# CSCI 512 Design and Implementation of Computer Programming Languages
# Winter, 2017

## Denotational Semantics for PostFix

### Semantic Domains

$t \in StackTransform = Stack \to Stack$
$s \in Stack = Value^* + Error$
$d \in Dictionary = Ident \to (Value + \{unbound\})$
$v \in Value = Ident + Int + StackTransform$
$r \in Result = Value + Error$
$a \in Answer = Int + Error$
$\quad Error = \{\texttt{error}\}$
$n \in Ident = \{\text{""}, \text{"}a\text{"}, \text{"}b\text{"}, \ldots, \text{"}ab\text{"}, \ldots\}$
$i \in Int = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$
$b \in Bool = \{true, false\}$

### Constants and Functions on PostFix Semantic Domains

*errorResult : Result*
: An error in the domain *Result*.

*errorAnswer : Answer*
: An error in the domain *Answer*.

*errorStack : Stack*
: The distinguished error stack.

*unbound : Dictionary*
: A token representing an unbound identifier in the domain *Dictionary*.

*ErrorTransform : StackTransform*
: A transform that maps all stacks to *errorStack*.

*push : Result → StackTransform*
: Given a result value $v$, return a transform that pushes $v$ onto a stack; otherwise return *errorTransform*.

*pop : StackTransform*
: For a nonempty stack $s$, return the stack resulting from popping the top value; otherwise return *errorStack*.

*top : Stack → Result*
: Given a nonempty stack $s$, return a result that is the top element of $s$; otherwise return *errorResult*.

*intAt : Int → Stack → Result*
: Given the integer $i_{index}$ and a stack whose $i_{index}$th element (starting from 1) is the integer $i_{result}$, return $i_{result}$; otherwise return *errorResult*.

*arithop : (Int → Int → Result) → StackTransform*

Let $f$ *: Int → Int → Result* be the functional argument to *arithop*. Return a transform with the following behavior: if the given stack has two integers $i_1$ and $i_2$ followed by $s_{rest}$, then return a stack whose top value $v_{result}$ is followed by by $s_{rest}$, where $(Value \rightarrowtail Result\ v_{result})$ is the result of the application $(f\ i_1\ i_2)$. If the given stack is not of this form or if the result of applying $f$ is *errorResult*, then return *errorStack*.

*transform : Result → StackTransform*

Given a result that is a stack transform, return it; otherwise return *errorTransform*.

*resToAns : Result → Answer*

Given a result that is an integer, return it as an answer; otherwise return *errorAnswer*.

*bind : Ident → Value → Dictionary*

Given a name, $n$, and a value, $v$, update the dictionary object, $d$, with a mapping between the $n$ and $v$ added to any existing bindings; if the name is already bound in the given dictionary, the new binding replaces the previous.

### Valuation functions for PostFix

The signatures of the valuation functions provide insight into the meaning of the language elements.

$\mathcal{P}$ : Prog→ $Int^* \rightarrow Answer$
$\mathcal{Q}$ : CommandSeq→ *StackTransform*
$\mathcal{C}$ : Command→ *StackTransform*
$\mathcal{A}$ : ArithmeticOperator → (*Int → Int → Result*)
$\mathcal{R}$ : RelationalOperator → (*Int → Int → Bool*)
$\mathcal{N}$ : IntLit → *Int*
$\mathcal{I}$ : Identifier → *Ident*

The meaning pf a program (`postfix` $N_{numargs}\ Q$) is a function that transforms an initial stack consisting of the integers in the argument sequence via the transform $\mathcal{Q}[\![Q]\!]$ and returns the top integer of the resulting stack.

The meaning of a command sequence is the composition of the transforms of its component command. The order of the composition

$$\mathcal{Q}[\![Q]\!] \circ \mathcal{C}[\![C]\!] = \lambda s\ .\ (\mathcal{Q}[\![Q]\!]\ (\mathcal{C}[\![C]\!]\ s))$$

is crucial because it guarantees that the stack manipulations of the first command can be observed by subsequent commands.

$\mathcal{P}[\![(\texttt{postfix}\ N_{numargs}\ Q)]\!]$
$= \lambda i^*\ .$ **if** $(length\ i^*) =_{Int} \mathcal{N}[\![N_{numargs}]\!]$
**then** $resToAns\ (top\ (\mathcal{Q}[\![Q]\!](Value^* \rightarrowtail Stack\ (map\ Int \rightarrowtail Value\ i^*))))$
**else** *errorAnswer* **end**

$\mathcal{Q}[\![C.Q]\!] = \mathcal{Q}[\![Q]\!] \circ \mathcal{C}[\![C]\!]$

$\mathcal{Q}[\![]\!] = \lambda s\ .\ s$

$$\mathcal{C}[\![N]\!] = push\ (Value \rightarrowtail Result\ (Int \rightarrowtail Value\ \mathcal{N}[\![N]\!]))$$

$$\mathcal{C}[\![I]\!] = push\ (Value \rightarrowtail Result\ (Int \rightarrowtail Value\ \mathcal{I}[\![I]\!]))$$

$$\mathcal{C}[\![(Q)]\!] = push\ (Value \rightarrowtail Result\ (StackTransform \rightarrowtail Value\ \mathcal{Q}[\![Q]\!]))$$

$$\mathcal{C}[\![\texttt{pop}]\!] = pop$$

$$\mathcal{C}[\![\texttt{swap}]\!] = \lambda s\,.\,(push\ (top\ (pop\ s))\ (push\ (top\ s)\ (pop\ (pop\ s))))$$

$$\mathcal{C}[\![\texttt{pair}]\!] = \lambda s\,.\,(push\ \langle (top\ s)\,.\,(top\ (pop\ s))\rangle\ (pop\ (pop\ s)))$$

$$\begin{array}{l}\mathcal{C}[\![\texttt{fst}]\!] = \lambda s\,.\ \textbf{match}\ top\ s\\ \qquad \triangleright(Value^* \rightarrowtail Stack\ (\langle v_1, v_2\rangle\,.\,v^*_{rest})[\!]\\ \qquad (push\ i_1\ (pop\ s))\\ \qquad \triangleright\ \textbf{else}\ errorStack\ \textbf{end}\end{array}$$

