

## Práctica 6

### Uso de vectores, matrices y cadenas.

#### Objetivos:

*Manejo de vectores y matrices*

#### Desarrollo:

Queremos controlar un aparcamiento de un centro comercial. Para ello consideraremos que el aparcamiento posee 3 plantas en cada una de las cuales hay 4 secciones de coches. En cada sección pueden aparcarse 10 coches. Cada planta está pintada de un color y su nombre será el color (roja, azul y amarilla).

En el programa a elaborar ya se han implementado algunas funciones:

- `void vaciarAparcamiento(int m[MAX_PLANTAS][MAX_SECCION], int disponibles[]);`

Deja el parking vacío de coches aparcados.

- `void ocupacion (int m[MAX_PLANTAS][MAX_SECCION],  
int disponibles[MAX_PLANTAS], char colorPlanta[MAX_PLANTAS][15]);`

Muestra en pantalla el número total de plazas disponibles en cada planta, así como en cada sección de cada planta.

Habrà que implementar las funciones siguientes:

1. `int plazasLibresPlanta (int m[MAX_PLANTAS][MAX_SECCION], int planta)`

Esta función devolverà, dado un número de planta, la primera sección en la que haya plazas libres. Si no hay secciones con plazas libres, devolverà -1.

2. `??? aparcar (???)` A esta función se le pasará el número de la planta en la que se quiere aparcar y buscarà una sección libre en dicha planta mediante la función `int plazasLibresPlanta (int m[MAX_PLANTAS][MAX_SECCION], int planta)` que devolverà una sección con plazas libres. Si resulta que la planta no tiene plazas libres, buscare en la siguiente planta una sección que tenga plazas libres **hasta alcanzar la planta superior**. La función devolverà mediante la instrucción `return` el número de planta en la que ha aparcado el coche; si no hay plazas libres entre la planta solicitada y la planta superior devolverà -1. Esta función también devolverà por referencia el total de plazas disponibles en el parking.

Por último, habrá que completar el programa principal incluido en el esqueleto llamando a las funciones adecuadas en cada uno de los casos.

**Ejercicio opcional 1:** Haz una función en la que dado un color, **devuelva por referencia** el número de planta asociado al mismo. Modifica el programa principal para que, en la opción 2, en lugar de solicitarle al usuario un número de planta, se le solicite el color de la planta donde desea aparcar y usa la función anterior para obtener el número de planta asociado al color.

**Ejercicio opcional 2:** La función `aparcar` no parará al llegar a la última planta, seguirá buscando plazas libres en todo el parking, continuando por las plantas inferiores.