

CSCI 135 Software Design and Analysis I

Homework 6

Due 11/14/08

Saad Mneimneh
Visiting Professor
Hunter College of CUNY

Problem 1

Do labs 9 (input and output strings) and 11 (two dimensional arrays).

Problem 2: Recursion with Palindromes

A palindrome is a string that spells the same forwards and backwards, e.g. “radar”. Given a string s , we would like to determine if $s[i] \dots s[j]$ is a palindrome.

Let us first find a recursive definition of palindrome. A string $s[i] \dots s[j]$ is a palindrome if $s[i+1] \dots s[j-1]$ is a palindrome and $s[i] = s[j]$. Therefore

$$palindrome(s, i, j) = \begin{cases} (s[i] == s[j]) \text{ and } palindrome(s, i+1, j-1) & j > i \\ true & j \leq i \end{cases}$$

- Write a recursive function for the above definition to determine whether a string is a palindrome or not.
- The definition above is not tail recursive because after the function call to `palindrome` returns, the result must be ANDed with $s[i] == s[j]$. Using an extra **if**, transform the recursion into a tail recursion.
- Transform the tail recursive implementation into an iterative one.
- With the help of the function `palindrome` in (c), write a function (call it `LargestPalindrome`) to find the largest palindrome contained in a string, using 3 nested loops (including the loop in function `palindrome`).
- The `palindrome` function in (c) works inward by checking extremities. Write another function `palindrome2` that finds a palindrome in a string by working outward given the center of the palindrome. Make it return the length of the largest such palindrome. Use it to solve (d) using only 2 nested loops (including the loop in the function `palindrome2`).