

Guía de Trabajos Prácticos VII – Programación Funcional 4

1. Definir una estructura que represente un punto en el plano. Crear una función que calcule la distancia entre dos puntos datos recibiendo como parámetros la estructura de cada uno.
2. Definir una estructura que represente un punto en el espacio. Crear una función que calcule el centro de gravedad de una lista de puntos en el espacio representados por la estructura definida previamente.
3. Calcule la distancia máxima entre dos puntos de una lista, usando el predicado MAP y estructuras.
4. Cuente la cantidad de palabras y de caracteres que contiene un archivo TXT.
5. Cree una función que permita mostrar por pantalla un archivo de texto.
6. Ordenar por Peso ASCII

Se desea crear una función que reciba como parámetro una lista de strings y devuelva una lista con las cadenas ordenadas por su peso ASCII. Ej:

```
> (ordenar '("moto" "auto" "casa" "juego" "aire")') --> ("casa" "aire" "auto" "moto" "juego")
```

El peso ascii de una palabra se calculará como la suma de los valores de cada uno de los caracteres que la componen. Ejemplo:

```
(pesopalabra "casa") --> 408
a = 97
c = 99
s = 115 ==> 99 + 97 + 115 + 97 = 408
```

7. Utilizando la función MAP cree una función llamada “fullreverse-list” que permite revertir completamente el contenido de una lista.

```
(fullreverse-list (1 (2 3 4 (4 5) (3 (5 6)) 4)))
--> (4 ((6 5) 3) (5 4) 4 3 2) 1)
```
8. Cree una función “app2list” que una dos elementos y los devuelve siempre como lista. Si los elementos son listas las junta, si es un elemento y una lista lo agrega y si son dos elementos crea una lista con ellos.
9. En una calle con adoquines unos niños juegan a la rayuela. Para esto numeran los adoquines de la siguiente forma:

| 0 | 1 | 2 | 3 | .. | n |

Los movimientos permitidos del juego son:

- Al principio del juego los niños se ubican en el adoquín 0.
- De un adoquín numerado i se puede saltar al adoquín numerado $i+1$.
- De un adoquín numerado i se puede saltar al adoquín numerado $i+2$ (sin pasar por el adoquín $i+1$)

Por ejemplo: (CaminosPosibles 3) $\rightarrow ((0\ 1\ 2\ 3)\ (0\ 2\ 3)\ (0\ 1\ 3))$

Escriban una función recursiva llamada CaminosPosibles que calcule el número de caminos posibles para alcanzar un adoquín objetivo numerado con n (mayor que cero). Utilice MAP para facilitar el desarrollo.