

Apellido, Nombre:

Padrón:

Fundamentos de Programación

Segundo recuperatorio - 01/07/2025

Condición de aprobación: Se debe tener al menos 1 (un) ejercicio con Bien en cada parte (teórica y práctica) y adicionalmente otro ejercicio con Regular o más.

Aclaración: Un ejercicio práctico se considera **Bien** cuando además de cumplir con lo pedido, aplica las buenas prácticas profesadas por la cátedra.

Parte teórica

Ejercicio 1

- Elija un método de ordenamiento de los vistos en clase y explique su algoritmo con sus palabras, de manera tal que una persona sin conocimientos de programación lo pueda entender. No explicar en forma de código ni usar ejemplos.
- Ejemplificar el método anteriormente elegido para ordenar de manera descendente el arreglo `['b', 'i', 'a', 'n', 'z', 'c', 'o', 'p']`.

Ejercicio 2

- ¿En qué influye el tipo de un puntero a la hora de realizar operaciones aritméticas con el mismo? ¿Cómo se podría usar la aritmética de punteros para acceder a una posición de un vector?
- Este programa tiene una expresión faltante (marcada con `...`). **Sin cambiar los valores de las edades inicializadas**, ¿Qué expresión agregaría en esa parte para que el programa imprima 42? Explique por qué funciona su respuesta.

```
int main() {
    int edades[4] = {23, 20, 42, -1};

    int* p = ...;

    int total = 0;
    int i = 0;

    while(p[i] != -1){
        total += p[i];
        i++;
    }

    printf("%i", total/i);

    return 0;
}
```

Ejercicio 1

Apu tiene la heladera llena de productos vencidos, los cuales no piensa tirar, sino que algunos los va a guardar en una heladera aparte y ponerlos de oferta a un **50% menos**. Además, la última pesca tuvo antecedentes de peces con radioactividad, así que aunque no estén vencidos los quiere poner aparte.

En definitiva, los productos que tiene que poner en oferta son aquellos que:

- Estén vencidos o sean "pescado"
- **y NO** estén en los bordes de la heladera, porque están congelados y no los puede sacar.

Se pide:

Dada una matriz que representa los productos en la heladera, hacer un procedimiento que guarde en un vector aquellos productos que tiene que poner en una heladera aparte, con su **precio actualizado**.

```
typedef struct producto {
    char nombre[MAX_NOMBRE]; // "pescado", "yogurt", "crema", etc.
    int precio;
    bool vencido;
} producto_t;
```

Ejercicio 2

El Sr Burns tiene una gran cantidad de autos importados de colección, pero nunca son suficientes, por lo que decidió comprar otro auto de colección: el más antiguo y que esté en buen estado. Además, no quiere que sea Audi porque no le gustan.

```
typedef struct auto{
    char marca[MAX_NOMBRE]; // "Audi", "Ferrari", etc
    int antiguedad; // en años
    int estado;
} auto_t;
```

Se pide:

Dado un arreglo de autos, hacer una función **recursiva** que devuelva el auto más antiguo que no sea Audi y que esté en buen estado. Se sabe que un auto está en buen estado cuando la puntuación del estado sea mayor o igual a 7.