#|

Second Racket Homework

Rodrigo Benavente García

01/09/21

|#

#lang racket

(define (get-pi-regular num);First exercise, recieves the terms to compute the value of pi

(if (zero? num) 0;If the number is 0 the recursion stops and we return the total, else we start calculating pi

(+ ;Add the result of every iteration

(/ (\* (expt -1 (+ num 1)) 4) (- (\* 2 num) 1.0)) (get-pi-regular (sub1 num)));We start calulating pi and we start a recursion to get a more accurate answer

);end of if

);End of my first exercise

(define (get-pi num);Second exercise, recieves the terms to compute the value of pi

(let loop;Start an anonymous function that will help with tail recursion

([i num] [total 0]);Create variables and assign them initial values

(if (zero? i);Check to see if the number is 0

total;If number is 0 we return total, else we calculate pi

(loop (sub1 i) (+ (/ (\* (expt -1 (+ i 1)) 4) (- (\* 2 i) 1.0)) total));We start calulating pi and we start a tail recursion to get a more accurate answer

);end of if

);End of anonymous function

);End of my second exercise

(define (ln-2 num);Third exercise, recieves a number to compute the value of the natural logarithm of 2

(let loop;Anonymous function that helps with tail recursion

([i num] [total 0]);Create variables and assign them initial values

(if (zero? i);Check to see if the number is 0

total;If number is 0 we return total, else we calculate ln of 2

(loop (sub1 i) (+ (/ 1 (\* (\* 2 i) (- (\* 2 i) 1.0))) total));We calculate ln2 and start tail recursion

);end of if

);End of anonymous function

);End of my third exercise

(define (russian-\*-regular num1 num2);Fourth exercise, recieves two numbers to multiply

(if (or (zero? num2) (zero? num1)) 0;Ends recursion and returns total

(if (even? num2);Checks if the number is even

(+ (russian-\*-regular (\* num1 2) (quotient num2 2)) 0);Add nothing to the total, multiply num1\*2, and start recursion

(+ (russian-\*-regular (\* num1 2) (quotient num2 2)) num1);Add "num1" to the total,, multiply num1\*2, and start recursion

);End of if that checks for even numbers

);End of if that checks if the quotient of num2 is 0

);End of my fourth exercise

(define (russian-\* num1 num2);Fourth exercise, recieves two numbers to multiply

(let loop;Anonymous function that helps with tail recursion

([a num1] [b num2] [total 0]);Create variables and assign them initial values

(if (or (zero? b) (zero? a)) total;Ends recursion and returns total

(if (even? b);Checks if the number is even

(loop (\* a 2) (quotient b 2) total);Add nothing to the total, multiply a\*2, and start recursion

(loop (\* a 2) (quotient b 2) (+ total a));Add "a" to the total, multiply a\*2, and start recursion

);End of if that checks for even numbers

);End of if that checks if the quotient of num2 is 0

);End of anonymous function

);End of my fourth exercise