1. Grammer

1.1 Root

$$\begin{aligned} <root> &::= <stmt-list> \\ <stmt-list> &::= <stmt><stmt-list> \mid <stmt> \\ <stmt> &::= <assignment> \mid <expr> \mid <if-stmt> \mid <loopexp> \\ &\mid <function-declaration> \mid <function-invocation> \end{aligned} \tag{1.1}$$

1.2 Assignment

$$<assignment> ::= <declarator>; \mid <reassign>; \tag{1.2}$$

1.2.1 Declarator

$$\begin{aligned} <declarator> &::= <type-name><declarator-list> \\ <type-name> &::= boolean|int|float|char|string \\ <declarator-list> &::= <delarator> \mid <declarator>, <declarator-list> \\ <declarator> &::= id|id = <expr> |id <array-list> |id <array-list> = \{<list-values>\} \\ <array-list> &::= [<value>]|[<value>] <array-list> \\ <list-values> &::= <expr>, <list-values> \mid <expr> |\{<list-values>\} \\ <value> &::= integer|float| <empty> \end{aligned} \tag{1.3}$$

1.2.2 Reassignment

$$\begin{aligned} <reassign> &::= id = <expr> |id <reassign-array-list> = <expr> \\ <reassign-array-list> &::= [<reassign-value>]|[<reassign-value>] <reassign-array-list> \\ <reassign-value> &::= integer|float \end{aligned} \tag{1.4}$$

1.3 Expressions

$$<expr> ::= <stringexp> \mid <logconj> \mid <addsubexp> \tag{1.5}$$

1.3.1 Boolean Operations

$$\begin{aligned} <logconj> &::= <logconj> \mid\mid <logdisj> \mid <logdisj> \\ <logdisj> &::= <logdisj> \&\& <logneg> \mid <logneg> \\ <logneg> &::= ! <compopt> \mid <compopt> \\ <compopt> &::= <stringexp> == <stringexp> \mid <addsubexp> >= <addsubexp> \\ &\mid <addsubexp> <= <addsubexp> \mid <stringexp> != <stringexp> \end{aligned} \tag{1.6}$$

1.3.2 String operators

$$\begin{aligned} < stringexp > &::= < stringroot > + < stringexp > \mid < stringroot > \mid < addsubexp > \\ < stringroot > &::= string|id \end{aligned} \tag{1.7}$$

1.3.3 Arithmatic expressions

$$\begin{aligned} < addsubexp > &::= < addsubexp > + < muldivexp > \mid < addsubexp > - < muldivexp > \mid < muldivexp > \\ < muldivexp > &::= < muldivexp > * < expexp > \mid < muldivexp > / < expexp > \mid < expexp > \\ < expexp > &::= < rootexp > \# < expexp > \mid < rootexp > \\ < rootexp > &::= (< expr >)|id|integer|float \end{aligned} \tag{1.8}$$

1.4 Conditional

$$< if - stmt > ::= if(< expr >)\{< stmt - list >\}else\{< stmt - list >\}|if(< expr >)\{< stmt - list >\} \tag{1.9}$$

1.5 Loops

$$\begin{aligned} < loopexp > &::= < forloop > \mid < whileloop > \\ < forloop > &::= for(assignment; < compopt >; < reassign >)\{< stmt - list >\} \\ < whileloop > &::= for(< compopt >)\{< stmt - list >\} \end{aligned} \tag{1.10}$$

1.6 Functions

1.6.1 Declare

$$\begin{aligned} < function - declaration > &::= < func - type > id(< declare - arg - list >)\{< stmt - list >\} \\ < func - type > &::= < type - name > |void \\ < declare - arg - list > &::= < declare - arg >, < arg - list > \mid < declare - arg > \\ < declare - arg > &::= < type - name > id| < empty > \end{aligned} \tag{1.11}$$

1.6.2 Invocation

$$\begin{aligned} < function - invocation > &::= < type - name > id = id(arg - list); |id(arg - list); \\ < arg - list > &::= < arg >, < arg - list > \mid < arg > \\ < arg > &::= < declarator > |id| < empty > \end{aligned} \tag{1.12}$$