

# Práctica 9

## Introducción a listas

Introducción a la Programación  
2<sup>do</sup> Semestre de 2017

Los ejercicios que corresponden a los **contenidos mínimos** recomendados se encuentran marcados con el símbolo  $\otimes$ .

### 1. Manejo básico de listas con el tablero

#### Ejercicio 1

Escribir la función `esSingular` que, dada una lista, retorne un booleano indicando si la lista tiene un único elemento.

#### Ejercicio 2

Escribir la función `direccionesAlBorde` que retorne la lista de direcciones en las que el cabezal no se puede mover. Por ejemplo, si el cabezal se encuentra en el origen, la función retorna `[Sur, Oeste]`.

#### Ejercicio 3

Escribir la función `coloresCelda` que retorne la lista de los colores que aparecen en la celda actual.

#### Ejercicio 4

Escribir el procedimiento `PonerLista` que, dada una lista de colores  $[c_1, c_2, c_3, \dots, c_k]$ , ponga en la celda actual una bolita de color  $c_i$  por cada  $i = 1, 2, 3, \dots, k$ .

Ejemplo: `PonerColores([minColor(), Verde, maxColor()])` pone dos bolitas Verdes y una Azul.

#### Ejercicio 5

Escribir el procedimiento `RecorrerCamino` que, dada una lista de direcciones  $[d_1, d_2, \dots, d_k]$ , mueva el cabezal primero en dirección  $d_1$ , después en dirección  $d_2$ , y así sucesivamente hasta mover el cabezal en dirección  $d_k$ .

Ejemplo: dada la lista `[Este, Este, Norte, Este]`, el procedimiento debe mover el cabezal primero hacia el Este dos veces, luego hacia el Norte y por último hacia el Este. ¿Cuál es la precondición de este procedimiento?

#### Ejercicio 6

Escribir la función `aparicionesDeColor` que, dado un color  $c$ , recorra las celdas del tablero —en cualquier orden— y retorne la cantidad total de bolitas de color  $c$  en las celdas recorridas.

**Ejercicio 7**

Escribir la función `aparicionesDeColorEnCamino` que, dado un color  $c$  y una lista de direcciones  $ds$ , retorne la cantidad total de bolitas de color  $c$  en las celdas recorridas, si las mismas se recorren en el orden en que indica  $ds$ , como se realiza en el Ejercicio 5 (incluyendo la celda en que se encuentra el cabezal antes de empezar y al finalizar el recorrido). ¿Cuál es la precondition del procedimiento?

**Ejercicio 8**

Escribir el procedimiento `PonerListaN` que, dada una lista de colores  $[c_1, c_2, c_3, \dots, c_k]$  y una lista de números  $[n_1, n_2, n_3, \dots, n_k]$ , ponga en la celda actual  $n_i$  bolitas de color  $c_i$  para cada  $i = 1, 2, 3, \dots, k$ . ¿Cuál es la precondition del procedimiento?

## 2. Manejo básico de listas sin uso del tablero

Los ejercicios de esta sección no hacen uso del tablero de XGOBSTONES.

**Ejercicio 9**

Escribir la función `repeticion` que, dado un número  $n$  y un valor  $x$  (de cualquier tipo), devuelva una lista con  $n$  repeticiones de  $x$ . Si  $n \leq 0$ , retorna la lista vacía.

Ejemplo: `repeticion(3, 8)` retorna la lista `[8, 8, 8]`

`repeticion(4, minBool())` retorna la lista `[False, False, False, False]`.

**Ejercicio 10**

Escribir la función `sumatoria` que, dada una lista de números, retorne la suma de todos los números de la lista.

Ejemplo: `sumatoria([3, 2, 1, 4])` retorna 10.

**Ejercicio 11**

Escribir la función `reverso` que, dada una lista  $[e_1, \dots, e_k]$ , retorne la lista  $[e_k, \dots, e_1]$ .

Ejemplo: `reverso([True, True, False, True])` retorna `[True, False, True, True]`.

