Fundamendez - 2C 2025 - Práctica semana 8

Repaso para el parcial

—

Ejercicio 1 (Pre-labo)

Moe Szyslak, el dueño de la taberna de Springfield, quiere organizar sus botellas de vino para facilitar la venta, puede hacerlo por antigüedad o alfabéticamente. La primera tiene la ventaja de que Moe sabe que sus clientes nunca pagarían una botella cara, así que las menos añejadas quedan más cerca. La segunda tiene la ventaja de que si le piden una botella específica es más fácil encontrarla. Cada botella está representada por el siguiente struct:

```
typedef struct vino {
    char nombre[MAX_NOMBRE];
    int edad;
} vino_t;
```

Dado un vector de vino_t, se pide:

a) Implementar una función que ordene las botellas utilizando el **método de ordenamiento por inserción**, de forma que las más nuevas queden **al principio del vector** y las más viejas al final.
b) Al vector ya ordenado por edad agregarle el orden **alfabético por nombre**.

Ejercicio 2

Los niños de la escuela primaria de Springfield están jugando al juego de la pelota que quema. Se ponen en ronda y se pasan la pelota entre ellos lo más rápido posible sin repetirse.

```
typedef struct ninio {
    char nombre[MAX_NOMBRE];
    int proximo_ninio;
    int fuerza;
} ninio_t;
```

Dado un vector de ninio_t crear una función recursiva que simule la ronda de pases de pelotas y devuelva la posición en el vector del niño que la pasó más fuerte.

Aclaraciones

- El último niño de la ronda tendrá un -1 en el campo `proximo_ninio`.
- La ronda siempre arranca con el primer elemento del vector.

- Puede ser que alguien en el vector nunca reciba la pelota.

Ejercicio 3

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 4

El Gato es un ladrón que está aterrorizando a todo Springfield. La policia de Springfield está haciendo un análisis de las casas robadas hasta el momento para determinar en qué barrio atacará El Gato próximamente.

Se sabe que su modus operandi es romper la cerradura de una casa y robarle algo importante a la familia. Siempre trata de llevarse un elemento por cada miembro del grupo familiar, y cuando no puede, se roba la mascota.

Se pide:

Dada una matriz de casas que fueron robadas, y sabiendo que cada columna es un barrio, realizar una función que determine qué barrio (representado por su índice) aún no tuvo un robo por El Gato.

Una casa se representa con el siguiente struct:

```c
typedef struct casa_robada {
    char elementos_robados[MAX_NOMBRE][MAX_ELEMENTOS];
    // "mascota", "saxofón", "horno", etc.
    int cant_elementos_robados;
    bool cerradura_forzada;
    char color_fachada[MAX_COLOR];
    int cant_familiares;
} casa_robada_t;
```