

## Hoja 5

**Importante:** Los ejercicios deben entregarse a través de web (**Domjudge y Blackboard**). Cada ejercicio deberá ir en un fichero con nombre:

`<<nombreEjercicio>>.cpp`

donde `<<nombreEjercicio>>` es el nombre indicado en negrita antes de cada ejercicio.

**La fecha de entrega:** consultar la página de la actividad en blackboard

**Laberinto (5 Puntos):** Un aventurero, un poco torpe ya que no sabe andar en diagonal, ha encontrado un laberinto que esconde en su interior un tesoro con multitud de monedas de oro. Sin embargo este laberinto es bastante complejo de recorrer así que necesita la ayuda de un sistema automático basado en backtracking que le diga cuál es el recorrido que debe realizar para llegar hasta el tesoro.

El laberinto estará descrito mediante un mapa de 10x10 caracteres cuyas celdas tendrán el siguiente significado:

- ‘\*’ Celda de pared que no puede atravesarse.
- ‘.’ Celda libre que puede atravesarse.
- ‘X’ Celda por la que ha pasado el aventurero.
- ‘T’ Celda donde se encuentra el tesoro.

El aventurero siempre empezará en la casilla (0,0) del laberinto y se moverá a una celda adyacente en cada paso pero al ser torpe no se moverá en diagonal e intentará moverse en las siguientes direcciones en orden si las celdas están libres:

- Arriba, Derecha, Abajo e Izquierda

El **input** presentará el mapa del laberinto de 10x10 con y con la codificación de celdas explicadas anteriormente. A continuación se presenta un ejemplo de mapa de laberinto:

Input:

```
. .*****
*.*.....*
*.*.****.*
*.*.*.*.*
*.*.*T.*.*
*.*.*.*.*
*.*.....*
*.******
*.*.....*
*****
```

## Algoritmos y Estructuras de Datos

Si el tesoro es alcanzable el programa deberá devolver el mapa con el recorrido del aventurero marcado en X y un mensaje diciendo “ENCONTRADO X Y” donde X e Y serán las coordenadas donde se encontró el tesoro.

Output:

```
XX*****  
*X*XXXXXX*  
*X*X****X*  
*X*X*XX*X*  
*X*X*TX*X*  
*X*X**X*X*  
*X*XXXX*X*  
*X*****X*  
*XXXXXXXX*  
*****  
ENCONTRADO 4 5
```

En el caso de que no se pueda alcanzar el tesoro se deberá imprimir “INALCANZABLE”.

Output:

```
INALCANZABLE
```

**DevolverCambio (3 Puntos):** Debido a la incierta situación del euro se está planteado volver a nuestra moneda original “La Peseta”. Este cambio supondría que habría que cambiar todas las máquinas expendedoras para que ofrezcan el cambio correctamente. Por tanto, será necesario implementar un sistema que sea capaz de devolver el cambio con el menor número de monedas usando un algoritmo voraz. En particular el tipo de monedas que se volverán a reeditar tendrán los siguientes valores:

{500 ptas, 200 ptas, 100 ptas, 50 ptas, 25 ptas, 10 ptas, 5 ptas, 1 peseta}

El **input** se leerá con el formato “%d\n”. Este entero estará indicando la cantidad de pesetas que se quiere devolver en monedas. El programa terminará cuando reciba una cantidad negativa.

El **output** utilizará el formato “%d %d %d %d %d %d %d %d \n” que corresponderá al número de monedas de cada tipo que habría que devolver, empezando por la de mayor valor 500ptas hasta la de menor valor 1 peseta. **OJO:** Observad que existe un espacio después de cada número incluso en el caso del último.

Input:

575	1 0 0 1 1 0 0 0
300	0 1 1 0 0 0 0 0
-1	

Output:

NOTA: Para cada una de las funciones implementados se deberá incluir una pequeña descripción de su funcionamiento, sus precondiciones mediante excepciones si las hubiera y el análisis de su complejidad temporal y espacial. Esta información deberá incluirse en la cabecera de cada función.