

Lab23: Recursion

Part 1: Linear Function

-Create a recursive `linear()` function that takes `int x`, `int m`, `int b`, and returns $y = mx + b$ without using multiplication for $m \cdot x$. You may assume positive coefficients.

-Use the `linear()` function to make a table for the function $y = 2x + 3$ for x values from 0-10,

Ex,

x	y
0	3
1	5
2	7
3	9

...

--

Part 2: Fibonacci

-Create a `fib()` function that takes a positive `int n`, and returns the n th Fibonacci number.

-Use the `fib()` function to print out the first 25 Fibonacci numbers.

Ex,

n	nth Fib number
1	1
2	1
3	2
4	3

...

--

Part 3: Pascal's Triangle

-Create a `pascals()` function that takes a row and a column, and returns the number in that spot of the Pascal's Triangle.

-Use the `pascals()` function to print out the first 10 rows of the Pascal's Triangle. You may slant the triangle left to make printing easier.

Ex,

1					
1	1				
1	2	1			
1	3	3	1		
1	4	6	4	1	

...

-Challenge: Don't slant left, and make the triangle look nice. Don't use tabs. Allow for up to 3 digit numbers max.

Lab23: Recursion

Part 4: Quadratic Function

-Create a recursive quadratic() function that takes int x, int a, int b, int c, and returns $y=ax^2+bx+c$ without using any multiplications besides one "a*". You may assume positive coefficients.

-Hint: How do you go from 1 term to the next? How do you make f(3) in terms of f(2)?

-Use the quadratic() function to make a table for the function $y = 2x^2 + x + 1$ for x values from 0-10.

ex,

x	y
0	1
1	4
2	11

...

-Create this function again using the linear function, and no multiplication.

Part 5: Double Letters

-Create a recursive doubleLetters() function that takes a String s, and returns a String with each letter doubled. Ex, doubleLetters("Hello") => "HHeellllloo"

Part 6: Cap Vowel

-Create a recursive capVowel() function that takes a String s, and returns a String with each vowel capitalized. Ex, capVowel("Hello") => "HEllO"

Part 7: Reverse

-Create a recursive reverse() function that takes a String s, and returns it reversed. Ex, reverse("Hello") => "olleH"

-Use the reverse function to make another function called isPallindrome(), that checks to see if a String is the same as its reverse. isPallindrome() should return a boolean.

-Use the isPallindrome() function to print out all numbers between 1000 - 9999 that are palindromes.

Lab23: Recursion

Part 8: Create these recursive functions (They should all use recursion):

String beforeVowel(String s)- Returns all letters before the first vowel.

String afterVowel(String s)- Returns all letters after the first vowel.

String noLeadingWhiteSpace(String s)- Removes all spaces at the front of the String.

String beforeSpace(String s)- Returns all letters before the first space.

String afterSpace(String s)- Returns all letters after the first space.

String pigLatin(String s)- Returns the sentence in Pig Latin.

Ex,

beforeVowel("vowel") => "v"

afterVowel("vowel") => "owel"

beforeVowel("owl") => ""

afterVowel("owl") => "owl"

noLeadingWhiteSpace(" He llo") => "He llo"

beforeSpace("Hello Dave") => "Hello"

afterSpace("Hello Dave") => "Dave"

pigLatin("hello world") => "ellohay orldway"

--