Introducción a la Programación - Práctica 10

Algoritmos sobre listas

CONSEJOS:

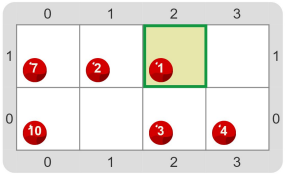
- Leer el enunciado en su totalidad y pensar en la forma de resolver el ejercicio ANTES de empezar a construir código.
- Si un ejercicio no sale, se puede dejar para después y continuar con los ejercicios que siguen.
- Los ejercicios están pensados para ser hechos después de haber mirado la teórica correspondiente.
- Algunos ejercicios están tomados de la guía complementaria realizada por Federico Aloi y Miguel Miloro, a su vez basada en las guías Ejercicios de Introducción a la Programación del CIU General Belgrano, elaboradas por Carlos Lombardi y Alfredo Sanzo, y Fundamentos de la Programación del Proyecto Mumuki. Agradecemos a todos los que nos ayudaron con su inspiración.

EJERCICIOS:

1. Construir el procedimiento **`PonerColores_EnLaCeldaActual`**, que dada una lista de Colores, pone una bolita del color correspondiente por cada uno de los elementos de la misma. Por ejemplo, **`PonerColores_EnLaCeldaActual([Verde, Verde, Azul])`** pone dos bolitas Verdes y una Azul en la celda actual.

2. Construir el procedimiento **`RecorrerCamino_`**, que dada una lista de Direcciones mueve el cabezal en la dirección indicada por cada elemento, en orden. Por ejemplo, **`RecorrerCamino_([Este, Este, Norte, Este])`** debe mover el cabezal primero hacia el Este dos veces, luego hacia el Norte 1 vez y por último se mueve hacia el Este. ¿Cuál es la precondición de este procedimiento?

3. Construir la función **`aparicionesDeColor_`**, que dado un color, **`colorBuscado`**, describa una lista de números que indican para cada una de las celdas del tablero recorridas en dirección principal Norte y dirección secundaria Oeste, la cantidad de bolitas del color buscado en esa celda del tablero. Por ejemplo, para el siguiente tablero, **`aparicionesDeColor_(Rojo)`** debería describir la lista **`[4,0,3,1,0,2,10,7]`**:



4. Construir el procedimiento **`Poner_Bolitas_EnElTablero`**, que dada una lista de números y un color, recorre el tablero con dirección principal Norte y dirección secundaria Oeste y pone en cada celda la cantidad de bolitas del color dado, según el elemento en la posición correspondiente. Si hubiera menos números que celdas, en las celdas restantes no se pondrán bolitas, y si hubiera más números que celdas, se ignoran los números sobrantes.
   Por ejemplo **`Poner_Bolitas_EnElTablero([4,0,3,1,0,2,10,7],Rojo)`**, si se ejecuta en un tablero inicial vacío de 4 columnas y 2 filas, produce el tablero de la figura del ejercicio anterior.

5. Construir la función **`coloresEnLaCeldaActual`**, que describa la lista de los colores que aparecen en la celda actual. ¿Cuál es el número mínimo de elementos que tendrá esta lista? ¿Y el máximo?

6. Construir la función **`aparicionesDelColor_EnElCamino_`**, que dado un color, **`colorBuscado`**, y una lista de Direcciones, **`direccionesDelCamino`**, describa la lista con la cantidad de bolitas del color dado en las celdas recorridas, si las mismas se recorren en el orden en que indica el camino dado, incluyendo la celda en que se encuentra el cabezal antes de empezar y al finalizar el recorrido. ¿Cuál es la precondición del procedimiento?

7. Construir la función **`longitudDe_`**, que dada una lista cualquiera, describa la cantidad de elementos de la misma.
   Por ejemplo, **`longitudDe_([Azul,Azul,Verde,Rojo])`** describe **`4`**.

8. Construir las funciones

   a. **`sumatoriaDe_`**, que dada una lista de Números, describa la suma de todos los elementos de la misma.
      Por ejemplo, **`sumatoriaDe_([1,10,15,7,9])`** describe el número **`42`**, porque **`1+10+15+7+9`** es igual a **`42`**;

   b. **`productoriaDe_`**, que dada una lista de Números, describa el producto de todos los elementos de la misma.
      Por ejemplo, **`productoriaDe_([1,5,7,9])`** describe el número **`315`**, porque **`1*5*7*9`** es igual a **`315`**.

   c. **`cantidadDeBolitas_DelTablero`**, que dado un color, indique la cantidad total de bolitas de ese color que hay en el tablero. ¿Puede hacerse reutilizando código ya hecho?
      AYUDA: pensar en combinar algunas de las funciones hechas en ejercicios anteriores con alguna de las de este.

