Introducción a la Programación - Práctica 9

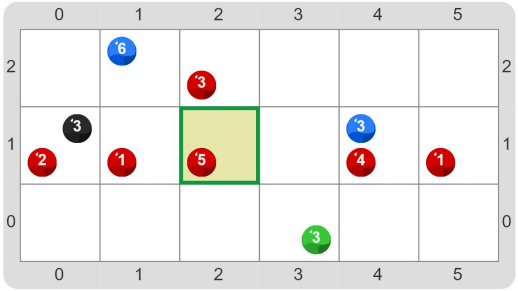
Listas

CONSEJOS:

- Leer el enunciado en su totalidad y pensar en la forma de resolver el ejercicio ANTES de empezar a escribir 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. Escribir la función **`cantidadesDeRojasEnFilaActual`** que describe una Lista de números cuyos elementos son las cantidades de bolitas rojas en cada una de las celdas de la fila actual, leídas desde el Oeste hasta el Este.
   Por ejemplo, dado el tablero de más abajo, la función describiría la siguiente lista: **`[2,1,5,0,4,1]`**.



2. Escribir la función **`cantidadDeRojasADistancia2DelBordeOesteDe_`** que dada una lista de números que representa las cantidades de bolitas rojas de una fila cualquiera, describa la cantidad de bolitas rojas que había en la celda a distancia 2 del borde Oeste de la fila dada al momento de su lectura. Por ejemplo, dada la lista del ejemplo anterior, debería describir el número **`5`**.

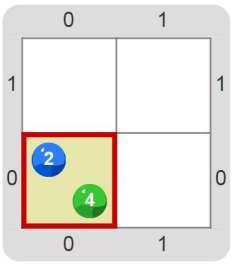
3. Escribir las funciones dadas a continuación, que reciben como parámetros una lista de números que representa las cantidades de bolitas rojas de una fila cualquiera:

   a. **`las3PrimerasDe_ConLaMismaCantidad`**, que indica si las primeras 3 celdas al Oeste de la fila representada tenían la misma cantidad de bolitas rojas al momento en que la fila fue leída.

b. **`las3PrimerasDe_ConDistintasCantidades`** que indica si en las primeras 3 celdas al Oeste hay diferente cantidad de bolitas de rojas (es decir, son todas distintas).

c. **`hayAlgunaCantidadIgualEnLas3PrimerasDe_`** que determine si en algún caso hay la misma cantidad de bolitas de dos colores (por ejemplo, la primera y la tercera celda tienen la misma cantidad de bolitas rojas, o la segunda y la tercera, etc.).

4. Escribir la función **`direccionesAlBorde`** que retorne la lista de direcciones en las que el cabezal no se puede mover. En el siguiente tablero **`direccionesAlBorde`** describe la lista **`[Sur, Oeste]`**, ya que el cabezal en este caso no puede moverse ni al Sur, ni al Oeste.



a. ¿Cómo puede utilizarse **`direccionesAlBorde`** para determinar si el tablero tiene una única celda?

5. Escribir la función **`esSingular_`**, que dada una Lista de elementos de cualquier tipo, indique si la lista tiene exactamente un único elemento.
SUGERENCIA: pensar en combinación con qué otras operaciones puede aprovecharse la expresión primitiva **`esVacía`**.

6. Escribir las siguientes funciones que toman un parámetro **`ruta`** que recibe una lista de Direcciones. Tener en cuenta que el siguiente tramo de la ruta solamente se compone de las primeras dos direcciones de la ruta si existen.

a. **`haySiguienteTramoEn_`**, función total que indique si la ruta tiene al menos un tramo completo.

b. **`sigueUnaCurvaEn_`**, función total que indique si el tramo que sigue en la ruta es una curva.

c. **`sigueCurvaADerechaEn_`**, que, teniendo en cuenta que viene una curva, indique si la curva que sigue en la ruta es hacia la derecha.

d. **`sigueCurvaAIzquierdaEn_`**, que, teniendo en cuenta que viene una curva, indique si la curva en la ruta es hacia la izquierda.

e. **`sigueRectaEn_`**, que indique si el tramo que sigue en la ruta es una recta.

7. Escribir las siguientes funciones que utilizan listas de elementos del tipo **`Carta`** definido en clase. Tener en cuenta que la mano de un jugador está representada por una **`Lista de Cartas`** dispuestas en el orden en el que las va jugar. No olvidarse de establecer las precondiciones necesarias.

    a. **`primerCartaDeLaMano_`**, que dado un parámetro **`cartasEnLaMano`**, que recibe la mano de un jugador, describa la primera carta a jugar.

    b. **`segundaCartaDeLaMano_`**, que dada una mano, describa la segunda carta a jugar.

    c. **`tercerCartaDeLaMano_`**, que dada una mano, describa la tercera carta a jugar.

    d. **`laMano_LuegoDeRobarUnaCartaDe_`**, que dada una mano y un mazo de Cartas del cual puede robar, describir la mano resultante luego de robar.

    e. **`laMano_LuegoDeJugarUnaCarta`**, que dada una mano, describir la mano resultante luego de jugar una carta en su turno.

    f. **`laMano_LuegoDeJugarLaSegundaCarta`**, que dada una mano, describir la mano resultante luego de jugar exclusivamente la segunda de todas sus cartas.

