Given two lists, write a program to find the minimum number of item insertions, item deletions, and item replacements to change one list into the other.

§ Here is the standard solution, which uses for-loops (Subsection 5.2.3), but for-loops are never necessary. We will change list \( A \) into list \( B \). Let \( D: [ (#A + 1) \times (#B + 1) \times \text{nat}] \) be an array-valued variable whose final value will be such that \( D' \ i \ j = (\text{the edit distance from } A[0..i] \text{ to } B[0..j]) \). So the final answer will be \( D' (#A) (#B) \).

(0) \hspace{1em} \text{for } i := 0;..#A+1 \text{ do } D i 0 := i \text{ od}.
(1) \hspace{1em} \text{for } j := 1;..#B+1 \text{ do } D 0 j := j \text{ od}.
(2) \hspace{1em} \text{for } i := 1;..#A+1 \text{ do }
(3) \hspace{2em} \text{for } j := 1;..#B+1 \text{ do }
(4) \hspace{3em} D i j := \min(D (i-1) (j-1) + \text{if } Ai=Bj \text{ then } 0 \text{ else } 1 \text{ fi})
(5) \hspace{3em} (\min(D (i-1) j + 1)) \text{ od}
(6) \hspace{2em} (D i (j-1) + 1)) \text{ od od}

Line 0 says \( A[0..i] \) can be changed to \([\text{nil}]\) by \( i \) deletions.
Line 1 says \([\text{nil}]\) can be changed to \( B[0..j] \) by \( j \) insertions.
Lines 2 and 3 fill in the interior of \( D \).
On line 4, if we can transform \( A[0..i-1] \) to \( B[0..j-1] \) in \( D (i-1) (j-1) \) steps, and if \( Ai=Bj \), we have transformed \( A[0..i] \) to \( B[0..j] \). But if \( Ai\neq Bj \) then we need to replace \( Ai \) by \( Bj \) which takes 1 step.
On line 5, if we can transform \( A[0..i-1] \) to \( B[0..j] \) in \( D (i-1) j \) steps, then we can transform \( A[0..i] \) to \( B[0..j] \) by deleting \( Ai \).
On line 6, if we can transform \( A[0..i] \) to \( B[0..j-1] \) in \( D i (j-1) \) steps, then we can transform \( A[0..i] \) to \( B[0..j] \) by appending \( Bj \).
The shortest way to transform \( A[0..i] \) to \( B[0..j] \) is the minimum of the three ways from lines 4, 5, and 6.

To prove the correctness of this solution, find invariants for the for-loops. Or eliminate the for-loops and write specifications as in Chapter 4.