

# Examen-parcial-de-Programacion-2...



albertomarcos



Programación II



1º Grado en Ingeniería Informática



Facultad de Ciencias  
Universidad de Salamanca



**VIVE AL 100% LA  
EXPERIENCIA DE JUGAR SIN  
USAR MOTHERLODE PARA  
COMPRARLO.**

**Gaming**

**msi**

Tener una laptop gamer no significa vender un riñón. MSI tiene opciones que no te hacen llorar al ver la cuenta bancaria. Básicamente, el "starter pack" para los que quieren entrar al mundo de los juegos de PC sin que su cartera sufra un infarto.





**VIVE AL 100% LA  
EXPERIENCIA DE JUGAR SIN  
USAR MOTHERLODE PARA  
COMPRARLO.**

**msi**

Alberto Marcos

## Enunciado examen parcial de Programación 2 para practicar.

*(Enunciado propio e inventado, pero muy similar al parcial de (2021-22) con el objetivo de practicar y con preguntas muy posiblemente semejantes para el examen parcial de este año (2022-2023).)*

### Ejercicio 1: (Posible ejercicio sobre structs)

**Apartado A):** Declara una estructura que se llamará **estructuraTalleres**, cuyos parámetros serán:

- **idTaller:** será el identificador del taller, único y distinto a todos los demás identificadores, representado con un entero (int).
- **cochesTaller:** será un vector que almacenará todos los coches del taller, que serán **20** coches por taller. Será del tipo **estructuraCoche** (estructura que también debemos declarar), y que contendrá a su vez los siguientes parámetros:
  - **matricula:** vector de tipo **char**, que contendrá la matrícula de cada coche, nos sirve como identificador único. Será de longitud 8 incluyendo el carácter nulo.
  - **color:** será indicado con un código que solo consiste en números, pudiendo inicializarlo como **int**. La lista de colores será: (1-azul, 2-rojo, 3-negro, 4-blanco, 5-verde, 6-naranja, 7-amarillo, 8-morado, 9-marrón, 0-otros).
  - **peso:** **float** que indicará el peso del coche.

**Apartado B):** Declara el prototipo e implementa la siguiente función:

```
int organizacionCoches(struct estructuraTalleres *talleresSalamanca, int  
numeroTalleres, int *cochesColor, char *matricula, int *taller, int  
*cochesBlancos, float *pesosTaller12Orden, float *cochePesado);
```

### Explicación de los parámetros:

- **talleresSalamanca:** vector de tipo struct estructuraTalleres de entrada que contiene los distintos talleres en salamanca.
- **numeroTalleres:** dimensión del vector anterior.
- **cochesColor:** parámetro de salida que es un vector de enteros que devolverá en sus celdas el número de coches correspondientes a cada color, por lo que tendrá 10 celdas.
- **matricula:** parámetro de entrada que nos proporciona una matrícula cualquiera.
- **taller:** parámetro de salida que nos devolverá el idTaller en el cual se encuentre la matrícula antes mencionada.
- **cochesBlancos:** parámetro de salida entero que devolverá el número de coches blancos que hay en todos los talleres.
- **pesosTaller12Orden:** vector de float que devolverá en los pesos de los coches que hay en el taller con idTaller = 12, ordenados de forma ascendente.
- **cochePesado:** parámetro de salida, float que devuelve el peso del coche más pesado que haya en cualquiera de los talleres de salamanca.

Tener una laptop gamer no significa vender un riñón. MSI tiene opciones que no te hacen llorar al ver la cuenta bancaria. Básicamente, el "starter pack" para los que quieren entrar al mundo de los juegos de PC sin que su cartera sufra un infarto.



**WUOLAH**

Comprobaciones previas:

1. Si el vector **talleresSalamanca** apunta a NULL la función finaliza y devuelve un **-1** por su valor de retorno.
2. Si **cochesColor**, o **pesosTaller12Orden** apuntan a NULL, la función finaliza y devuelve un **-2** por su valor de retorno.
3. Si la dimensión del vector **talleresSalamanca** es menor o igual que 0 la función finaliza y devuelve un **-3** por su valor de retorno.

Si todo va bien, al final de la función se devolverá un **0** por el valor de retorno.

Tareas de la función:

1. El vector **cochesColor** devolverá en la celda correspondiente a su color: (1-azul, 2-rojo, 3-negro, 4-blanco, 5 verde, 6-naranja, 7-amarillo, 8-morado, 9-marrón, 0-otros), el número de coches existentes en todo el vector **talleresSalamanca** con su respectivo color.
2. El parámetro **taller** devolverá por referencia el ID del taller en el cual se encuentre el coche que coincida con la matrícula que recibimos en el parámetro de entrada **matricula**. Si la matrícula no coincide con ningún coche, se devolverá el valor 0 por la variable **taller**.
3. El vector **pesosTaller12Orden** devolverá en sus celdas los pesos de cada coche del taller número 12 ([11] celda 11), ordenados de forma ascendente.
4. El parámetro **cochePesado**, devolverá por referencia el peso del coche más pesado de todos los talleres de Salamanca.

**Apartado C):**

Te pedirán realizar una llamada (invocación) a la función con los parámetros que ellos te digan.