

```
def fibo(n):  
    if n < 0:  
        return  
    elif n == 0:  
        return 0  
    elif n == 1:  
        return 1  
    else:  
        return fibo(n-2) + fibo(n-1)  
  
n = 9  
  
print ("Calculando Fibonacci para n =", n, "->", fibo(n))
```

```
def sumaNEnteros(n):  
    if n<0:  
        return  
    elif n==0:  
        return 0  
    else:  
        return n + sumaNEnteros(n-1)  
  
n=4  
  
print("La suma de los primeros", n, "enteros es", sumaNEnteros(n))
```

```
def sumaNEnterosPares(n):  
    if (n < 0) or ((n % 2) == 1):  
        print("ERROR!")  
        return  
    elif n==2:  
        return 2  
    else:  
        return n + sumaNEnterosPares(n-2)  
  
n=8  
  
print("La suma de los enteros PARES desde", n, "hasta 2 es", sumaNEnterosPares(n))
```

```
from linkedlist import *

# función recursiva que añade los elementos de LDes a LOrd ordenándolos de menor a mayor:
def anhademenoramayor(LDes, LOrd):
    if length(LDes) == 0:
        return
    else:
        add(LOrd, encuentraMayor(LDes))
        anhademenoramayor(LDes, LOrd)

# dada una lista de números enteros devuelve el MAYOR de ellos y lo borra de la lista
def encuentraMayor(L):
    nodo = L.head
    # suponemos que el primer valor de la lista podría ser el mayor de toda la lista
    mayor = nodo.value
    # iteramos el resto de la lista buscando el mayor valor
    while nodo.nextNode != None:
        nodo = nodo.nextNode
        if nodo.value > mayor:
            mayor = nodo.value
    # borramos el nodo que vamos a devolver:
    delete(L, mayor)
    return mayor

# lista desordenada para hacer pruebas:
LDes = LinkedList()

add(LDes, 678)
add(LDes, 3)
add(LDes, 22)
add(LDes, 54)
add(LDes, 345)
add(LDes, 89)
add(LDes, 67)
add(LDes, 7)
add(LDes, 45)
add(LDes, 24)

# creamos lista vacía para meter en ella los elementos ordenados de menor a mayor
LOrd = LinkedList()

anhademenoramayor(LDes, LOrd)

# mostramos la lista ya ordenada
nodo = LOrd.head
for i in range(length(LOrd)):
    print(nodo.value, " ", end='', sep='')
    nodo = nodo.nextNode
print("")
```

```
from mystack import *

def fiboPila(n):
    S=LinkedList()
    # metemos los dos primeros valores a mano:
    push(S,1)
    push(S,0)
    for i in range(n):
        mayor = pop(S)
        menor = pop(S)
        suma = menor + mayor
        push(S,mayor)
        push(S,suma)
    return suma

n = 8

print ("Calculando Fibonacci para n =", n, "->", fiboPila(n))
```

```
correo@bunta:~/UNC/AyED1/TP06/recursividad% python ejercicio1.py
Calculando Fibonacci para n = 9 -> 34
correo@bunta:~/UNC/AyED1/TP06/recursividad% python ejercicio2.py
La suma de los primeros 4 enteros es 10
correo@bunta:~/UNC/AyED1/TP06/recursividad% python ejercicio3.py
La suma de los enteros PARES desde 8 hasta 2 es 20
correo@bunta:~/UNC/AyED1/TP06/recursividad% python ejercicio4.py
3 7 22 24 45 54 67 89 345 678
correo@bunta:~/UNC/AyED1/TP06/recursividad% python ejercicio5.py
Calculando Fibonacci para n = 8 -> 21
correo@bunta:~/UNC/AyED1/TP06/recursividad% █
```