Exercises 21

recursion

Exercise 21.1 - Recursive subprograms

1. Write a recursive subroutine that calculates the sum of the first n squares. For example, if n = 3, this sub-program will calculate $1^2 + 2^2 + 3^2$. This subroutine is only defined for n greater than 0.
2. Write a recursive subroutine which calculates the sum of the positive elements of an array.
3. Write a recursive routine that checks whether a string is a palindrome.
Reminder: A palindrome is a word that reads the same from left to right and right to left, such as **radar** or **ada**. To do this you use the methods `charAt` () and `length`() of class; `String.s.charAt` (i) returns the i-th character of the string s and `s.length`() returns the length of s.
4. Write a recursive subroutine that rearranges the elements of an array in reverse order.
5. Write a recursive subroutine which calculates the value of a string of digital digits.

If it helps, you can start by looking for a formula that expresses the recursive calculation in general.

Exercise 21.2 - Fibonacci

Write a function that calculates the values of the Fibonacci series, which is defined by:

$u_0 = 0$

$u_1 = 1$

$u_n = u_{n-1} + u_{n-2}$

Write this function as iterative and recursive form.
Which of the two variants is better here?