8. Escribir las siguientes funciones que continúan utilizando listas de elementos del tipo **`Carta`**. Una tira de cartas es una secuencia de cartas en orden creciente; para agregar una carta a una tira, debe ser de diferente palo y menor a la primera carta de la tira.

    a. **`primerasTresCartasDeLaTira_`**, que dado una tira, describa una lista con las primeras 3 cartas de la misma.

    b. **`laMano_TieneJugadaParaAgregarALaTira_`**, que dada una mano de cartas de un jugador, y una tira de cartas, indique si la primera carta que puede jugar el jugador puede ser agregada en la tira o no.

    c. **`laTira_DespuésDeJugar_`**, que dada una tira de cartas y una carta que cumple las condiciones para ser agregada en la misma, describa la tira que resulta de jugar esa carta en esa tira.

9. Escribir las siguientes funciones que continúan utilizando listas de elementos de tipo **`Carta`**. Si bien varias de las funciones están inspiradas en el juego del Truco, no es necesario conocer las reglas para realizarlas. Solo debe saberse que las Figuras son las cartas con números 10, 11 y 12, ya que se dibujan en las cartas reales utilizando las figuras de una Sota (una especie de paje o escudero), un Caballo (usualmente con su caballero) y un Rey; y además, que el valor de una carta es el número de la misma si no es una figura, o cero si es una figura.

    a. **`manoDe3CartasDe_`**, que dado un mazo, describa una mano de 3 cartas tomadas del mismo

b. **`tieneEnvido_`**, que dada una mano de 3 cartas, indique si tiene exactamente 2 cartas del mismo **`Palo`**.

c. **`tieneFlor_`**, que dada una mano de 3 cartas, indique si las 3 cartas son del mismo **`Palo`**.

d. **`cuántoDeFlor_`**, que, dada una mano de 3 cartas, describe la suma de los valores de las cartas más 30 en caso que tenga flor, o cero en caso contrario. Recordar que el valor de las figuras es 0.

e. **`cuántoDeEnvido_`**, que, dada una mano de 3 cartas, describe cuánto se suma para un envido. Este valor es: si tiene envido, la suma de los valores de las cartas del mismo palo más 20; si tiene flor, la suma de las 2 cartas de mayor valor más 20; y si no tiene ni envido ni flor, el valor de la carta de mayor valor. Recordar que el valor de las figuras es 0.

10. Se desea modelar boletos de colectivo como eran antes de existir la Sube. Todo boleto tenía que contener ciertos datos que eran obligatorios por las regulaciones vigentes en la época. En particular, un boleto debe contener
    - el número de línea al cual pertenece el boleto (p.ej. 324, 159, 257, 85, etc.),
    - la cantidad de tramos para la cual se sacó dicho boleto (siendo 1 el mínimo y 4 el máximo),
    - la serie, que puede estar compuesta de números y letras, y
    - el número del boleto, que corresponde a un número siempre de cinco dígitos.

    Definir el tipo **`BoletoDeColectivo`**, con los datos necesarios para modelar el boleto de un colectivo según fue explicado. No olvidar escribir los tipos de los campos, y la invariante de representación.

11. Pensar una estrategia para realizar la función **`esCapicúa_`**, que dado un boleto, determine si el número del mismo es capicúa. Un boleto es capicúa cuando el primer dígito es igual al último, y el segundo al anteúltimo.
    ¿Fué fácil la pensar la estrategia? ¿Puede usarse otra representación de **`BoletoDeColectivo`** que haga más fácil resolver el problema?

12. Construir las siguientes funciones sobre boletos.

    a. **`costoDeBoleto_`**, que determina el costo del boleto. El costo de un boleto es de $18, si se abona un solo tramo, y aumenta en $3 por cada tramo adicional.

    b. **`elBoleto_TieneElMismoNúmeroQue_`**, que dados dos boletos, **`boleto1`** y **`boleto2`**, indique si ambos boletos tienen el mismo número de boleto.

    c. **`elBoleto_EsParejaCapicúaDe_`**, que dados dos boletos, **`boleto1`** y **`boleto2`**, indique si el número del primer boleto es igual al número del segundo boleto dado vuelta. Por ejemplo, los boletos con números 12345 y 54321 serían pareja capicúa.

    d. **`esCapicúaDePrimeraSerie_`**, que dado un boleto, indique si el mismo es capicúa de una primera serie. La primera serie debe ser la serie “1” o serie “A”.