

Práctica de Racket

Tema: Programación Racket

1) Escriba funciones recursivas (10 cada ejercicio). Utilice los nombres de las funciones como se presentan en el ejemplo para facilitar la revisión de los ejercicios.

a) Escriba una función recursiva que reciba como parámetro un número n y calcule la suma de todos los enteros comprendidos entre [1,n].

```
>(sum-rank 4)  
10
```

```
(sum-rank 6)  
21
```

```
> (sum-rank 10)  
55
```

b) Escriba una función recursiva que reciba una lista y elimine los duplicados continuos
Ejemplo:

```
> (compress (list 1 2 2 3 4 5 5 5 5))  
'(1 2 3 4 5)
```

c) Escriba una función recursiva que reciba una lista de listas y devuelva una lista plana (ie. sin listas internas)

Ejemplo:

```
> (flatten (list 1 2 3 (list 4 5 (list 6 7 8 (list 9 10 11)) )))  
'(1 2 3 4 5 6 7 8 9 10 11)
```

d) Escriba una función recursiva que duplique cada elemento de una lista

Ejemplo

```
(duplicate '(A B C C D))
```

```
(A A B B C C C C D D)
```

e) Escriba una función recursiva que divida una lista en dos partes dada la longitud de la primera parte.

Ejemplo:

```
(split '(A B C D E F G H I K) 3)
```

```
Retorna ( (A B C) (D E F G H I K))
```

f) Escriba una función recursiva que empaquete duplicados consecutivos de elementos en sub-listas. Es decir, si una lista contiene elementos consecutivos repetidos, deben colocarse en sub-listas separadas.

Ejemplo:

(pack '(A A A B C C A A D E E E E))

Devuelve ((A A A A) (B) (C C) (A A) (D) (E E E E))