Algoritmos T.P. N° 5 – Recursividad - Juan Cruz Ambrosini

Ejercicio 1:

from algo1 import \*

def fibo (n):

    if n == 0:

        return 0

    if n == 1:

        return 1

    n1 = fibo(n-1)

    n2 = fibo(n-2)

    return (n1 + n2)

print(fibo(0))

print(fibo(1))

print(fibo(12))



Ejercicio 2:

def firstIntegers(n):

    sum = 0

    if n == 0:

        return 0

    if n == 1 :

        return 1

    return (n + firstIntegers(n-1))

print(firstIntegers(10))



Ejercicio 3:

def recursive\_even\_add(n):

    if n < 2:

        print("El valor ingresado tiene que ser mayor que 2.")

        return None

    elif(n == 2):

        return 2

    else:

        if n % 2 != 0 :

            print(n,"es impar.")

            return recursive\_even\_add(n-1)

        else:

            return n + recursive\_even\_add(n-2)

print("La suma de todos los enteros pares antes de 10 es: ", recursive\_even\_add(10))



Ejercicio 4:

from mylinkedlist import \*

def sortList(list : LinkedList, start = 0):

    length = lengthList(list)

    if start == length - 1 :

        return

    else :

        min\_index = start

        min\_value = access(list, start)

        for i in range (start + 1, length):

            if access(list, i) < min\_value :

                min\_value = access(list, i)

                min\_index = i

        if min\_index != start :

            move(list, min\_index, start)

        sortList(list, start + 1)

list = LinkedList()

add(list, 58)

add(list, 12)

add(list, 73)

add(list, 4)

add(list, 89)

add(list, 31)

add(list, 66)

add(list, 25)

showList(list)

sortList(list,0)

showList(list)



Ejercicio 5:

from mystack import \*

def stack\_fibo(n):

    stack = LinkedList()

    push(stack, 0)

    if n == 0 :

        return 0

    push(stack, 1)

    if n == 1 :

        return 1

    for i in range (2, n + 1):

        push(stack, (stack.head.value + stack.head.nextNode.value))

    return pop(stack)