$$\begin{array}{l}\mathcal{C}[\![\texttt{snd}]\!] = \lambda s\,.\ \textbf{match}\ top\ s\\ \qquad \triangleright(Value^* \rightarrowtail Stack\ (\langle v_1, v_2\rangle\,.\,v^*_{rest})[\!]\\ \qquad (push\ i_2\ (pop\ s))\\ \qquad \triangleright\ \textbf{else}\ errorStack\ \textbf{end}\end{array}$$

$$\begin{array}{l}\mathcal{C}[\![\texttt{for}]\!] = \lambda s\,.\ \textbf{match}\ top\ s\\ \qquad \triangleright(Value^* \rightarrowtail Stack\ \langle (Int \rightarrowtail Value\ i), (Value^* \rightarrowtail StackTransform\ t^*)\rangle)[\!]\\ \qquad \textbf{if}\ 1 \leq_{Int} i\\ \qquad \textbf{then}\ (push\ (i\,.\,t^*\ @\ \langle (i-1)\,.\,t^*\rangle)\ (pop\ s))\ \ \textbf{end}\\ \qquad \triangleright(Value^* \rightarrowtail Stack\ \langle (Int \rightarrowtail Value\ i), (Value^* \rightarrowtail StackTransform\ t^*)\rangle)[\!]\\ \qquad \textbf{if}\ 0 =_{Int} i\\ \qquad \textbf{then}\ (pop\ s)\ \ \textbf{end}\\ \qquad \triangleright\ \textbf{else}\ errorStack\ \textbf{end}\end{array}$$

$$\begin{array}{l}\mathcal{C}[\![\texttt{nget}]\!] = \lambda s\,.\ \textbf{match}\ top\ s\\ \qquad \triangleright(Value \rightarrowtail Result\ (Int \rightarrowtail Value\ i))\ [\!]\ (push\ (intAt\ i\ (pop\ s))\ (pop\ s)\\ \qquad \triangleright\ \textbf{else}\ errorStack\ \textbf{end}\end{array}$$

$$\begin{array}{l}\mathcal{C}[\![\texttt{sel}]\!] = \lambda s\,.\ \textbf{match}\ top\ (pop\ (pop\ s))\\ \qquad \triangleright\ (Value \rightarrowtail Result\ (Int \rightarrowtail Value\ i))[\!]\\ \qquad\ push\ (\textbf{if}\ i =_{Int} 0\ \textbf{then}\ top\ s\ \textbf{else}\ top\ (pop\ s)\ \textbf{end})\\ \qquad\qquad (pop\ (pop\ (pop\ s))\\ \qquad \triangleright\ \textbf{else}\ errorStack\ \textbf{end}\end{array}$$

$$\mathcal{C}[\![\texttt{exec}]\!] = \lambda s\,.\,(transform\ (top\ s)\ (pop\ s))$$

$$\mathcal{C}[\![A]\!] = arithop\ \mathcal{A}[\![A]\!]$$

$$\begin{array}{l}\mathcal{C}[\![R]\!] = arithop\ (\lambda i_1 i_2\,.\,(Value \rightarrowtail Result\\ \qquad\qquad\qquad (Int \rightarrowtail Value\ (\textbf{if}\ (\mathcal{R}[\![R]\!]\ i_1\ i_2)\ \textbf{then}\ \mathit{1}\ \textbf{else}\ \mathit{0}\ \textbf{end}))))\end{array}$$

$$\begin{array}{l}\mathcal{C}[\![\texttt{ref}]\!] = \lambda d\,.\,(\lambda s\,.\ \textbf{match}\ top\ s\\ \qquad\qquad \triangleright(Value \rightarrowtail Result\ (Ident \rightarrowtail Value\ v_{name}))[\!]\\ \qquad\qquad\ \textbf{then match}\ v_{name}\end{array}$$

$$
\begin{array}{l}
\quad \triangleright(Value \rightarrowtail Dictionary\ ((Ident \rightarrowtail Value\ v_{name})\ (Value \rightarrowtail Result\ v_{val})))[] \\
\quad (push\ v_{val}\ (pop\ s)) \\
\quad \textbf{else}\ errorStack \\
\quad \triangleright\ \textbf{else}\ errorStack\ \textbf{end})
\end{array}
$$

$$
\begin{array}{l}
\mathcal{C}[\![\texttt{def}]\!] = \lambda d\,.\,(\lambda s\,.\ \textbf{match}\ s \\
\quad \triangleright(Value^* \rightarrowtail Stack\ ((Int \rightarrowtail Value\ v_1)\,.\,(Ident \rightarrowtail Value\ v_2)\,.\,v^*_{rest}))[] \\
\quad \textbf{if}\ v_2 \notin \{pop, swap, pair, fst, snd, for, nget, sel, exec, \\
\quad\ add, sub, mul, div, rem, ref, def\} \\
\quad \textbf{then}\ (bind\ v_2\ v_1) \wedge (pop\ (pop\ s))\ \textbf{end} \\
\quad \triangleright\ \textbf{else}\ errorStack\ \textbf{end})
\end{array}
$$

$$\mathcal{A}[\![\texttt{add}]\!] = \lambda i_1 i_2\,.\,(Value \rightarrowtail Result\ (Int \rightarrowtail Value\ (i_1 +_{Int} i_2)))$$

$$\mathcal{A}[\![\texttt{sub}]\!] = \lambda i_1 i_2\,.\,(Value \rightarrowtail Result\ (Int \rightarrowtail Value\ (i_1 -_{Int} i_2)))$$

$$\mathcal{A}[\![\texttt{mul}]\!] = \lambda i_1 i_2\,.\,(Value \rightarrowtail Result\ (Int \rightarrowtail Value\ (i_1 \times_{Int} i_2)))$$

$$
\begin{array}{l}
\mathcal{A}[\![\texttt{div}]\!] = \lambda i_1 i_2\,.\,\textbf{if}\ i_2 =_{Int} 0 \\
\qquad\qquad \textbf{then}\ errorResult \\
\qquad\qquad (Value \rightarrowtail Result\ (Int \rightarrowtail Value\ (i_1 \div_{Int} i_2)))\ \textbf{end}
\end{array}
$$

$$
\begin{array}{l}
\mathcal{A}[\![\texttt{rem}]\!] = \lambda i_1 i_2\,.\,\textbf{if}\ i_2 =_{Int} 0 \\
\qquad\qquad \textbf{then}\ errorResult \\
\qquad\qquad (Value \rightarrowtail Result\ (Int \rightarrowtail Value\ (i_1\ \%_{Int}\ i_2)))\ \textbf{end}
\end{array}
$$

$$\mathcal{R}[\![\texttt{lt}]\!] = <_{Int}$$

$$\mathcal{R}[\![\texttt{eq}]\!] = ==_{Int}$$

$$\mathcal{R}[\![\texttt{gt}]\!] = >_{Int}$$

$\mathcal{N}$ maps integer literals to the integer numbers that they denote.