9. Construir las funciones

a. **repetición_VecesDe_**, que dado un número y un elemento de cualquier tipo, describa una lista con la cantidad indicada de repeticiones del elemento dado. Si el número es menor o igual que cero, retorna la lista vacía.
   Por ejemplo, **repetición_VecesDe_(3, 8)** describe la lista **[8,8,8]**, mientras que **repetición_VecesDe_(5, Verde)** describe la lista **[Verde,Verde,Verde,Verde,Verde]**.

b. **laLista_Clonada_Veces**, que dados una lista de elementos y un número, describa la lista resultante de clonar la lista dada tantas veces como se indica.
   Por ejemplo, **laLista_Clonada_Veces([Rojo,Azul,Verde], 3)** retorna **[Rojo,Azul,Verde,Rojo,Azul,Verde,Rojo,Azul,Verde]**

c. **losElementosDe_Clonados_Veces**, que dados una lista de elementos y un número, describa la lista que contenga los elementos dados en el orden de dicha lista, pero repetidos la *cantidad* de veces indicada.
   Por ejemplo, **losElementosDe_Clonados_Veces([Rojo,Azul,Verde], 3)** retorna **[Rojo,Rojo,Rojo,Azul,Azul,Azul,Verde,Verde,Verde]**.

d. La solución anterior, ¿fue construida reutilizando **repetición_VecesDe_**? Si no es así, resolverla nuevamente utilizando esa función.

10. Construir la función **reversoDe_**, que dada una lista, describa la lista que tiene los mismos elementos que la dada, pero en orden reverso.
    Por ejemplo, **reversoDe_([Negro,Azul,Azul,Verde,Rojo])** describe **[Rojo,Verde,Azul,Azul,Negro]**.

11. Construir las funciones

    a. **direccionesOpuestasDe_**, que dada una lista de direcciones, describa la lista de direcciones en donde cada elemento es el opuesto al de la posición original.
       Por ejemplo, **direccionesOpuestasDe_([Oeste,Sur,Norte])** describe **[Este,Norte,Sur]**.

    b. **siguientesDe_**, que dada una lista de colores, describa la lista de colores en donde cada elemento es el siguiente del original.
       Por ejemplo, **siguientesDe_([Rojo,Azul,Verde,Azul,Azul])** describe **[Verde,Negro,Azul,Negro,Negro]**.

12. Construir la función **homologar_PorDebajoDe_A_**, que dada una lista de números, y dos números **umbral** y **default**, describa una lista de números que está basada en la lista dada de la siguiente manera: aquellos números de la lista dada que son mayores o iguales al **umbral**, permanecen igual, pero aquellos que son menores, son reemplazados por el valor **default**.

Por ejemplo, **`homologar_PorDebajoDe_A_([3,7,8,5,1,3,2,4], 4, 2)`** describe **`[2,7,8,5,2,2,2,4]`**.

13. Construir las funciones

    a. **`númerosParesDe_`**, que dada una lista de Números, describa la lista de números pares que aparecen en la misma.
    Por ejemplo, **`númerosParesDe_([3,4,5,2,5])`** describe **`[4,2]`**.

    b. **`laLista_SinElElemento_`**, que dados una lista y un elemento, describa la lista que resulta de quitar todas las apariciones del elemento dado que ocurren en lista dada. ¿De qué tipo debe ser elemento?
    Por ejemplo, **`laLista_SinElElemento_([Azul,Verde,Azul,Rojo], Azul)`** describe **`[Verde,Rojo]`**.

14. Construir la función **`contiene_A_`**, que dada una lista *y* un elemento, indica si el elemento está en la lista.
    Por ejemplo, **`contiene_A_([21,3,42], 3)`** indica que es verdadero que la lista contiene a **`3`**, mientras que **`contiene([21,3,42], 5)`** indica que es falso que la lista contiene a **`5`**.

15. Construir la función **`sinDuplicados_`**, que dada una lista, describa una lista que tenga todos los elementos de la lista dada, pero donde no aparecen elementos repetidos, pues las repeticiones que aparecen luego de la primera fueron eliminadas.
    Por ejemplo, **`sinDuplicados_([1,3,4,2,4,3,5])`** describe a la lista **`[1,3,4,2,5]`**. Observar que no es lo mismo describir a la lista **`[1,2,4,3,5]`**, que podría ser un resultado válido, pero no es el solicitado (porque no se conservó el primero de cada uno).

