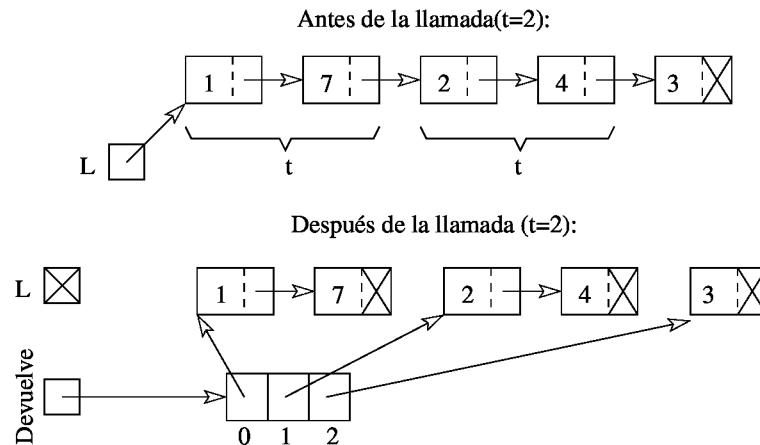


# Metodología de la Programación

Convocatoria ordinaria. Curso 2013/2014

7 de Julio de 2014

1. (1.0 puntos) Se desea dividir una lista de celdas enlazadas en secuencias de celdas de tamaño  $t$ . Escriba una función que recibe una lista de celdas enlazadas que almacena enteros y devuelva un vector de listas que contiene cada una de estas secuencias. La siguiente figura muestra gráficamente el efecto de la llamada:



Observe que en la parte superior se muestra la lista original y el valor  $t=2$ . La lista, que contiene 5 elementos tendrá que dividirse en 3 trozos. En la parte inferior se muestra el efecto de la llamada. Tenga en cuenta que:

- La lista original queda vacía.
- No es necesario reservar ni liberar ninguna celda.
- La última lista del vector podría quedar con menos de  $t$  casillas. En este ejemplo, se queda sólo con 1.
- Será necesario reservar el vector para almacenar cada una de las sublistas. En este caso es un vector con 3 listas.
- Cada lista es una secuencia de celdas enlazadas terminada en el puntero nulo.
- Puede suponer que la lista no está vacía.

Define una estructura adecuada para almacenar cada una de las celdas enlazadas y escriba la función que implementa la operación descrita.

2. Se desea resolver el problema del juego de los barquitos. Para ello, se propone diseñar una clase *Barquitos* que contiene el tablero de un jugador. La clase debe incluir una matriz de enteros para codificar el estado del tablero. El estado se codifica con una estructura de dos dimensiones de tamaño variable, reservada en memoria dinámica.

En la siguiente figura se presenta una matriz de codificación con enteros, la correspondiente representación gráfica y las indicaciones para entender la representación.

En este ejercicio debe implementar parte de la clase *Barquitos*. Para realizarlo, comience proponiendo la **representación de la parte privada de la clase**. Una vez establecida la representación, resuelva los siguientes apartados:

	1	2	3	4	5	6	7	8	9	10	11
A											
B		•	□				•				
C			•	•	•	×	×	×		•	
D			×				•				
E		□				×					
F							•			□	
G											
H		•			•						
I		•			•						
J			•							×	
K		□	□				•	•			
L							×	×		•	

	1	2	3	4	5	6	7	8	9	10	11
A	9	9	9	9	9	9	9	9	9	9	9
B	9	-9	1	9	9	9	-9	9	9	9	9
C	9	9	-9	-9	-9	-9	-3	-3	-3	-9	9
D	9	9	9	-4	9	9	-9	9	9	9	9
E	9	1	9	4	9	9	-1	9	9	9	9
F	9	9	9	4	9	9	-9	9	9	2	2
G	9	9	9	4	9	9	9	9	9	9	9
H	9	-9	9	9	-9	9	3	9	9	9	9
I	9	-9	9	9	-9	9	3	9	9	9	9
J	9	9	-9	9	9	9	3	9	9	-1	9
K	9	2	2	9	9	9	9	-9	-9	9	9
L	9	9	9	9	9	9	9	-2	-2	-9	9

Codificación:

Valor negativo: se ha disparado en la casilla

Valor 9: agua

Valor entero i: parte de un barco de i casillas

Ejemplos:

B3: valor 1. Barco de 1 casilla

B2: valor -9. Agua donde se ha disparado

D4: valor -4. Barco de 4 alcanzado

C8: valor -3. Barco de 3 alcanzado

- (0.75 puntos) Implemente el constructor de la clase y el destructor. Este constructor recibe las dimensiones del tablero –filas y columnas– y construye un tablero en memoria dinámica con todas las casillas con agua.
- (1.25 puntos) Implemente el constructor de copias y la sobrecarga del operador de asignación.
- (0.75 puntos) Implemente un método que recibe la posición de un barco y devuelve si es posible colocarlo. La posición de un barco viene determinada por:

- Una casilla: un carácter para codificar la fila (desde la 'A') y un entero para la columna (desde el 1). Puede suponer que no habrá más filas que letras consecutivas en la tabla ASCII.
- El tamaño del barco: un número positivo mayor que cero y menor que 9.
- La dirección de dibujo: 'H' o 'V' indicando horizontal-derecha y vertical-abajo, respectivamente.

Por ejemplo, en la figura anterior, el barco de tamaño 3 hundido (con todas las casillas tachadas) está en la posición: casilla C7, de tamaño 3 y dirección H.

*Nota: El barco se puede poner si las casillas no están ocupadas y no hay ningún barco con el que “se toque”. Por ejemplo, en la figura anterior no se puede poner un barco de tamaño 1 en la posición A2.*

- (1.0 puntos) Implemente un método para insertar un barco en el tablero. Este método debe recibir el tamaño del barco e insertarlo en una posición y dirección aleatorias. Supondremos como precondition que seguro que hay algún sitio en el tablero que permite insertarlo.

Para resolverlo suponga que dispone del método del apartado anterior y la siguiente función:

```
int Aleatorio (int min, int max); // un valor aleatorio del intervalo [min,max]
```

*Nota: Tenga en cuenta que para generar una dirección no tiene más que generar un valor Aleatorio(1,2) para escoger entre dos posibilidades.*

- En este ejercicio se desea ampliar la clase *Barquitos* con operaciones de E/S. Se desea almacenar el contenido de un tablero en un fichero. El formato de almacenamiento es el siguiente:

- Una cadena “mágica” compuesta por los caracteres “MP-BARQ-V1.0” seguida de un salto de línea.
- Un entero (número de filas), un espacio, un entero (número de columnas) y un salto de línea.
- Tantos valores enteros como casillas tiene el tablero codificados en binario.

- (0.75 punto) Implemente un método *Leer* que recibe el nombre de un fichero y lee el contenido del tablero. Devuelve si ha tenido éxito.
- (0.5 puntos) Implemente un método *Escribir* que recibe el nombre de un fichero y guarda el contenido del tablero. Devuelve si ha tenido éxito.

**Duración del examen: 2 horas y media.**