```
//*************************************************
// TITLE: DynLlcs.java                            *
// AUTHOR: Kim Skak Larsen                        *
// DATE: 26/11 2001                               *
//                                                *
// Reads two strings from stdin and computes the  *
// lengths of the longest common subsequence using *
// dynamic programming.                           *
//*************************************************

import cs1.Keyboard;

public class DynLlcs
{
   static int llcsw(String s, String t, int i, int j, int[][] table)
   {
      if (table[i][j] == -1)
         if (i == s.length() || j == t.length())
            table[i][j] = 0;
         else
            if (s.charAt(i) == t.charAt(j))
               table[i][j] = 1 + llcsw(s, t, i+1, j+1, table);
            else
               table[i][j] = Math.max(llcsw(s, t, i+1, j, table),
                                      llcsw(s, t, i, j+1, table));
      return table[i][j];
   } // llcsw

   static int llcs(String s, String t)
   {
      int[][] table = new int[s.length()+1][t.length()+1];
      for (int i = 0; i < s.length()+1; i++)
         for (int j = 0; j < t.length()+1; j++)
            table[i][j] = -1;
      return llcsw(s, t, 0, 0, table);
   } // llcs

   public static void main(String[] args)
   {
      String s, t;

      System.out.print("1. string: ");
      s = Keyboard.readString();
      System.out.print("2. string: ");
      t = Keyboard.readString();
      System.out.print("llcs(" + s + "," + t + ") = "
                       + llcs(s, t) + "\n");
```

```
    } // main

} // DynLlcs
```