$\mathcal{I}$ maps name literals, except for PostFix command names (reserved words), to the alphabetic strings that they denote.

**Semantic functions for PostFix**

$$
\begin{array}{l}
errorResult\ :\ Result = (Error \rightarrowtail Result\ \texttt{error}) \\
errorAnswer\ :\ Answer = (Error \rightarrowtail Answer\ \texttt{error}) \\
errorStack\ :\ Stack = (Error \rightarrowtail Stack\ \texttt{error}) \\
errorTransform\ :\ StackTransform = \lambda s\,.\,errorStack
\end{array}
$$

$$
\begin{array}{l}
push\ :\ Result \rightarrow\ StackTransform \\
= \lambda r\,.\,(\lambda s\,.\,\textbf{match}\ \langle r, s\rangle \\
\qquad \triangleright\ \langle (Value \rightarrowtail Result\ v), (Value^* \rightarrowtail Stack\ v^*)\rangle\ []\ (v\,.\,v^*) \\
\qquad \triangleright\ \textbf{else}\ errorStack\ \textbf{end})
\end{array}
$$

$$
\begin{array}{l}
pop\ :\ StackTransform \\
= \lambda s\,.\,\textbf{match}\ s \\
\quad \triangleright\ (Value^* \rightarrowtail Stack\ (v_{head}\,.\,v^*_{tail}))\ []\ (Value^* \rightarrowtail Stack\ v^*_{tail}) \\
\quad \triangleright\ \textbf{else}\ errorStack\ \textbf{end}
\end{array}
$$

$top : Stack \to Result$
$= \lambda s$ . **match** $s$
$\quad \triangleright\ (Value^* \rightarrowtail Stack\ (v_{head} \,.\, v^*_{tail}))\ [\!]\ (Value \rightarrowtail Result\ v_{head})$
$\quad \triangleright$ **else** *errorResult* **end**

$intAt : Int \to Stack \to Result$
$= \lambda is$ . **match** $s$
$\quad \triangleright\ (Value^* \rightarrowtail Stack\ v^*)[\!]$
$\quad\quad$ **if** $1 \leq_{Int} i$ **and** $i \leq_{Int} (length\ v^*)$
$\quad\quad\quad$ **then match** $(nth\ i\ v*)$
$\quad\quad\quad\quad \triangleright\ (Int \rightarrowtail Value\ i_{result})\ [\!]\ (Value \rightarrowtail Result\ (Int \rightarrowtail Value\ i_{result}))$
$\quad\quad\quad\quad \triangleright$ **else** *errorResult*
$\quad\quad\quad$ **else** *errorResult* **end**
$\quad\quad \triangleright$ **else** *ErrorResult* **end**

$arithop : (Int \to Int \to Result) \to StackTransform$
$= \lambda f$ . $(\lambda s$ . **match** $s$
$\quad\quad \triangleright (Value^* \rightarrowtail Stack\ ((Int \rightarrowtail Value\ i_1) \,.\, (Int \rightarrowtail Value\ i_2) \,.\, v^*_{rest}))[\!]$
$\quad\quad\ (push\ (f\ i_2\ i_1)\ v^*_{rest})$
$\quad\quad \triangleright$ **else** *errorStack* **end**)

$transform : Result \to StackTransform$
$= \lambda r$ . **match** $r$
$\quad \triangleright\ (Value \rightarrowtail Result\ (StackTransform \rightarrowtail Value\ t))\ [\!]\ t$
$\quad \triangleright$ **else** *errorTransform* **end**

$resToAns : Result \to Answer$
$= \lambda r$ . **match** $r$
$\quad \triangleright\ (Value \rightarrowtail Result\ (Int \rightarrowtail Value\ i))\ [\!]\ (Int \rightarrowtail Answer\ i)$
$\quad \triangleright$ **else** *ErrorAnswer* **end**

$bind : Ident \to Value \to Dictionary$
$= \lambda I_{bind} V D$ . $\lambda I_{ref}$ . $^-$**if** $I_{bind} = I_{ref}$ **then** $V$ **else** $(D\ I_{ref})$ **end**

**Solutions**

By applying the developed denotational semantics to the following PostFix programs, their meaning can be determined. The meaning of each program is an element of a function domain that maps the PostFix context domains to an answer that is either in the domain of integers or an error token.

**Question 2.a.i**

(IF ⟨(postfix 0 0 (for 11 (pop 2 add))), []⟩)
= ⟨0 (for 11 (pop 2 add)), []⟩
⇒ ⟨(for 11 (pop 2 add)), [0]⟩ [num]
⇒ ⟨11 pop 2 add (for 10 (pop 2 add)), [0]⟩ [for]
⇒ ⟨pop 2 add (for 10 (pop 2 add)), [11, 0]⟩ [num]
⇒ ⟨2 add (for 10 (pop 2 add)), [0]⟩ [pop]

