

Control 1 Paradigmas de Programación (2022-1)

Abril

Duración: 45 minutos

NOMBRE: _____

RUT: _____

Profesor: _____

Pregunta I (Procure responder todos los ítems de esta pregunta en esta misma hoja (puede usar ambos lados))

Dada la siguiente función que permite calcular la sumatoria desde i hasta n

```
(define sumatoria (lambda (i n)
  (if (= i n)
      i
      (+ i (sumatoria (+ i 1) n)))))
```

a) ¿Qué tipo de recursión se aprecia en el código? ____Natural____

b) Señale en el código anterior aquellos elementos distintivos que lo llevaron a la clasificación del tipo de recursividad que señaló en a)?

(+ i (sumatoria (+ i 1) n)))))

Estado pendiente

c) Respetando el encabezado de la función, esto es, su nombre, los argumentos de entrada y el orden de estos; implementar una alternativa usando otro tipo de recursividad distinta a la señalada en a.1)

;Wrapper / envoltorio

(define sumatoria (lambda (i n)

(define sumatoriaCola (lambda (i n resultado) ;auxiliares , helper

(if (= i n)

(+ i resultado)

(sumatoriaCola (+ i 1) n (+ i resultado))))

if (= n 0)

0

(sumatoriaCola i n 1)))))

d) Documente su implementación indicando Dominio, Recorrido, Descripción y Tipo de recursividad empleada.

;Descripción: Calcula la sumatoria desde i hasta n

;DOM: Enteros positivos (\mathbb{Z}^+) X Enteros positivos (\mathbb{Z}^+)

;REC: \mathbb{Z}^+

;Tipo Recursión: Cola

```
(define sumatoria (lambda (i n)
```

```
  (define sumatoriaCola (lambda (i n resultado)
```

```
    (if (= i n)
```

```
      (+ i resultado)
```

```
      (sumatoriaCola (+ i 1) n (+ i resultado))))
```

```
  if (= n 0)
```

```
    0
```

```
    (sumatoriaCola i n 1))))
```

Control 1 Paradigmas de Programación (2022-1)

Abril

Duración: 45 minutos

NOMBRE: _____

RUT: _____

Profesor: _____

Pregunta II (Procure responder todos los ítems de esta pregunta en esta misma hoja (puede usar ambos lados))

- a) Implementar una función que permita cambiar los elementos de una lista que cumplan con un determinado criterio. La función se llamará `updateElements` y tendrá el siguiente encabezado (`updateElements Lista Filtro Value`). A modo de ejemplo:

`(updateElements (list 1 2 3 4 5 6) odd? -1)) => (-1 2 -1 4 -1 6)`

Reemplaza todos los valores impares de la lista por -1.

;Desc: Actualiza elementos de una lista por un nuevo valor a partir de un criterio de entrada

;DOM: Lista X Función Booleana X Value

;REC: Lista

;Rec: Natural

`(define updateElements (lambda (L f v)`

`(if (null? L)`

`null`

`(if (f (car L))`

`(cons v (updateElements (cdr L) f v)) ;esto es cuando cond es true`

`(cons (car L) (updateElements (cdr L) f v))) ;cuando cond es false`

o bien podría ser

`(define updateElements (lambda (L f v)`

`(map (lambda (e) (if (f e) v e) L))`

y `(updateElements (list 1 2 3 4 5 6) even? -1)) => (1 -1 3 -1 5 -1)`

donde se reemplazan todos los valores pares de la lista por un -1.

- b) Documente su implementación indicando Dominio, Recorrido, Descripción y Tipo de recursividad empleada.
- c) Muestre cómo se usaría la función implementada en (a) para sustituir todos los elementos numéricos que son múltiplos de 3. Procure expresar el criterio a través de una función anónima que se pasa como parámetro a `updateElements`. Para este

caso puede usar la función “remainder” que permite obtener el resto de la división entera.

```
(updateElements (list 1 2 3 4 5 6 7 8 9) (lambda (n) (= (remainder n 3) 0)) -1)
```