**Ejercicio 12**

Escribir la función `longitud` que, dada una lista, retorne la cantidad de elementos de la misma.

Ejemplo: `longitud([Azul, Azul, Verde, Rojo])` retorna 4.

**Ejercicio 13**

Escribir la función `opuestos` que, dada una lista  $[e_1, \dots, e_k]$ , retorne la lista de opuestos  $[\text{opuesto}(e_1), \dots, \text{opuesto}(e_k)]$ .

Ejemplo: `opuestos([Oeste, Sur, Norte])` retorna `[Este, Norte, Sur]`.

**Ejercicio 14**

Escribir la función `multiplicacionLista` que, dada una lista  $ls$  y un número  $m$ , retorne una lista que contenga  $ls$  repetida  $m$  veces.

Ejemplo: `multiplicacionLista([1, 2, 3], 3)` retorna `[1, 2, 3, 1, 2, 3, 1, 2, 3]`.

**Ejercicio 15**

Escribir la función `multiplicacionInterna` que, dada una lista  $[e_1, \dots, e_k]$  y un número  $m$ , retorne la lista en la que cada  $e_i$  aparece  $m$  veces repetido. Sugerencia: reutilizar la función `repeticion` junto con el operador `++`

Ejemplo: `multiplicacionInterna([1, 2, 3], 3)` retorna `[1, 1, 1, 2, 2, 2, 3, 3, 3]`.

**Ejercicio 16**

Escribir la función `sumatoriaVecinos` que, dada una lista  $ns = [n_1, \dots, n_k]$ , retorne la lista  $ms = [m_2, \dots, m_{k-1}]$  tal que  $m_i$  es la suma de  $n_{i-1}$ ,  $n_i$  y  $n_{i+1}$ .

Ejemplo: `sumatoriaVecinos([1, 2, 3, 4])` retorna `[6, 9]` (i.e. `[1+2+3, 2+3+4]`). Notar que  $ms$  tiene dos elementos menos que  $ns$ .

**Ejercicio 17**

Escribir la función `sinElemento` que, dada una lista  $ls$  y un elemento  $e$ , retorne la lista que resulta de quitar todas las apariciones de  $e$  que ocurren en  $ls$ .

Ejemplo: `sinElemento([Azul, Verde, Azul, Rojo], Azul)` retorna `[Verde, Rojo]`.

**Ejercicio 18**

Escribir la función `pares` que, dada una lista de números  $ns$ , retorne la lista de números pares que aparecen en  $ns$ .

Ejemplo: `pares([3, 4, 5, 2, 5])` retorna `[4, 2]`.

**Ejercicio 19**

Un elemento de una lista se dice *interno* si es igual al anterior de la lista. Escribir la función `reduccionBloques` que, dada una lista, retorne la lista que se obtiene de quitar todos los elementos internos.

Ejemplo: `reduccionBloques([1, 1, 2, 2, 2, 2, 3, 1, 2, 2])` retorna `[1, 2, 3, 1, 2]`.

**Ejercicio 20**

Escribir la función `zip` que, dadas dos listas de  $[e_1, \dots, e_k]$  y  $[f_1, \dots, f_k]$  de igual longitud, retorne la lista  $[e_1, f_1, \dots, e_k, f_k]$ .

Ejemplo: `zip([1, 3, 5], [2, 4, 6])` retorna `[1, 2, 3, 4, 5, 6]`.

**Ejercicio 21**

Modificar la función del ejercicio anterior para que funcione cuando se invoca con listas de distinta longitud como argumento.

**Ejercicio 22**

Escribir la función `unzipImpares` que, dada una lista, devuelva la lista que se obtiene de quitar los elementos en las posiciones pares.

Ejemplo: `unzipImpares([1, 2, 3, 4, 5, 6])` retorna `[1, 3, 5]`.

### 3. Ejercicios adicionales

#### Ejercicio 23

Escribir la función `pertenece` que, dado un elemento  $e$  y una lista  $xs$ , indique si el elemento pertenece a la lista.