⇒ ⟨add (for 10 (pop 2 add)), [2, 0]⟩ [num]
⇒ ⟨((for 10 (pop 2 add)), [2]⟩ [arithop]
⇒ ⟨10 pop 2 add (for 9 (pop 2 add)), [2]⟩ [for]
⇒ ⟨pop 2 add (for 9 (pop 2 add)), [10, 2]⟩ [num]
⇒ ⟨2 add (for 9 (pop 2 add)), [2]⟩ [pop]
⇒ ⟨add (for 9 (pop 2 add)), [2, 2]⟩ [num]
⇒ ⟨((for 9 (pop 2 add)), [4]⟩ [arithop]
⇒ ⟨9 pop 2 add (for 8 (pop 2 add)), [4]⟩ [for]
⇒ ⟨pop 2 add (for 8 (pop 2 add)), [9, 4]⟩ [num]
⇒ ⟨2 add (for 8 (pop 2 add)), [4]⟩ [pop]
⇒ ⟨add (for 8 (pop 2 add)), [2, 4]⟩ [num]
⇒ ⟨((for 8 (pop 2 add)), [6]⟩ [arithop]
⇒ ⟨8 pop 2 add (for 7 (pop 2 add)), [6]⟩ [for]
⇒ ⟨pop 2 add (for 7 (pop 2 add)), [8, 6]⟩ [num]
⇒ ⟨2 add (for 7 (pop 2 add)), [6]⟩ [pop]
⇒ ⟨add (for 7 (pop 2 add)), [2, 6]⟩ [num]
⇒ ⟨((for 7 (pop 2 add)), [8]⟩ [arithop]
⇒ ⟨7 pop 2 add (for 6 (pop 2 add)), [8]⟩ [for]
⇒ ⟨pop 2 add (for 6 (pop 2 add)), [7, 8]⟩ [num]
⇒ ⟨2 add (for 6 (pop 2 add)), [8]⟩ [pop]
⇒ ⟨add (for 6 (pop 2 add)), [2, 8]⟩ [num]
⇒ ⟨((for 6 (pop 2 add)), [10]⟩ [arithop]
⇒ ⟨6 pop 2 add (for 5 (pop 2 add)), [10]⟩ [for]
⇒ ⟨pop 2 add (for 5 (pop 2 add)), [6, 10]⟩ [num]
⇒ ⟨2 add (for 5 (pop 2 add)), [10]⟩ [pop]
⇒ ⟨add (for 5 (pop 2 add)), [2, 10]⟩ [num]
⇒ ⟨((for 5 (pop 2 add)), [12]⟩ [arithop]
⇒ ⟨5 pop 2 add (for 4 (pop 2 add)), [12]⟩ [for]
⇒ ⟨pop 2 add (for 4 (pop 2 add)), [5, 12]⟩ [num]
⇒ ⟨2 add (for 4 (pop 2 add)), [12]⟩ [pop]
⇒ ⟨add (for 4 (pop 2 add)), [2, 12]⟩ [num]
⇒ ⟨((for 4 (pop 2 add)), [14]⟩ [arithop]
⇒ ⟨4 pop 2 add (for 3 (pop 2 add)), [14]⟩ [for]
⇒ ⟨pop 2 add (for 3 (pop 2 add)), [4, 14]⟩ [num]
⇒ ⟨2 add (for 3 (pop 2 add)), [14]⟩ [pop]
⇒ ⟨add (for 3 (pop 2 add)), [2, 14]⟩ [num]
⇒ ⟨((for 3 (pop 2 add)), [16]⟩ [arithop]
⇒ ⟨3 pop 2 add (for 2 (pop 2 add)), [16]⟩ [for]
⇒ ⟨pop 2 add (for 2 (pop 2 add)), [3, 16]⟩ [num]
⇒ ⟨2 add (for 2 (pop 2 add)), [16]⟩ [pop]
⇒ ⟨add (for 2 (pop 2 add)), [2, 16]⟩ [num]
⇒ ⟨((for 2 (pop 2 add)), [18]⟩ [arithop]
⇒ ⟨2 pop 2 add (for 1 (pop 2 add)), [18]⟩ [for]
⇒ ⟨pop 2 add (for 1 (pop 2 add)), [2, 18]⟩ [num]

$\Rightarrow \langle$2 add (for 1 (pop 2 add)), [18]$\rangle$ [pop]
$\Rightarrow \langle$add (for 1 (pop 2 add)), [2, 18]$\rangle$ [num]
$\Rightarrow \langle$((for 1 (pop 2 add)), [20]$\rangle$ [arithop]
$\Rightarrow \langle$1 pop 2 add (for 0 (pop 2 add)), [20]$\rangle$ [for]
$\Rightarrow \langle$pop 2 add (for 0 (pop 2 add)), [1, 20]$\rangle$ [num]
$\Rightarrow \langle$2 add (for 0 (pop 2 add)), [20]$\rangle$ [pop]
$\Rightarrow \langle$add (for 0 (pop 2 add)), [2, 20]$\rangle$ [num]
$\Rightarrow \langle$((for 0 (pop 2 add)), [22]$\rangle$ [arithop]
$\Rightarrow \langle$(), [22]$\rangle \in$ FC [for]
$\Rightarrow$ (OF $\langle$(), [22]$\rangle$) = 22

