In the exercises that have class composition with ; , you must use (p.14) before you use (p.18), and (p.18) before you assume any conjunct of the antecedent.

1. Prove that the following three statements swap the values of x and y by showing that the original value of variable x is indeed rigid variable X and the same for y.

   ```c
   int x, y, t
   { ? } t := x; x := y; y := t {x = Y ∧ y = X }
   ```

   In each of the following exercises, calculate the expression E so that the program meets its specification. Begin each calculation with \textit{wp}.\textit{S}.\textit{post} and write the final program with preconditions and postconditions after your calculation. To save writing, you can abbreviate the invariant. For example, in 2. you can define \textit{P1}: \( x = (\Sigma k \mid i \leq k < n : b[k]) \) and then use \textit{P1} in your proof and the final statement of your program. You must state your definition of \textit{P1} in each exercise in which you use this technique.

2. \textit{const int n}
   \textit{int i, x, b[n]}
   \{ \textit{x} = (\Sigma k \mid i \leq k < n : b[k]) \}
   \textit{i, x} := i - 1, \textit{E}
   \{ \textit{x} = (\Sigma k \mid i \leq k < n : b[k]) \}

3. Caution: In this exercise, the unknown expression is in the second assignment statement. So, you must use the technique described here.
   \textit{const int n}
   \textit{int i, x, b[n]}
   \{ \textit{x} = (\Sigma k \mid 0 \leq k < i : b[k]) \}
   \textit{i} := i + 1; \textit{x} := \textit{E}
   \{ \textit{x} = (\Sigma k \mid 0 \leq k < i : b[k]) \}

4. \textit{const int n}
   \textit{int i, x, b[n]}
   \{ \textit{x} = (\Sigma k \mid 0 \leq k < i : b[k]) \}
   \textit{x} := \textit{E}; \textit{i} := i + 1
   \{ \textit{x} = (\Sigma k \mid 0 \leq k < i : b[k]) \}