Lab04 - Recursion (and not)

Due: Wed Apr 22, 2015

Let's solve a problem recursively and then not. Or vise versa.
- ❏ Chapter 15 Recursion

Palindromes

Complete Programming Challenge 5 on page 951 twice… **recursively**, and then **iteratively**. Lab04 is meatball surgery. Create a class for checking palindromes with three static methods: `main()`, `isPalindrome()` and `isPalindromeIterative()`.

You main() pseudocode is:

> *main() {*
>> *Print a welcome (to Lab04) message*
>> *Ask the user for a word*
>> *Report if it's a palindrome twice… once recursively and once iteratively*
>
> *}*

For each palindrome method, write down the signature and some pseudocode. For the recursive method, what is your base case? **Show me first**… then proceed to your toaster and code it up.

BTW, here's a fun palindrome site to try: www.palindromelist.net

Fibonacci

Create a class `FibNumbers`, just like on page 941 in our text. You can copy it from my k: drive.
Two changes:
- Change `main()` so that you ask the user for a number N, and then print Fib(N). Go!
- Change `fib()` to return a `long` rather than an `int`. This way we can handle very large integers.

Run it, and once that works...
Add a method to calculate Fib **iteratively**. Let's call it: `long fibIterative( int n)`.
Call your new method in `main()` so that Fib is calculated and printed both recursively and iteratively for each N.

Quiz time!!!
- Quiz #1: What is Fib(50)?
- Quiz #2: What is Fib(60)?

Hint: Run your iterative method first. HA!

So, why is Fib recursive so slow? I think it's because it does a LOT of method calls.
Let's see how many:
- Add a class variable to count the number of times `fib()` is called (hint - your methods are static, so your class variable must be as well!)
- Set it to zero before calling `fib()`
- Increment it inside `fib()`, once for each call
- Report the tally after `fib()` returns in `main()`

The **iterative** solution to Fib(50) takes 1 method call and 50 loop iterations.
The recursive solution to Fib(50) takes how many method calls?!?!

What do you think?
How do you rate Fib recursive v. Fib iterative based on
1. Elegance of solution, and
2. Performance

thanks… yow, bill