**Question 2.a.ii**

(IF $\langle$((postfix 0 0 (for 6 (pop (for 7 (pop 1 add))))), []$\rangle$)
$= \langle$0 (for 6 (pop (for 7 (pop 1 add)))), []$\rangle$
$\Rightarrow \langle$((for 6 (pop (for 7 (pop 1 add)))), [0]$\rangle$ [num]
$\Rightarrow \langle$6 pop (for 7 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [0]$\rangle$ [for]
$\Rightarrow \langle$pop (for 7 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [6, 0]$\rangle$ [num]
$\Rightarrow \langle$((for 7 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [0]$\rangle$ [pop]
$\Rightarrow \langle$7 pop 1 add (for 6 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [0]$\rangle$ [for]
$\Rightarrow \langle$pop 1 add (for 6 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [7, 0]$\rangle$ [num]
$\Rightarrow \langle$1 add (for 6 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [0]$\rangle$ [pop]
$\Rightarrow \langle$add (for 6 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [1, 0]$\rangle$ [num]
$\Rightarrow \langle$((for 6 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [1]$\rangle$ [arithop]
$\Rightarrow \langle$6 pop 1 add (for 5 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [1]$\rangle$ [for]
$\Rightarrow \langle$pop 1 add (for 5 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [6, 1]$\rangle$ [num]
$\Rightarrow \langle$1 add (for 5 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [1]$\rangle$ [pop]
$\Rightarrow \langle$add (for 5 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [1, 1]$\rangle$ [num]
$\Rightarrow \langle$((for 5 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [2]$\rangle$ [arithop]
$\Rightarrow \langle$5 pop 1 add (for 4 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [2]$\rangle$ [for]
$\Rightarrow \langle$pop 1 add (for 4 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [5, 2]$\rangle$ [num]
$\Rightarrow \langle$1 add (for 4 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [2]$\rangle$ [pop]
$\Rightarrow \langle$add (for 4 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [1, 2]$\rangle$ [num]
$\Rightarrow \langle$((for 4 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [3]$\rangle$ [arithop]
$\Rightarrow \langle$4 pop 1 add (for 3 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [3]$\rangle$ [for]
$\Rightarrow \langle$pop 1 add (for 3 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [4, 3]$\rangle$ [num]
$\Rightarrow \langle$1 add (for 3 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [3]$\rangle$ [pop]
$\Rightarrow \langle$add (for 3 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [1, 3]$\rangle$ [num]
$\Rightarrow \langle$((for 3 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [4]$\rangle$ [arithop]
$\Rightarrow \langle$3 pop 1 add (for 2 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [4]$\rangle$ [for]
$\Rightarrow \langle$pop 1 add (for 2 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [3, 4]$\rangle$ [num]
$\Rightarrow \langle$1 add (for 2 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [4]$\rangle$ [pop]
$\Rightarrow \langle$add (for 2 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [1, 4]$\rangle$ [num]
$\Rightarrow \langle$((for 2 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [5]$\rangle$ [arithop]
$\Rightarrow \langle$2 pop 1 add (for 1 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [5]$\rangle$ [for]
$\Rightarrow \langle$pop 1 add (for 1 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [2, 5]$\rangle$ [num]

⇒ ⟨1 add (for 1 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [5]⟩ [pop]
⇒ ⟨add (for 1 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [1, 5]⟩ [num]
⇒ ⟨(for 1 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [6]⟩ [arithop]
⇒ ⟨1 pop 1 add (for 0 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [6]⟩ [for]
⇒ ⟨pop 1 add (for 0 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [1, 6]⟩ [num]
⇒ ⟨1 add (for 0 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [6]⟩ [pop]
⇒ ⟨add (for 0 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [1, 6]⟩ [num]
⇒ ⟨(for 0 (pop 1 add)) (for 5 (pop (for 7 (pop 1 add)))), [7]⟩ [arithop]
⇒ ⟨(for 5 (pop (for 7 (pop 1 add)))), [7]⟩ [for]
⇒ ⟨5 pop (for 7 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [7]⟩ [for]
⇒ ⟨pop (for 7 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [5, 7]⟩ [num]
⇒ ⟨(for 7 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [7]⟩ [pop]
⇒ ⟨7 pop 1 add (for 6 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [7]⟩ [for]
⇒ ⟨pop 1 add (for 6 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [7, 7]⟩ [num]
⇒ ⟨1 add (for 6 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [7]⟩ [pop]
⇒ ⟨add (for 6 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [1, 7]⟩ [num]
⇒ ⟨(for 6 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [8]⟩ [arithop]
⇒ ⟨6 pop 1 add (for 5 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [8]⟩ [for]
⇒ ⟨pop 1 add (for 5 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [6, 8]⟩ [num]
⇒ ⟨1 add (for 5 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [8]⟩ [pop]
⇒ ⟨add (for 5 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [1, 8]⟩ [num]
⇒ ⟨(for 5 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [9]⟩ [arithop]
⇒ ⟨5 pop 1 add (for 4 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [9]⟩ [for]
⇒ ⟨pop 1 add (for 4 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [5, 9]⟩ [num]
⇒ ⟨1 add (for 4 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [9]⟩ [pop]
⇒ ⟨add (for 4 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [1, 9]⟩ [num]
⇒ ⟨(for 4 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [10]⟩ [arithop]
⇒ ⟨4 pop 1 add (for 3 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [10]⟩ [for]
⇒ ⟨pop 1 add (for 3 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [4, 10]⟩ [num]
⇒ ⟨1 add (for 3 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [10]⟩ [pop]
⇒ ⟨add (for 3 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [1, 10]⟩ [num]
⇒ ⟨(for 3 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [11]⟩ [arithop]
⇒ ⟨3 pop 1 add (for 2 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [11]⟩ [for]
⇒ ⟨pop 1 add (for 2 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [3, 11]⟩ [num]
⇒ ⟨1 add (for 2 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [11]⟩ [pop]
⇒ ⟨add (for 2 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [1, 11]⟩ [num]
⇒ ⟨(for 2 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [12]⟩ [arithop]
⇒ ⟨2 pop 1 add (for 1 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [12]⟩ [for]
⇒ ⟨pop 1 add (for 1 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [2, 12]⟩ [num]
⇒ ⟨1 add (for 1 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [12]⟩ [pop]
⇒ ⟨add (for 1 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [1, 12]⟩ [num]
⇒ ⟨(for 1 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [13]⟩ [arithop]
⇒ ⟨1 pop 1 add (for 0 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [13]⟩ [for]
⇒ ⟨pop 1 add (for 0 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [1, 13]⟩ [num]

