v.8.1

Programación 1 - Práctica 5, parte 2.

1 Clasificando elementos de una lista

Ejercicio 1. Diseñe una función `pares` que tome una lista de números `l` y devuelva una lista con los números pares de `l`.

Ejemplo:

```
pares (list 4 6 3 7 5 0)
= (list 4 6 0)
```

Ejercicio 2. Diseñe una función `cortas` que tome una lista de strings y devuelva una lista con aquellas palabras de longitud menor a 5.

Ejemplo:

```
cortas (list "Lista" "de" "palabras" "sin" "sentido")
= (list "de" "sin")
```

Ejercicio 3. Diseñe una función `cerca` que tome una lista de puntos del plano (representados mediante estructuras `posn`), y devuelva la lista de aquellos puntos que están a distancia menor a `MAX`, donde `MAX` es una constante de su programa.

Ejemplo (considerando `5` para la constante) :

```
cerca (list (make-posn 3 5) (make-posn 1 2) (make-posn 0 1) (make-posn 5 6))
= (list (make-posn 1 2) (make-posn 0 1))
```

Ejercicio 4. Diseñe una función `positivos` que tome una lista de números y se quede sólo con aquellos que son mayores a 0.

```
(positivos (list -5 37 -23 0 12))
= (list 37 12)
```

Ejercicio 5. Diseñe una función `eliminar` que tome una lista de números y un número y devuelva la lista luego de eliminar el número que indica el 2do argumento

Ejemplo:

```
(eliminar (list 1 2 3 2 7 6) 2)
= (list 1 3 7 6)

(eliminar (list 1 2 3 2 7 6) 0)
= (list 1 2 3 2 7 6)
```

---

2 Aplicando transformaciones a cada elemento de una lista

Ejercicio 6. Diseñe la función `raices`, que dada una lista de números, devuelve una lista con las raíces cuadradas de sus elementos.

```
(raices (list 9 16 4))
= (list 3 4 2)
```

Ejercicio 7. Diseñe una función `distancias` que tome una lista de puntos del plano y devuelva una lista con la distancia al origen de cada uno.

Ejemplo:

```
(distancias (list (make-posn 3 4) (make-posn 0 4) (make-posn 12 5)))
= (list 5 4 13)
```

Ejercicio 8. Diseñe una función `anchos` que tome una lista de imágenes y devuelva una lista con el ancho de cada una.

Ejemplo:

```
(anchos (list (circle 30 "solid" "red") (rectangle 10 30 "outline" "blue")))
= (list 60 10)
```

Ejercicio 9. Diseñe la función `signos`, que dada una lista de números, devuelve una lista con el resultado de aplicarle a cada elemento la función `sgn2` definida en la práctica 1.

```
(signos (list 45 32 -23 0 12))
= (list 1 1 -1 0 1)
```

Ejercicio 10. Diseñe una función `cuadrados` que tome una lista de números y devuelva otra lista donde los elementos que aparezcan sean el cuadrado de los elementos de la lista original.

Ejemplo:

```
(cuadrados (list 1 2 3))
= (list 1 4 9)
```

Ejercicio 11. Diseñe una función `longitudes` que tome una lista de cadenas y devuelva una lista de números que corresponda con la longitud de cada cadena de la lista original.

Ejemplo:

```
(longitudes (list "hola" "cómo" "estás?"))
= (list 4 4 6)
```

Ejercicio 12. Diseñe la función `convertirFC`, que convierte una lista de temperaturas medidas en Fahrenheit a una lista de temperaturas medidas en Celsius.

---

3 Operando con los elementos de una lista

Ejercicio 13. Diseñe una función `prod` que multiplica los elementos de una lista de números. Para la lista vacía, devuelve `1`.

```
(prod (list 1 2 3 4 5))
= 120
```

Ejercicio 14. Diseñe una función `pegar` que dada una lista de strings, devuelve el string que se obtiene de concatenar todos los elementos de la lista.

```
(pegar (list "Las " "lis" "tas " "son " "complicadas" "."))
= "Las listas son complicadas."
```

Ejercicio 15. Diseñe una función `max` que devuelve en máximo de una lista de naturales. Para la lista vacía, devuelve `0`.

```
(max (list 23 543 325 0 75))
= 543
```

Ejercicio 16. Diseñe una función `sumdist` que tome una lista de puntos del plano y devuelva la suma de sus distancias al origen.

Ejemplo:

```
(sumdist (list (make-posn 3 4) (make-posn 0 4) (make-posn 12 5)))
= 22
```

Ejercicio 17. Diseñe una función `sumcuad` que dada una lista de números devuelve la suma de sus cuadrados. Para la lista vacía,

devuelve `0`.

```
(sumcuad (list 1 2 3))
= 14
```