# Palindromes

A *palindrome* is a string that is the same forward and backward. In Chapter 5 you saw a program that uses a loop to determine whether a string is a palindrome. However, it is also easy to define a palindrome recursively as follows:

- A string containing fewer than 2 letters is always a palindrome.
- A string containing 2 or more letters is a palindrome if
  - its first and last letters are the same, and
  - the rest of the string (without the first and last letters) is also a palindrome.

Write a program that prompts for and reads in a string, then prints a message saying whether it is a palindrome. Your main method should read the string and call a recursive (static) method *palindrome* that takes a string and returns true if the string is a palindrome, false otherwise. Recall that for a string *s* in Java,

- *s.length()* returns the number of charaters in *s*
- *s.charAt(i)* returns the $i^{th}$ character of *s*, 0-based
- *s.substring(i,j)* returns the substring that starts with the $i^{th}$ character of *s* and ends with the $j–1^{st}$ character of *s* (not the $j^{th}$), both 0-based.

So if *s*="happy", *s.length*=5, *s.charAt(1)*=a, and *s.substring(2,4)* = "pp".