⇒ ⟨1 add (for 0 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [13]⟩ [pop]
⇒ ⟨add (for 0 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [1, 13]⟩ [num]
⇒ ⟨(for 0 (pop 1 add)) (for 4 (pop (for 7 (pop 1 add)))), [14]⟩ [arithop]
⇒ ⟨(for 4 (pop (for 7 (pop 1 add)))), [14]⟩ [for]
⇒ ⟨4 pop (for 7 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [14]⟩ [for]
⇒ ⟨pop (for 7 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [4, 14]⟩ [num]
⇒ ⟨(for 7 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [14]⟩ [pop]
⇒ ⟨7 pop 1 add (for 6 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [14]⟩ [for]
⇒ ⟨pop 1 add (for 6 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [7, 14]⟩ [num]
⇒ ⟨1 add (for 6 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [14]⟩ [pop]
⇒ ⟨add (for 6 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [1, 14]⟩ [num]
⇒ ⟨(for 6 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [15]⟩ [arithop]
⇒ ⟨6 pop 1 add (for 5 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [15]⟩ [for]
⇒ ⟨pop 1 add (for 5 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [6, 15]⟩ [num]
⇒ ⟨1 add (for 5 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [15]⟩ [pop]
⇒ ⟨add (for 5 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [1, 15]⟩ [num]
⇒ ⟨(for 5 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [16]⟩ [arithop]
⇒ ⟨5 pop 1 add (for 4 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [16]⟩ [for]
⇒ ⟨pop 1 add (for 4 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [5, 16]⟩ [num]
⇒ ⟨1 add (for 4 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [16]⟩ [pop]
⇒ ⟨add (for 4 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [1, 16]⟩ [num]
⇒ ⟨(for 4 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [17]⟩ [arithop]
⇒ ⟨4 pop 1 add (for 3 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [17]⟩ [for]
⇒ ⟨pop 1 add (for 3 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [4, 17]⟩ [num]
⇒ ⟨1 add (for 3 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [17]⟩ [pop]
⇒ ⟨add (for 3 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [1, 17]⟩ [num]
⇒ ⟨(for 3 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [18]⟩ [arithop]
⇒ ⟨3 pop 1 add (for 2 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [18]⟩ [for]
⇒ ⟨pop 1 add (for 2 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [3, 18]⟩ [num]
⇒ ⟨1 add (for 2 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [18]⟩ [pop]
⇒ ⟨add (for 2 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [1, 18]⟩ [num]
⇒ ⟨(for 2 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [19]⟩ [arithop]
⇒ ⟨2 pop 1 add (for 1 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [19]⟩ [for]
⇒ ⟨pop 1 add (for 1 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [2, 19]⟩ [num]
⇒ ⟨1 add (for 1 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [19]⟩ [pop]
⇒ ⟨add (for 1 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [1, 19]⟩ [num]
⇒ ⟨(for 1 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [20]⟩ [arithop]
⇒ ⟨1 pop 1 add (for 0 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [20]⟩ [for]
⇒ ⟨pop 1 add (for 0 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [1, 20]⟩ [num]
⇒ ⟨1 add (for 0 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [20]⟩ [pop]
⇒ ⟨add (for 0 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [1, 20]⟩ [num]
⇒ ⟨(for 0 (pop 1 add)) (for 3 (pop (for 7 (pop 1 add)))), [21]⟩ [arithop]
⇒ ⟨(for 3 (pop (for 7 (pop 1 add)))), [21]⟩ [for]
⇒ ⟨3 pop (for 7 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [21]⟩ [for]

⇒ ⟨pop (for 7 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [3, 21]⟩ [num]
⇒ ⟨((for 7 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [21]⟩ [pop]
⇒ ⟨7 pop 1 add (for 6 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [21]⟩ [for]
⇒ ⟨pop 1 add (for 6 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [7, 21]⟩ [num]
⇒ ⟨1 add (for 6 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [21]⟩ [pop]
⇒ ⟨add (for 6 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [1, 21]⟩ [num]
⇒ ⟨((for 6 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [22]⟩ [arithop]
⇒ ⟨6 pop 1 add (for 5 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [22]⟩ [for]
⇒ ⟨pop 1 add (for 5 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [6, 22]⟩ [num]
⇒ ⟨1 add (for 5 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [22]⟩ [pop]
⇒ ⟨add (for 5 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [1, 22]⟩ [num]
⇒ ⟨((for 5 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [23]⟩ [arithop]
⇒ ⟨5 pop 1 add (for 4 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [23]⟩ [for]
⇒ ⟨pop 1 add (for 4 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [5, 23]⟩ [num]
⇒ ⟨1 add (for 4 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [23]⟩ [pop]
⇒ ⟨add (for 4 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [1, 23]⟩ [num]
⇒ ⟨((for 4 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [24]⟩ [arithop]
⇒ ⟨4 pop 1 add (for 3 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [24]⟩ [for]
⇒ ⟨pop 1 add (for 3 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [4, 24]⟩ [num]
⇒ ⟨1 add (for 3 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [24]⟩ [pop]
⇒ ⟨add (for 3 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [1, 24]⟩ [num]
⇒ ⟨((for 3 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [25]⟩ [arithop]
⇒ ⟨3 pop 1 add (for 2 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [25]⟩ [for]
⇒ ⟨pop 1 add (for 2 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [3, 25]⟩ [num]
⇒ ⟨1 add (for 2 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [25]⟩ [pop]
⇒ ⟨add (for 2 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [1, 25]⟩ [num]
⇒ ⟨((for 2 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [26]⟩ [arithop]
⇒ ⟨2 pop 1 add (for 1 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [26]⟩ [for]
⇒ ⟨pop 1 add (for 1 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [2, 26]⟩ [num]
⇒ ⟨1 add (for 1 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [26]⟩ [pop]
⇒ ⟨add (for 1 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [1, 26]⟩ [num]
⇒ ⟨((for 1 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [27]⟩ [arithop]
⇒ ⟨1 pop 1 add (for 0 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [27]⟩ [for]
⇒ ⟨pop 1 add (for 0 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [27]⟩ [num]
⇒ ⟨1 add (for 0 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [27]⟩ [pop]
⇒ ⟨add (for 0 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [1, 27]⟩ [num]
⇒ ⟨((for 0 (pop 1 add)) (for 2 (pop (for 7 (pop 1 add)))), [28]⟩ [arithop]
⇒ ⟨((for 2 (pop (for 7 (pop 1 add)))), [28]⟩ [for]
⇒ ⟨2 pop (for 7 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [28]⟩ [for]
⇒ ⟨pop (for 7 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [2, 28]⟩ [num]
⇒ ⟨((for 7 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [28]⟩ [pop]
⇒ ⟨7 pop 1 add (for 6 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [28]⟩ [for]
⇒ ⟨pop 1 add (for 6 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [7, 28]⟩ [num]
⇒ ⟨1 add (for 6 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [28]⟩ [pop]

