Assignment 13.2 (Finding nth Digit)

Task1: Finding using standard for loop

Here input is n which is the nth digit. Initialize a string variable y with blank. In a for loop calculate the terms of Fibonacci seq and append to y. When nth digit Is reached, he digit is printed. The code is as below:

Code is as below:

def fibonacciSeq(n : Int) : String = {

var term1:BigInt = 0

var term2:BigInt = 1

var y = ""

while (y.length < n) {

y = y + term2.toString

var temp = term1

term1 = term2

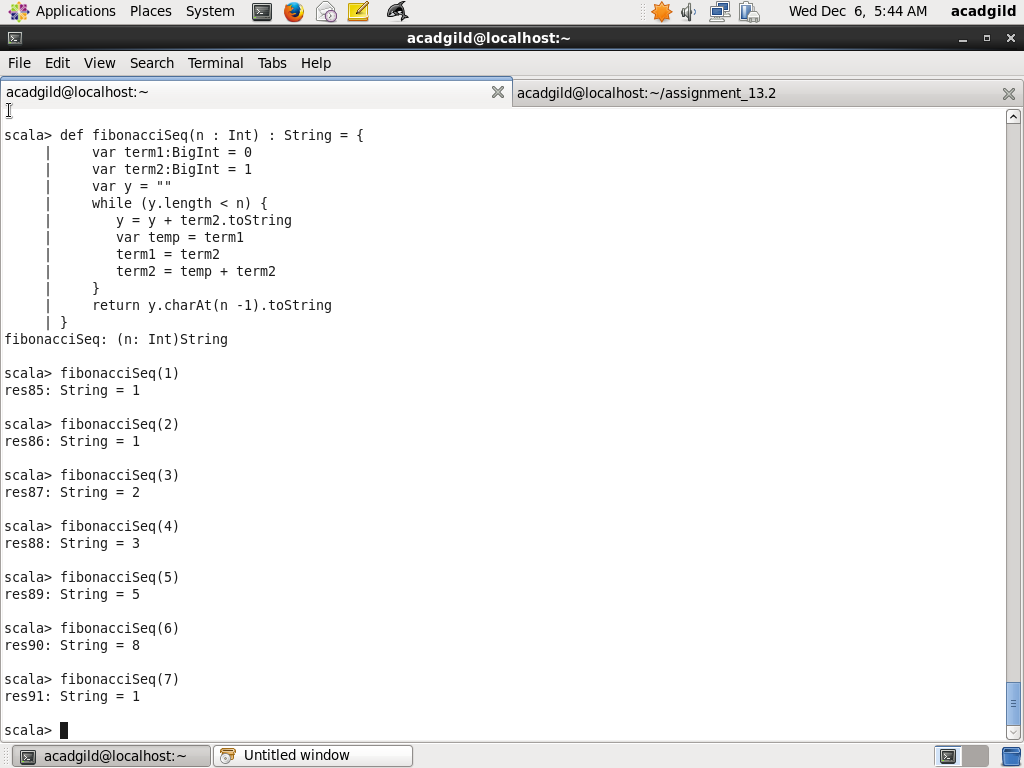
term2 = temp + term2

}

return y.charAt(n -1).toString

}

Screenshot is as below. Please note the 7th digit is 1 this is because 7th term is 13



Task1: Finding using recursion

Here I use the tail recusion. In the inner method of tail recursion, I use parameters n , origina\_n, term1 intialized to 0, term2 initialized to 1 and string x which is initialized to blank string. A variable y is used, which is concatenation of x and term2. Each recursive loop, term1 and term2 are modified. The terminating condirion is when nth digit is reached, that time, nth term in x is returned.

Scala code is as below:

import scala.annotation.tailrec

def fibonacciSeq(n : Int) : String = {

@tailrec

def fibonacci(n: Int, original\_n: Int, term1: BigInt =0, term2: BigInt = 1, x:String ="") : String = {

if (x.length >= original\_n) {

return x.charAt(original\_n -1).toString

} else {

var y = x + term2.toString

return fibonacci(n-1, original\_n, term2, (term1 + term2), y)

}

}

return fibonacci(n, n)

}

Screenshot is as below. Please note the 7th digit is 1 this is because 7th term is 13

