

Introducción a la Programación - Práctica 4

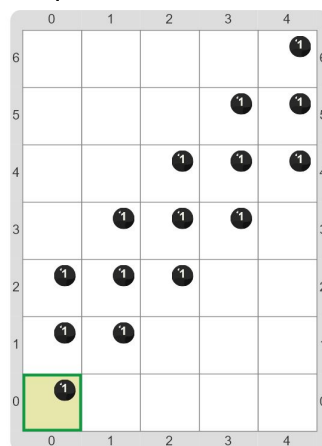
Expresiones y tipos

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 teoría correspondiente.
- Algunos ejercicios están tomados de la guía complementaria realizada por Federico Aloí 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.

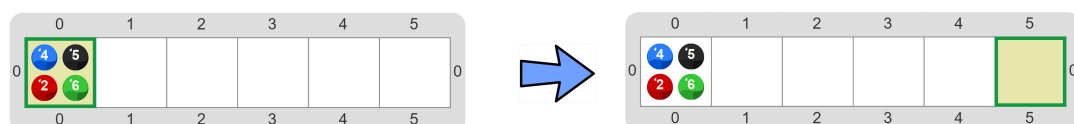
EJERCICIOS:

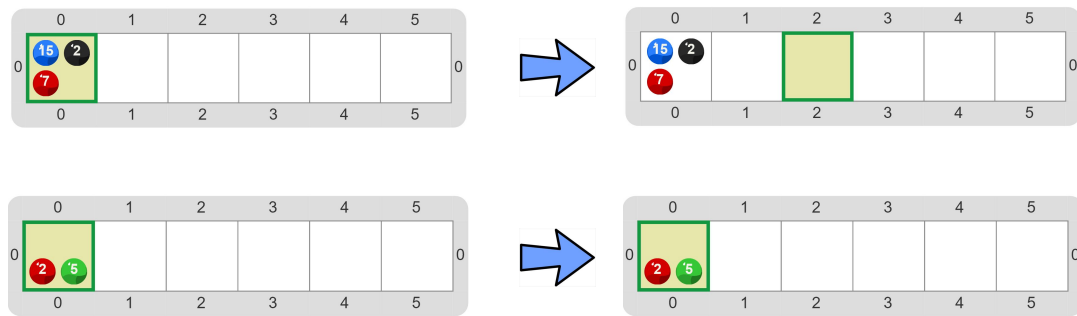
1. Escribir `DibujarBanda_EnDiagonalNEDe_x_(color, alto, largo)` que dibuja una banda diagonal ancha. Por ejemplo, en la figura se observa el resultado de ejecutar `DibujarBanda_EnDiagonalNEDe_x_(Negro, 3, 5)` en un tablero de 5x7, comenzando en la esquina SO del tablero.



Observar que 3 es el alto de la banda, y 5 el largo.

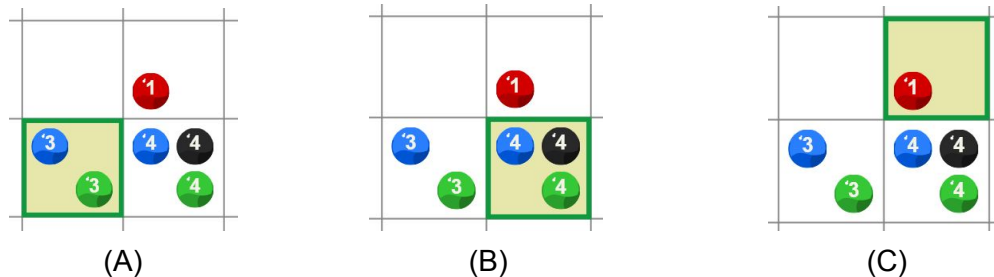
2. Escribir el procedimiento `Mover_SegúnColor_(dirección,color)`, que mueve el cabezal en la dirección dada tantas celdas como bolitas de color dado hay en la celda actual. Como ejemplos se ofrecen los resultados de evaluar el comando `Mover_SegúnColor_(Este, Negro)`, en diferentes tableros iniciales.





En el último caso, como la celda no tiene bolitas negras (o sea tiene 0 bolitas negras), entonces el cabezal no se mueve (o sea, se mueve 0 celdas hacia el Este).

3. Indicar el valor y el tipo que representan las expresiones dadas en los ítems en cada uno de los tableros A , B y C, suponiendo definido un procedimiento con el contrato dado al final.