⇒ ⟨add (for 6 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [1, 28]⟩ [num]
⇒ ⟨((for 6 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [29]⟩ [arithop]
⇒ ⟨6 pop 1 add (for 5 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [29]⟩ [for]
⇒ ⟨pop 1 add (for 5 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [6, 29]⟩ [num]
⇒ ⟨1 add (for 5 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [29]⟩ [pop]
⇒ ⟨add (for 5 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [1, 29]⟩ [num]
⇒ ⟨((for 5 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [30]⟩ [arithop]
⇒ ⟨5 pop 1 add (for 4 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [30]⟩ [for]
⇒ ⟨pop 1 add (for 4 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [5, 30]⟩ [num]
⇒ ⟨1 add (for 4 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [30]⟩ [pop]
⇒ ⟨add (for 4 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [1, 30]⟩ [num]
⇒ ⟨((for 4 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [31]⟩ [arithop]
⇒ ⟨4 pop 1 add (for 3 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [31]⟩ [for]
⇒ ⟨pop 1 add (for 3 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [4, 31]⟩ [num]
⇒ ⟨1 add (for 3 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [31]⟩ [pop]
⇒ ⟨add (for 3 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [1, 31]⟩ [num]
⇒ ⟨((for 3 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [32]⟩ [arithop]
⇒ ⟨3 pop 1 add (for 2 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [32]⟩ [for]
⇒ ⟨pop 1 add (for 2 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [3, 32]⟩ [num]
⇒ ⟨1 add (for 2 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [32]⟩ [pop]
⇒ ⟨add (for 2 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [1, 32]⟩ [num]
⇒ ⟨((for 2 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [33]⟩ [arithop]
⇒ ⟨2 pop 1 add (for 1 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [33]⟩ [for]
⇒ ⟨pop 1 add (for 1 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [2, 33]⟩ [num]
⇒ ⟨1 add (for 1 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [33]⟩ [pop]
⇒ ⟨add (for 1 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [1, 33]⟩ [num]
⇒ ⟨((for 1 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [34]⟩ [arithop]
⇒ ⟨1 pop 1 add (for 0 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [34]⟩ [for]
⇒ ⟨pop 1 add (for 0 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [1, 34]⟩ [num]
⇒ ⟨1 add (for 0 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [34]⟩ [pop]
⇒ ⟨add (for 0 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [1, 34]⟩ [num]
⇒ ⟨((for 0 (pop 1 add)) (for 1 (pop (for 7 (pop 1 add)))), [35]⟩ [arithop]
⇒ ⟨((for 1 (pop (for 7 (pop 1 add)))), [35]⟩ [for]
⇒ ⟨1 pop (for 7 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [35]⟩ [for]
⇒ ⟨pop (for 7 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [1, 35]⟩ [num]
⇒ ⟨((for 7 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [35]⟩ [pop]
⇒ ⟨7 pop 1 add (for 6 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [35]⟩ [for]
⇒ ⟨pop 1 add (for 6 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [7, 35]⟩ [num]
⇒ ⟨1 add (for 6 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [35]⟩ [pop]
⇒ ⟨add (for 6 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [1, 35]⟩ [num]
⇒ ⟨((for 6 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [36]⟩ [arithop]
⇒ ⟨6 pop 1 add (for 5 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [36]⟩ [for]
⇒ ⟨pop 1 add (for 5 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [6, 36]⟩ [num]
⇒ ⟨1 add (for 5 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [36]⟩ [pop]

$\Rightarrow \langle$add (for 5 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [1, 36]$\rangle$ [num]
$\Rightarrow \langle$((for 5 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [37]$\rangle$ [arithop]
$\Rightarrow \langle$5 pop 1 add (for 4 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [37]$\rangle$ [for]
$\Rightarrow \langle$pop 1 add (for 4 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [5, 37]$\rangle$ [num]
$\Rightarrow \langle$1 add (for 4 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [37]$\rangle$ [pop]
$\Rightarrow \langle$add (for 4 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [1, 37]$\rangle$ [num]
$\Rightarrow \langle$((for 4 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [38]$\rangle$ [arithop]
$\Rightarrow \langle$4 pop 1 add (for 0 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [38]$\rangle$ [for]
$\Rightarrow \langle$pop 1 add (for 3 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [4, 38]$\rangle$ [num]
$\Rightarrow \langle$1 add (for 3 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [38]$\rangle$ [pop]
$\Rightarrow \langle$add (for 3 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [1, 38]$\rangle$ [num]
$\Rightarrow \langle$((for 3 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [39]$\rangle$ [arithop]
$\Rightarrow \langle$3 pop 1 add (for 2 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [39]$\rangle$ [for]
$\Rightarrow \langle$pop 1 add (for 2 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [3, 39]$\rangle$ [num]
$\Rightarrow \langle$1 add (for 2 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [39]$\rangle$ [pop]
$\Rightarrow \langle$add (for 2 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [1, 39]$\rangle$ [num]
$\Rightarrow \langle$((for 2 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [40]$\rangle$ [arithop]
$\Rightarrow \langle$2 pop 1 add (for 1 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [40]$\rangle$ [for]
$\Rightarrow \langle$pop 1 add (for 1 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [2, 40]$\rangle$ [num]
$\Rightarrow \langle$1 add (for 1 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [40]$\rangle$ [pop]
$\Rightarrow \langle$add (for 1 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [1, 40]$\rangle$ [num]
$\Rightarrow \langle$((for 1 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [41]$\rangle$ [arithop]
$\Rightarrow \langle$1 pop 1 add (for 0 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [41]$\rangle$ [for]
$\Rightarrow \langle$pop 1 add (for 0 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [1, 41]$\rangle$ [num]
$\Rightarrow \langle$1 add (for 0 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [41]$\rangle$ [pop]
$\Rightarrow \langle$add (for 0 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [1, 41]$\rangle$ [num]
$\Rightarrow \langle$((for 0 (pop 1 add)) (for 0 (pop (for 7 (pop 1 add)))), [42]$\rangle$ [arithop]
$\Rightarrow \langle$((for 0 (pop (for 7 (pop 1 add)))), [42]$\rangle$ [for]
$\Rightarrow \langle$(), [42]$\rangle \in$ FC [for]
$\Rightarrow$ (OF $\langle$(), [42]$\rangle$) = 42

