Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores

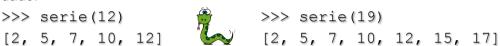
Introducción a la Programación / Taller de programación.

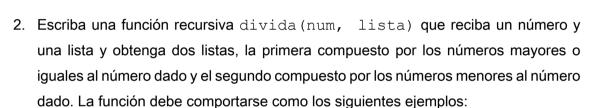
Profesor: Jeff Schmidt Peralta II Semestre 2019

EJERCICIOS RECURSIVIDAD DE COLA 5b.

1. Analice la siguiente serie numérica:

Determine su fórmula y construya una función recursiva en cola llamada serie (n) que reciba como argumento un número y genere la serie desde 2 hasta el número dado.





3. Escriba una función en cola parejas (num) en la cual num es un valor entre 2 y 100 que encuentre todas las parejas (sin importar el orden) de números que suman el número dado. Ejemplos:

```
>>> parejas(4)
[[0, 4], [1, 3], [2, 2]]
```

4. En una competencia deportiva 10 jueces califican a los competidores en una escala de 1 a 100. Para obtener la calificación de un competidor, se eliminan la más alta y la más baja y se hace un promedio de las calificaciones restantes. Debe implementar la función usando recursividad de cola calificacion (lista) que recibe una lista con las calificaciones de los 10 jueces y que obtenga la calificación de un competidor.

```
>>> calificacion([6, 8, 8, 8, 8, 8, 8, 10, 8, 8])
```



TEC Tecnológico de Costa Rica

5. Hacer una función llamada invertir (lista), que reciba una lista e invierta el orden de sus elementos:

```
>>> invertir([0, 1, 2, 3, 4])
[4, 3, 2, 1, 0]
```

6. Escriba una función recursiva elimine (lista, num) que reciba una lista y un número y elimine de la lista los números que coincidan con el número dado, a excepción de la última aparición. No puede invertir la lista de entrada.

```
>>> elimine([4,8,2,4,0,1], 4) >>> elimine([5,2,3], 6)
[8,2,4,0,1] [5,2,3]
```

7. Escriba una función a llamar eliminar_todos (lista, ele) que reciba una lista y un elemento y elimine todas las apariciones de ese elemento en la lista.

```
>>> eliminar_todos([0, 5, 1, 2, 5, 3, 4], 5)
[0, 1, 2, 3, 4]
>>> eliminar_todos([0, 5, 1, 2, 5, 3, 4], 8)
[0, 5, 1, 2, 5, 3, 4]
```

8. Escriba una función intercambia (listal, listal, posición) que reciba como argumentos dos listas del mismo tamaño y una posición entre 0 y el largo de las listas – 1 e intercambie las posiciones desde la 0 hasta la posición dada en ambas listas.

```
>>> intercambia(['a', 'b', 'c', 'd'], [1, 2, 3, 4], 1) [['a', 'b', 3, 4], [1 2, 'c', 'd']]
```

9. Escriba una función ultimo_par(lista) utilizando recursividad de cola, que recibe una lista no nula y retorna el último número par de la lista dada. No puede invertir la lista. Asuma que la lista está compuesta solo por números.

```
>>> ultimo_par([23, 72, 149, 34])
34
>>> ultimo_par([12, 5, 21, 3])
12
>>> ultimo_par([1, 3, 5, 7])
No hay par
```

10. Escriba una función moda (lista) que reciba una lista de números y obtenga el número que más veces se repite.

```
>>> moda([9,1,7,8])
  -1 # -1 porque no se repite ninguno
>>> moda([1,0,0,0,0,9])
0
```

11. Escriba una función cuente_todos (ele, lista) que reciba un elemento y una lista y cuente todas las apariciones del elemento en la lista, aún si el elemento se encuentra en una sublista (a cualquier nivel).

```
>>> cuente_todos(4, [4, 5, [[[3, 4], 5], [3]], 4, 8])
3
```