16. Construir las funciones

    a. **`intersecciónDe_Con_`**, que dadas dos listas que no contienen elementos repetidos, describe la lista de todos los elementos que aparecen en ambas.
    Por ejemplo: **`intersecciónDe_Con_([1,3,4], [2,4,3,5])`** describe **`[3,4]`**.

    b. **`uniónDe_Con_`**, que dadas dos listas que no contienen elementos repetidos, describe una lista sin repetidos que contenga todos los elementos que aparecen en alguna de las 2 listas.
    Por ejemplo, **`uniónDe_Con_([1,3,4],[2,4,3,5])`** describe **`[1,3,4,2,5]`**. ¿De qué tipo es la primera lista? ¿Y la segunda? ¿Pueden ser de tipos diferentes?

17. Construir la función **`algunoDe_Entre_Y_`**, que, dada una lista de Números *y* dos números **`desde`** *y* **`hasta`**, indica si la lista contiene algún elemento que se encuentre

entre los números desde y hasta, sin incluirlos. Es decir, si algún elemento *k* de la lista, cumple **desde** < *k* < **hasta**.
Por ejemplo, **algunoDe_Entre_Y_([7,3,1,25,16], 13, 18)** indica que es verdadero que hay un elemento entre **13** y **18** (el **16**), mientras que **algunoDe_Entre_Y_([7,3,1,25,16], 13, 16)** indica que es falso que haya algún elemento entre **13** y **16** (el **16** no es menor que **16**).

¿La solución propuesta requirió el uso de variables que recuerdan valores booleanos? Si fue así, considerar ofrecer una solución que NO utilice variables booleanas.

18. Construir la función **lista_estáIncluidaEn_** que dadas 2 listas que no contienen elementos repetidos, describe si la primer lista se encuentra contenida en la segunda lista (o sea, todos los elementos de la primera aparecen en la segunda).
Por ejemplo, las expresiones **lista_estáIncluidaEn_([4,5],[5,3,4,6])** y **lista_estáIncluidaEn_([4,5],[2,3,4,6,5])** indican que es verdadero que está incluida, mientras que **lista_estáIncluidaEn_([4,5,8],[4,3,5,6])** indica que es falso que esté incluida.

19. Construir la función **estáOrdenada_** que, dada una lista de Números indica si está ordenada de menor a mayor. Para que una lista esté ordenada, cada elemento debe ser menor o igual al que le sigue.
Por ejemplo, la expresión **estáOrdenada_([2,7,9,15])** indica verdadero, mientras que la expresión **estáOrdenada_([2,15,9,7])** indica falso.

¿La solución propuesta requirió el uso de variables que recuerdan valores diferentes a la lista que todavía falta recorrer? Si fue así, considerar ofrecer una solución que NO utilice variables además de la que se utiliza para recordar la lista a recorrer.

20. Construir la función **posiciónDe_enLaQueAparece_** que dada una lista y un elemento, describe la posición de la lista en la que aparece ese elemento por primera vez. Se define la posición de un elemento como un número que representa la cantidad de veces que debe usarse la función **sinElPrimero** para acceder a ese elemento. ¿Cuál es la precondición de la función? ¿Por qué no tendría sentido que esta función sea total?
Por ejemplo, **posiciónDe_enLaQueAparece_([4,8,15,16,42],8)** describe al número **1**, porque solo se usa una vez **sinElPrimero** (para quitar el **4**), **posiciónDe_enLaQueAparece_([4,8,15,16,42],16)** describe **3**, porque se deben usar 3 veces la función (para quitar el 4, el 8 y el 15), y **posiciónDe_enLaQueAparece_([4,8,15,16,23,42],4)** describe **0**, porque no hace falta usar **sinElPrimero**.

21. Construir la función **sinLaPrimeraAparicionDe_en_** que dado un *elemento* y una lista, describe la lista que se obtiene de eliminar una única vez el *elemento*, si es que este aparece en la lista.
Por ejemplo, **sinLaPrimeraAparicionDe_en_(8,[4,8,42,15,8,42])** describe

`[4,42,15,8,42]` y `sinLaPrimerAparicionDe_en_(42,[4,8,42,15,8,42])` describe `[4,8,15,8,16,42]`.