**Question 2.b**

**Proof:** To prove the termination property of PostFix does not change, we show that every transition reduces the energy of a configuration. First, we know that the energy of a configuration, sequence, or stack is greater than or equal to the sum of the energy of its components. By considering the rewrite rules of *for*, it is possible to see that the energy strictly decreases with each transition. The energy of the *for* command is defined as follows:

$$
\begin{aligned}
& E_{config}\ [\![\langle(\texttt{for}\ \texttt{N}(\texttt{Q}_{\texttt{for}}))\ .\ \texttt{Q}_{\texttt{rest}}, \texttt{S}\rangle]\!] \\
&= E_{seq}\ [\![\langle \texttt{for}\ .\ \texttt{N}\ .\ (\texttt{Q}_{\texttt{for}})]\!] + E_{stack}\ [\![\texttt{Q}_{\texttt{rest}}\ .\ \texttt{S}]\!] \\
&= 1 + E_{com}\ [\![\texttt{for}]\!] + E_{com}\ [\![\texttt{N}]\!] + E_{com}\ [\![(\texttt{Q}_{\texttt{for}})]\!] + E_{seq}\ [\![\texttt{Q}_{\texttt{rest}}]\!] + E_{stack}\ [\![\texttt{S}]\!] \\
&= 2 + E_{com}\ [\![\texttt{N}]\!] + E_{seq}\ [\![\texttt{Q}_{\texttt{for}}]\!] + E_{seq}\ [\![\texttt{Q}_{\texttt{rest}}]\!] + E_{stack}\ [\![\texttt{S}]\!] \\
&= 2 + E_{config}\ [\![\langle \texttt{N}\ .\ \texttt{Q}_{\texttt{for}}\ @\ ((\texttt{N} - 1)\ .\ \texttt{Q}_{\texttt{for}}), \texttt{S}\rangle]\!]
\end{aligned}
$$

$E_{config}\ [\![\langle(\texttt{for N}(\texttt{Q}_{\texttt{for}}))\ .\ \texttt{Q}_{\texttt{rest}}, \texttt{S}\rangle]\!]$
$= E_{seq}\ [\![\langle \texttt{for}\ .\ \texttt{N}\ .\ (\texttt{Q}_{\texttt{for}})]\!] + E_{stack}\ [\![\texttt{Q}_{\texttt{rest}}\ .\ \texttt{S}]\!]$
$= 1 + E_{com}\ [\![\texttt{for}]\!] + E_{com}\ [\![\texttt{N}]\!] + E_{com}\ [\![(\texttt{Q}_{\texttt{for}})]\!] + E_{seq}\ [\![\texttt{Q}_{\texttt{rest}}]\!] + E_{stack}\ [\![\texttt{S}]\!]$
$= 2 + E_{com}\ [\![\texttt{N}]\!] + E_{seq}\ [\![\texttt{Q}_{\texttt{for}}]\!] + E_{seq}\ [\![\texttt{Q}_{\texttt{rest}}]\!] + E_{stack}\ [\![\texttt{S}]\!]$
$= 2 + E_{seq}\ [\![\texttt{Q}_{\texttt{for}}]\!] + E_{seq}\ [\![\texttt{Q}_{\texttt{rest}}]\!] + E_{stack}\ [\![\texttt{S}]\!]$
$= 2 + E_{config}\ [\![\langle\texttt{Q}_{\texttt{rest}}, \texttt{S}\rangle]\!]$

Thus, since we know that executing a command consumes at least one unit of energy, the termination property of PostFix does not change with the introduction of the *for* command.

**Question 3.a.i**
The following program returns the average of the values 3 and 7, which is 5.
(postfix 0 average (add 2 div) def 3 7 average ref exec) $\xrightarrow{[\,]}$ average (add 2 div) def 3 7 average ref exec
⇒ average (add 2 div) def 3 7 average ref exec → [ ], [ ]
⇒ (add 2 div) def 3 7 average ref exec → [average], [ ] [ident]
⇒ def 3 7 average ref exec → [(add 2 div)], [ ] [seq]
⇒ 3 7 average ref exec → [ ], [(average = (add 2 div))] [def]
⇒ 7 average ref exec → [3], [(average = (add 2 div))] [num]
⇒ average ref exec → [7, 3], [(average = (add 2 div))] [num]
⇒ ref exec → [average, 7, 3], [(average = (add 2 div))] [ident]
⇒ exec → [(add 2 div), 7, 3], [(average = (add 2 div))] [ref]
⇒ ( ) → [5], [(average = (add 2 div))] [exec]
= 5

**Question 3.a.ii**
The following program returns an error on the second *def* command since the second value on the stack, 5, is not an identifier.
(postfix 0 a 5 def a ref 7 def b ref) $\xrightarrow{[\,]}$ a 5 def a ref 7 def b ref
⇒ a 5 def a ref 7 def b ref → [ ], [ ]
⇒ 5 def a ref 7 def b ref → [a], [ ] [ident]
⇒ def a ref 7 def b ref → [5, a], [ ] [num]
⇒ a ref 7 def b ref → [ ], [(a = 5)] [def]
⇒ ref 7 def b ref → [a], [(a = 5)] [ident]
⇒ 7 def b ref → [5], [(a = 5)] [ref]
⇒ def b ref → [7, 5], [(a = 5)] [num]
= **error**

**Question 3.a.iii**
The following program returns an error on the *ref* command since the first value on the stack, d, is not bound in the dictionary.

(postfix 0 c 4 def d ref 1 add) $\xrightarrow{[\,]}$ c 4 def d ref 1 add
$\Rightarrow$ c 4 def d ref 1 add $\rightarrow$ [ ], [ ]
$\Rightarrow$ 4 def d ref 1 add $\rightarrow$ [c], [ ] [ident]
$\Rightarrow$ def d ref 1 add $\rightarrow$ [4, c], [ ] [num]
$\Rightarrow$ d ref 1 add $\rightarrow$ [ ], [(c = 4)] [def]
$\Rightarrow$ ref 1 add $\rightarrow$ [d], [(c = 4)] [ident]
= **error**