Here is one way that we might consider defining the \textbf{for}-loop. Let \( j, n, k \) and \( m \) be integer expressions, and let \( i \) be a fresh name.

\[
\text{for } i := \text{nil} \text{ do } P \text{ od} = \text{ok}
\]

\[
\text{for } i := j \text{ do } P \text{ od} = \text{(substitute } j \text{ for } i \text{ in } P \text{)}
\]

\[
\text{for } i := n;..k ; k;..m \text{ do } P \text{ od} = \text{for } i := n;..k \text{ do } P \text{ od, for } i := k;..m \text{ do } P \text{ od}
\]

(a) From this definition, what can we prove about \textbf{for } i := 0;..n \text{ do } n := n + 1 \text{ od} where \( n \) is an integer variable?

(b) What kinds of \textbf{for}-loop are in the programming languages you know?