Name: Taylor Heilman and Kevin Benson

Mon. May 2

CS 271: Project 1001

1. a.) The structure of an optimal solution to $f(m, n)$ would be a palindromic string of max length.

   b.) $f(m, n)$ represents the length of the longest palindrome subsequence from m^{th} character to the n^{th} character

$$f(m,n) \begin{cases} 1, & m = n \\ f(m+1, n-1) + 2, & a_m = a_n \\ max[f(m+1, n), f(m, n-1)] & else \end{cases}$$

   c.)

   Create a square array with dimensions ixj where both i and j are equal to the length of the input string.

   Make the diagonal indices = 1 (since a string of length 1 is a palindrome of itself) and all other indices = 0.

   Iterate through all columns and examine every item $c[i][j]$ above the diagonal. If the i^{th} and j^{th} characters in the input string are identical, the solution to subproblem f(i,j) is two greater than the solution to subproblem f(i+1, j-1), which is stored at index [i-1][j+1].

   If this is not the case, the value of the solution is the greater of either the item to the left or the item below the item in question.

   Our c++ implementation of this algorithm is provided below.