



Requisitos de memoria de un struct

Se dispone de una estructura con la siguiente definición.

```
#define MAX_CHARS_FABRICANTE 16
#define MAX_CHARS_MODELO 32

enum Motor {gasolina, gasoil, hidrogeno, electrico, ninguno};

typedef struct Vehiculo {
    char fabricante[MAX_CHARS_FABRICANTE];
    char modelo[MAX_CHARS_MODELO];
    enum Motor motor;
    int fecha;
    float longitud;
    char codigo;
    float tara;
    double cargaMaxima;
    double pesoMaxRemolcable;
} Vehiculo;
```

Realizar un programa que permita estudiar la ocupación en memoria de una variable estructura **Vehículo**, comparada con la suma de la ocupación de sus miembros.

Comprobar la ocupación de la estructura ...

- con todos los miembros.
- sin el miembro **codigo**.
- sin el miembro **tara**.
- sin el miembro **codigo** ni el miembro **tara**.

Puede utilizarse el operador **sizeof** para obtener la cantidad de espacio que ocupa una variable estructura **Vehículo**. Como se verá, este valor puede ser distinto del que se obtiene sumando los espacios ocupados por cada miembro, y no es un error: el compilador desplaza el comienzo de algunas variables para agilizar el acceso a las mismas. Las variables más pequeñas (**char**) son las que más “huecos” producen. A veces es posible reducir el espacio desperdiciado, declarando una tras otra aquellos miembros cuya suma de tamaños coincide con una palabra del ordenador.