Ejemplo: `pertenece(3, [2,3,4])` retorna `True`.

`pertenece(5, [2,3,4])` retorna `False`.

#### Ejercicio 24

Escribir la función `sinDuplicados` que, dada una lista  $ls$ , retorne una lista sin elementos repetidos que tenga todos los elementos de  $ls$ .

Ejemplo: `sinDuplicados([1, 3, 4, 2, 4, 3, 5])` retorna (algún reordenamiento de) `[1, 2, 3, 4, 5]`.

#### Ejercicio 25

Escribir la función `union` que, dada dos listas  $ls$  y  $ms$  que no contienen elementos repetidos, devuelva una lista sin repetidos que contenga todos los elementos que aparecen en alguno de  $ls$  o  $ms$ .

Ejemplo: `union([1, 3, 4], [2, 4, 3, 5])` retorna (algún reordenamiento de) `[1, 2, 3, 4, 5]`.

#### Ejercicio 26

Escribir la función `interseccion` que, dada dos listas  $ls$  y  $ms$  que no contienen elementos repetidos, devuelva la lista de todos los elementos que aparecen tanto en  $ls$  como en  $ms$ .

Ejemplo: `interseccion([1, 3, 4], [2, 4, 3, 5])` retorna (algún reordenamiento de) `[3, 4]`.

#### Ejercicio 27

Escribir la función `incluye` que dada dos listas  $xs$  y  $ys$  que no contienen elementos repetidos, indique si la segunda se encuentra contenida en la primera.

Ejemplos: `incluye([2,3,4,5,6], [4,5])` retorna `True`.

`incluye([2,3,4,5,6], [4,5,8])` retorna `False`.

`incluye([2,3,4,6,5], [4,5])` retorna `True`.

#### Ejercicio 28

Escribir la función `ordenada` que, dada una lista, retorne `true` si la lista esta ordenada de menor a mayor y `false` en caso contrario.

Ejemplos: `ordenada([2, 3, 4, 5])` retorna `true`

`ordenada([2, 4, 3, 5])` retorna `false`.

#### Ejercicio 29

Escribir la función `minimo` que, dada una lista, retorne el mínimo elemento. ¿Cuál es la precondición del procedimiento?

Ejemplo: `minimo([2, 1, 0, 3])` retorna 0.

### Ejercicio 30

Escribir la función `sinMinimo` que, dada una lista, retorne la lista que se obtiene de eliminar la primer aparición de el mínimo. ¿Cuál es la precondition del procedimiento?

Ejemplo: `sacarMinimo([2, 1, 0, 0, 3])` retorna `[2, 1, 0, 3]`

Sugerencia: escribir una función `sinPrimerAparicion` que, dada una lista y un elemento, retorne la lista sin la primer aparicion del elemento.

### Ejercicio 31

Escribir la función `ordenacion` que, dada una lista, retorne la lista ordenada de menor a mayor. **Ayuda:** pensar cómo se pueden combinar los dos ejercicios anteriores.

Ejemplo: `ordenacion([2, 1, 0, 3])` retorna `[0,1,2,3]`.

### Ejercicio 32

Escribir la función `agrupar` que, dada una lista `xs` y un número `n`, retorne una lista de listas que agrupe los elementos de `xs` en grupos de `n` elementos.

Ejemplo: `agrupar([1,2,3,4,5,6,7,8], 3)` retorna `[[1,2,3], [4,5,6], [7,8]]`.

### Ejercicio 33

Escribir la función `desagrupar` que, dada una lista de listas `xs`, retorne una lista producto de concatenar todas las listas internas de `xs`.

Ejemplo: `desagrupar([[1,2,3], [4,5,6], [7,8,9]])` retorna `[1,2,3,4,5,6,7,8,9]`.

`desagrupar(agrupar([2,3,4,5,6],2))` retorna `[2,3,4,5,6]`.