- a. `nroBolitas(Negro) + nroBolitas(Azul)`
- b. `opuesto(opuesto(Este))`
- c. `nroBolitas(siguiente(Azul))`
- d. `2*nroBolitas(colorAImitar)`

suponiendo que esta expresión aparece dentro del cuerpo del procedimiento `PonerElDobleDe_QueDe_`, y que se lo invocó como `PonerElDobleDe_QueDe_(Rojo, Azul)`

El contrato del procedimiento dado es el siguiente:

```

procedure PonerElDobleDe_QueDe_(colorAPoner
                                , colorAImitar)
/* PROPÓSITO: Poner bolitas del colorAPoner en una cantidad
   que sea el doble de las que hay del colorAImitar en la
   celda actual
PRECONDICIONES: ninguna
PARÁMETROS:
  * colorAPoner, un color del que se pondrán bolitas
  * colorAImitar, un color del que se mirará cuántas
                 bolitas hay en la celda actual
*/

```

4. El bosque, parte 1

En este ejercicio, se usará el tablero para representar un bosque. Cada celda representa a una parcela. Cada bolita verde representa un árbol. Cada bolita roja representa una semilla. Una bolita negra representa una bomba. Una bolita azul representa una unidad de nutrientes.

Escribir los siguientes procedimientos de representación, que hacen lo que su nombre indica (todos, salvo los últimos, trabajan sobre la celda actúa, y esos dos solo trabajan sobre la celda lindante correspondiente):

- a. `Poner_Semillas(cantidadDeSemillasAPoner)`
- b. `Sacar_Semillas(cantidadDeSemillasASacar)`
- c. `Poner_Árboles(cantidadDeÁrbolesAPoner)`
- d. `Sacar_Árboles(cantidadDeÁrbolesASacar)`
- e. `PonerBomba()`
- f. `SacarBomba()`
- g. `Poner_Nutrientes(cantidadDeNutrientesAPoner)`
- h. `Sacar_Nutrientes(cantidadDeNutrientesASacar)`
- i. `Poner_SemillasAl_(cantidad, dirección)`
// deja el cabezal en la celda inicial
- j. `Sacar_ÁrbolesAl_(cantidad, dirección)`
// deja el cabezal en la celda inicial

5. Hacer el procedimiento `PasarPalabraActualAMinúsculas()` que suponiendo que en la fila actual se codifica una palabra en minúsculas usando bolitas, ponga la misma palabra en mayúsculas en la fila al Norte.

- Cada letra se representan con una cantidad diferente de bolitas negras, según un código numérico llamado ASCII.
- En la celda más al Oeste de la fila actual se codifica la cantidad de letras usando bolitas rojas.
- La primera letra de la palabra está en la celda lindante al Este de la que contiene la cantidad de letras.
- En el código ASCII si las letras mayúsculas se codifican con un número N entonces la misma letra minúscula se representa con N+32 (pej. la 'a' minúsculas se representa con el número 97 y la 'A' mayúsculas, con el 65).
- El cabezal se encuentra en la celda más al Oeste de una fila donde hay una palabra representada.

Ejemplo de (fragmento de) un tablero inicial posible:

2												
1	9	112	114	111	103	114	97	109	97	114		
		*	*	*	*	*		*		*		

Ejemplo del (fragmento de) tablero final correspondiente al anterior:

2		80	82	79	71	82	65	77	65	82		
1	9	112	114	111	103	114	97	109	97	114		
		*	*	*	*	*		*		*		

¿Cómo comenzar la resolución? En cada procedimiento, ¿qué parte debe escribirse primero?

6. Escribir un procedimiento **SacarTodas_(colorASacar)**, que quite de la celda actual todas las bolitas del color indicado por el parámetro.

Ayuda: considerar utilizar el procedimiento **Sacar_Veces**, definido en la práctica anterior. ¿Qué argumentos se le deberían pasar?

7. Suponiendo que se está programando un juego donde en las celdas del tablero se representan Soldados (los aliados con una bolita de color Negro y los enemigos con una bolita de color rojo por cada soldado), escribir los siguientes procedimientos:

- a. **EnviarSoldadosParaDuplicarEnemigos()**, que agrega soldados aliados en la celda actual en cantidad suficiente para que haya el doble de aliados que de soldados enemigos.

- b. **PelearLaBatalla()**, que simula una batalla, suponiendo que hay suficiente cantidad de soldados aliados como para ganar la batalla. Durante una batalla, 2 soldados enemigos pelean contra 3 soldados aliados y todos mueren. Por ejemplo, si hay 6 enemigos y 10 aliados, mueren los 6 enemigos y 9 de los aliados; si hay 11 enemigos y 21 aliados, mueren los 11 enemigos y 15 soldados aliados.

Ayuda: ¿qué cuenta hay que hacer para saber cuántos soldados aliados morirán?

Ayuda 2: para la división en Gobstones se usa el operador **div**.

8. Realizar el ejercicio “Soporte técnico” que se encuentra en el capítulo “Ejercicios de práctica” del curso online

(<https://gobstones.github.io/gobstones-sr/?course=gobstones/curso-InPr-UNQ&github=gobstones/curso-InPr-UNQ&path=Proyectos/12.Ejercicios%20de%20pr%C3%A1ctica/2.Soporte%20T%C3%A9cnico>)