22. Construir las siguientes funciones:

    a. **`mínimoElementoDe_`**, que dada una lista de Números, describe el elemento más chico que se encuentra en la lista. ¿Cuál es la precondición de la función?
    Por ejemplo, **`mínimoElementoDe_([13,21,3,9,45,3,7])`** describe `3`.

    b. **`sinElMínimoElemento_`**, que dada una lista de Números, describe la lista que se obtiene de eliminar una única vez el elemento más chico. ¿Cuál es la precondición de la función?
    Por ejemplo, **`sinElMínimoElemento_([13,21,3,9,45,3,7])`** describe `[13,21,9,45,3,7]`.

    c. **`lista_ordenada`**, que dada una lista de Números, describe la lista con los mismos elementos que la dada, pero ordenada de menor a mayor. ¿Se puede elaborar una estrategia combinando los ejercicios anteriores?
    Por ejemplo, **`lista_ordenada([13,21,3,9,45,17,8,3,7])`** describe `[3,3,7,8,9,13,17,21,45]`.

    d. ¿Puede hacerse la función anterior con alguna otra estrategia diferente a esa, que no sea simplemente utilizar la variante de funciones del ejercicio siguiente…? Este tema es un tema amplio que se tratará en detalle en la materia Estructuras de Datos.

23. Construir las siguientes funciones. ¿Cuáles son las precondiciones de estas funciones?

    a. **`máximoElementoDe_`**, que dada una lista de Números, describe el elemento más grande que se encuentra en la lista.
    Por ejemplo, **`máximoElementoDe_([13,21,3,9,45,3,7])`** describe `45`.

    b. **`sinElMáximoElemento_`**, que dada una lista de Números, describe la lista que se obtiene de eliminar una única vez el elemento más grande.
    Por ejemplo, **`sinElMáximoElemento_([13,21,3,9,45,3,7])`** describe `[13,21,3,9,3,7].`

24. Construir la función **`segmentoInicialDeLargo_de_`** que, dado un número que representa a una cantidad y una lista de cualquier tipo, describe la lista que se obtiene de quedarse únicamente con esa cantidad de elementos de la lista comenzando por el primero (sus primeros elementos), o todos, si no hay suficientes. ¿Cuál es la precondición de esta función? ¿Qué nombre te parece adecuado en este caso para el parámetro de tipo número? ¿Para qué tipos de lista funciona tu solución?
    Por ejemplo, **`segmentoInicialDeLargo_de_(4,[4,8,15,16,99,42])`** describe `[4,8,15,16]`, mientras que **`segmentoInicialDeLargo_de_(2,[4,8,15,42])`**

describe `[4,8]` y `segmentoInicialDeLargo_de_(7,[4,8,15,42])` describe `[4,8,15,42]`.

25. Construir la función **`agrupar_deA_`**, que dadas una lista de elementos cualesquiera y un número que representa a una cantidad, describe la lista de listas que agrupa los elementos de la lista dada en grupos, donde cada grupo tiene exactamente la cantidad de elementos indicada, excepto el último grupo, que puede tener menos elementos.
    Por ejemplo, la expresión `agrupar_deA_([1,2,3,4,5,6,7,8,9,10,11],3)` describe a la lista `[[1,2,3],[4,5,6],[7,8,9],[10,11]]`. Observar que la la lista original tiene 11 elementos, mientras que la lista resultante tiene 4.

    Para realizarla, considerar utilizar la función del ejercicio anterior al pensar una estrategia. ¿Qué otra subtarea haría falta para completar dicha estrategia?

26. Construir la función **`desagruparLista_`**, que dada una lista de listas, describe una lista que contiene cada uno de los elementos de las listas internas de la lista dada.
    Por ejemplo, `desagruparLista_([[1,2,3],[4,5,6],[7,8,9]])` describe la lista `[1,2,3,4,5,6,7,8,9]`. Observar que la lista argumento tiene 3 elementos, mientras que la lista resultado tiene 9.

27. Construir la función **`laLista_SinLosPrimeros_`**, que dados una lista de elementos de cualquier tipo y un número que representa a una cantidad, describe la lista que se obtiene de quedarse con todos los elementos excepto el segmento inicial del largo dado.
    Por ejemplo, `laLista_SinLosPrimeros_([4,8,15,23,42],4)` describe `[42]`, `laLista_SinLosPrimeros_([4,8,16,23,9,42],4)` describe `[9,42]`, y la expresión `laLista_SinLosPrimeros_([4,8,15,16,23,42],8)` describe `[]`.

    ATENCIÓN: según qué estrategia se haya utilizado al escribir la función **`agrupar_deA_`**, es posible que ya se haya resuelto este ejercicio como subtarea de aquél (quizás con otro nombre). En ese caso, ¿sería necesario volverlo a escribir? ¿Puede reutilizarse lo hecho con anterioridad?

28. Construir la función **`sinInternos_`**, que dada una lista de Números, describa la lista que se obtiene de quitar todos los elementos internos. Un elemento de una lista se dice interno si es igual al anterior de la lista.
    Por ejemplo, `sinInternos_([1,1,2,2,2,3,1,2,2])` describe `[1,2,3,1,2]`.