

Práctica 6

Introducción a listas

Introducción a la Programación
1^{er} Semestre de 2014

1. XGobstones

Ejercicio 1

Adapte de Gobstones a XGobstones las funciones y procedimientos: `MoverN`, `PonerN`, `SacarN`, `IrAlInicioT`, `MoverT`, `puedeMoverT` y `puedeMoverN`.

2. Manejo básico de listas con el tablero

Ejercicio 2

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

Ejercicio 3

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 4

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

Ejercicio 5

Escribir el procedimiento `PonerLista` que, dada una lista de colores $[c_1, \dots, c_k]$, ponga en la celda actual una bolita de color c_i por cada $i = 1, \dots, k$. Por ejemplo, `PonerColores([minColor(), Verde, maxColor()])` pone dos bolitas Verdes y una Azul.

Ejercicio 6

Escribir el procedimiento `RecorrerCamino` que, dada una lista de direcciones $[d_1, \dots, d_k]$, mueva el cabezal en 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 precondition de este procedimiento?

Ejercicio 7

Escribir la función `aparicionesDeColor` que, dado un color c , recorra las celdas del tablero —en cualquier orden— y retorne una lista con el número de bolitas de color c que hay en cada celda recorrida.

Ejercicio 8

Escribir la función `aparicionesDeColorEnCamino` que, dado un color c y una lista de direcciones ds , retorne la lista de números de bolitas de color c que hay en cada celda, si las mismas se recorren en el orden en que indica ds , como se realiza en el Ejercicio 6 (incluyendo la celda en que se encuentra el cabezal antes de empezar y al finalizar el recorrido). ¿Cuál es la precondition del procedimiento?

Ejercicio 9

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

3. Manejo básico de listas sin uso del tablero

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

Ejercicio 10

Escribir la función `repeticion` que, dado un número n y un valor x (de cualquier tipo), devuelva una lista con n repeticiones de x . Sin $n \leq 0$, 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 11

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 12

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

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

Ejercicio 13

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 14

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 15

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

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

Ejercicio 16

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

Ejercicio 17

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: `sumarVecinos([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 18

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 19

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

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

Ejercicio 20

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 21

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 22

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

Ejercicio 23

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]`.

4. Ejercicios adicionales

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]`.

Sugerencia: escribir una función `pertenece` que indique si un elemento pertenece a una lista.

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, 5]`.

Ejercicio 27

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 28

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

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

Ejercicio 29

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 30

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]`.