle \text{term} \rangle ::= \langle \text{term} \rangle \langle \text{multiplying operator} \rangle \langle \text{factor} \rangle \mid \langle \text{factor} \rangle$

$\langle \text{factor} \rangle ::= \langle \text{identifier} \rangle \mid \langle \text{number literal} \rangle \mid \underline{(\langle \text{expression} \rangle)}$

$\langle \text{relational operator} \rangle ::= \leq \mid \geq \mid =$

$\langle \text{adding operator} \rangle ::= + \mid =$

$\langle \text{multiplying operator} \rangle ::= * \mid /$

$\langle \text{string literal} \rangle ::= \underline{\text{string_literal}}$

$\langle \text{number literal} \rangle ::= \underline{\text{number_literal}}$

$\langle \text{type} \rangle ::= \underline{\text{int}}$

EBNF

$\langle \text{program} \rangle ::= \underline{\text{program}} \langle \text{identifier} \rangle \langle \text{compound statement} \rangle$

$\langle \text{compound statement} \rangle ::= \underline{\text{begin}} \{ \langle \text{statement} \rangle \} \langle \text{statement} \rangle \underline{\text{end}}$

$\langle \text{statement} \rangle ::= \langle \text{conditional statement} \rangle \mid \langle \text{simple statement} \rangle ;$

$\langle \text{conditional statement} \rangle ::= \underline{\text{if}} \langle \text{expression} \rangle \langle \text{compound statement} \rangle$
 $\quad \{ \underline{\text{else_if}} \langle \text{expression} \rangle \langle \text{compound statement} \rangle \}$
 $\quad [\underline{\text{else}} \langle \text{compound statement} \rangle]$

$\langle \text{simple statement} \rangle ::= \langle \text{assignment statement} \rangle \mid \langle \text{print statement} \rangle \mid \langle \text{declaration statement} \rangle$

$\langle \text{assignment statement} \rangle ::= \langle \text{identifier} \rangle \underline{=} \langle \text{expression} \rangle$

$\langle \text{print statement} \rangle ::= \underline{\text{print_line}} \underline{(} \langle \text{string literal} \rangle \underline{)}$

$\langle \text{declaration statement} \rangle ::= \langle \text{type} \rangle \{ \langle \text{variable declaration} \rangle \} \underline{,} \langle \text{variable declaration} \rangle$

$\langle \text{variable declaration} \rangle ::= \langle \text{identifier} \rangle [\underline{=} \langle \text{expression} \rangle]$

$\langle \text{identifier} \rangle ::= \underline{\text{identifier}}$

$\langle \text{expression} \rangle ::= \langle \text{simple expression} \rangle [\langle \text{relational operator} \rangle \langle \text{simple expression} \rangle]$

$\langle \text{simple expression} \rangle ::= \{ \langle \text{term} \rangle \langle \text{adding operator} \rangle \} \langle \text{term} \rangle$

$\langle \text{term} \rangle ::= \{ \langle \text{factor} \rangle \langle \text{multiplying operator} \rangle \} \langle \text{factor} \rangle$

$\langle \text{factor} \rangle ::= \langle \text{identifier} \rangle \mid \langle \text{number literal} \rangle \mid \underline{(} \langle \text{expression} \rangle \underline{)}$

$\langle \text{relational operator} \rangle ::= \underline{=} \mid \underline{>} \mid \underline{<} \mid \underline{! =}$

$\langle \text{adding operator} \rangle ::= \underline{+} \mid \underline{=}$

$\langle \text{multiplying operator} \rangle ::= \underline{*} \mid \underline{/}$

$\langle \text{string literal} \rangle ::= \underline{\text{string_literal}}$

$\langle \text{number literal} \rangle ::= \underline{\text{number_literal}}$

$\langle \text{type} \rangle ::= \underline{\text{int}}$