

## Práctica 6

### Introducción a CGOBSTONES

Introducción a la Programación  
2<sup>do</sup> Semestre de 2013

#### 1. Procedimientos Básicos

##### Ejercicio 1

Escribir el procedimiento `main` que ponga una bolita azul en la celda en que se encuentre el cabezal. Probar su ejecución en CGOBSTONES.

##### Ejercicio 2

Escribir un procedimiento que ponga una bolita roja en la celda del cabezal, si el mismo se encuentra en el extremo este del tablero.

##### Ejercicio 3

Escribir un procedimiento que ponga una bolita roja en la celda del cabezal si el mismo se encuentra en el borde del tablero.

##### Ejercicio 4

Escribir un procedimiento que vacíe la columna en la que se encuentra el cabezal.

##### Ejercicio 5

Escribir un procedimiento que reemplace cada una de las bolitas del tablero por su color siguiente.

#### 2. Procedimientos con parámetros

##### Ejercicio 6

Escribir un procedimiento `PonerAl` que, dada una dirección  $d$  y un color  $c$ , ponga una bolita de color  $c$  en la celda lindante en dirección  $d$ . El procedimiento debe ser total; si no se puede poner dicha bolita, el procedimiento no hace nada.

##### Ejercicio 7

Escribir el procedimiento `MoverN` que, dado un número  $n$  y una dirección  $d$ , mueva el cabezal  $n$  posiciones en dirección  $d$ .

**Ejercicio 8**

Escribir el procedimiento `UbicarCabezal` que, dado un número  $x$  y una dirección  $d$ , ubique el cabezal en la celda que se obtiene de la siguiente forma. Primero se mueve el cabezal al extremo opuesto( $d$ ) y luego se mueve el cabezal  $x$  posiciones en dirección  $d$ . Por ejemplo, `UbicarCabezal(3, Este)` debería ubicar el cabezal en la tercer columna del tablero. De ser necesario, escribir el procedimiento `IrAlExtremo`.

**Ejercicio 9**

Escribir los procedimientos `IrAFila` e `IrAColumna` que, dado un numero  $x$ , ubiquen el cabezal en la fila y columna  $x$ , respectivamente. **Hint:** usar el ejercicio anterior.

**Ejercicio 10**

Escribir el procedimiento `IrACelda` que, dados dos números  $x$  e  $y$ , ubique el cabezal en la fila  $x$  y en la columna  $y$  del tablero. **Hint:** usar el ejercicio anterior.

### 3. Procedimientos que devuelven valores

**Ejercicio 11**

Escribir un procedimiento `nroBolitasAl` que, dada una dirección  $d$ , retorne la cantidad de bolitas en dirección  $d$ . ¿En qué posición queda ubicado el cabezal y por qué? Si fuera necesario, modificar el procedimiento de forma tal que el cabezal quede en ubicación inicial.

**Ejercicio 12**

Escribir un procedimiento `aislada` que retorne un booleano indicando si todos los vecinos de la celda actual (en direcciones N, E, S, O) están vacíos. Utilizar `nroBolitasAl` y detallar qué ocurre con el cabezal.

**Ejercicio 13**

Escribir un procedimiento `vecinosConColor` que, dado un color  $c$ , retorne un número indicando la cantidad de vecinos de la celda actual (en direcciones N, E, S, O) que tienen el color  $c$ .

**Ejercicio 14**

Escribir una procedimiento `fila` que retorne un número indicando la fila en que se encuentra el cabezal. ¿Qué ocurre con el cabezal? Si fuera necesario, modificar el procedimiento de forma tal que el cabezal quede en ubicación inicial.(ayudarse del procedimiento `IrAFila` del Ejercicio 9).

**Ejercicio 15**

Escribir una procedimiento `columna` que retorne un número indicando la columna en que se encuentra el cabezal. Al finalizar, el cabezal debe quedar en la posición inicial.

**Ejercicio 16**

Escribir un procedimiento que reciba un color como parámetro y retorne la cantidad de bolitas de ese color en el tablero. ¿Qué ocurre con el cabezal? Si fuera necesario, modificar el procedimiento de forma tal que el cabezal quede en ubicación inicial. **Hint:** usar los Ejercicios 8, 14 y 15.

**Ejercicio 17**

Escribir un procedimiento que reciba una dirección como parámetro y retorne la cantidad de celdas en esa dirección hasta el borde del tablero.

**Ejercicio 18**

Escribir un procedimiento que devuelva el color del que haya más bolitas en todo el tablero.

## 4. Ejercicios con números (opcional)

**Ejercicio 19**

Definir las siguientes funciones entre números enteros

1. `cubo`, que dado un número  $x$  devuelva  $x^3$  (recordar:  $x^3 = x \times x \times x$ ).
2. `areaRectangulo`, que dadas la base y la altura de un rectángulo devuelva su superficie.
3. `abs`, que dado un número, retorne su valor absoluto (i.e., quitándole el signo).
4. `max`, que dado un par de números, devuelva el máximo entre los dos.
5. `max3`, que dados 3 números, devuelva el máximo entre ellos.  
Aprovechar la función `max` realizada en el punto anterior para implementar esta función con una sola línea de código.
6. `divideA`, que dados dos números naturales, determine si el primero divide al segundo (i.e., si el resto de la división es 0).
7. `esMultiploDe`, que dados dos números naturales, determine si el primero es múltiplo del segundo.  
Utilizar la función `divideA` para resolverlo sin utilizar condicionales en la definición de `esMultiploDe`.
8. `esPar`, que dado un número determine si es par.  
Utilizar la función `divideA` para resolverlo sin utilizar condicionales en la definición de `esPar`.

**Ejercicio 20**

Definir los siguientes procedimientos que devuelven un valor utilizando repetición condicional:

1. `potencia`, que dados dos números  $x$  e  $y$ , devuelva  $x^y$ .

2. **factorial**, que dado un número positivo devuelva su factorial. El factorial de un número  $n$  es el numero  $1 \times 2 \times 3 \times \dots \times n - 1 \times n$ .
3. **sumaRango**, que dados dos números  $n$  y  $m$  devuelva la suma de todos los números enteros en el rango  $[n, m)$  (incluyendo  $n$  y excluyendo  $m$ ).
4. **sumaDigitos**, que dado un número devuelva la suma de sus dígitos. Por ejemplo, **sumaDigitos**(1782) es  $1 + 7 + 8 + 2$ .
5. **log2**, que dado un número positivo devuelva su logaritmo natural en base 2. El logaritmo natural en base 2 de un número  $n$  es el numero  $\ell$  tal que  $2^\ell \leq n$  y  $2^{\ell+1} > n$ .