

Ejercicios de parcial (prácticos)

—

Ejercicio 1

La señora Krabappel está revisando los cuadernos de comunicados para asegurarse que todos los alumnos tengan el permiso firmado por mínimo dos adultos responsables para la excursión a la ****fábrica de cajas****.

Teniendo el struct `cuaderno_t`:

```
typedef struct comunicado{
    char firmas[MAX_LETRAS][MAX_FIRMAS];
    int cantidad_firmas;
    char excursion[MAX_NOMBRE]; // "museo", "fabrica de cajas", "circo", etc.
} comunicado_t;

typedef struct cuaderno{
    comunicado_t comunicados[MAX_COMUNICADOS];
    int cantidad_comunicados;
} cuaderno_t;
```

Se pide:

Crear una función que recorra una matriz no cuadrada de cuadernos que representa al autobús y devuelva el índice de la columna en la que hay la menor cantidad de alumnos que tengan el comunicado de la excursión a la **fábrica de cajas** con por lo menos dos firmas.

Ejercicio 2

En la casa de ancianos de Springfield están haciendo un ejercicio para ejercitar la memoria. Consiste en que un anciano dice una palabra y el siguiente tiene que decir la palabra que dijo el anterior más una nueva, formando una frase. Así sucesivamente, hasta que alguno se equivoque (que la frase que diga no coincida con la que se va formando).

Un turno se representa de la siguiente forma:

```
typedef struct turno{
    char nombre_anciano[MAX_NOMBRE];
    char palabra[MAX_PALABRA];
    char frase[MAX_FRASE];
} turno_t;
```

Crear un procedimiento recursivo que, dado un vector de turnos, devuelva el nombre del anciano que se equivocó o un string vacío en caso de que ninguno pierda.

Ejercicio 3

Rayo Mc Queen se prepara para la próxima carrera en un nuevo campeonato. Como no conoce a todos sus contrincantes quiere saber quien puede ser un posible gran rival.

Si un auto puede representarse con el siguiente registro:

```
typedef struct auto{
    int cantidad_victorias;
    char color[MAX_COLOR];
    bool tiene_neumaticos_nuevos;
} auto_t;
```

Un auto es un posible gran rival si:

- Es rojo (el Rayo quiere ser el único rojo que destaque).
- Tiene más de 5 victorias.
- Tiene neumáticos nuevos.

Se pide:

Crear una función recursiva que devuelva la cantidad de autos que sean posibles grandes rivales.

Ejercicio 4

En la guarida de Izma hay un cuarto lleno de trampas. Las trampas están dispuestas en un formato de grilla y cada una hace una cantidad distinta de daño. En la entrada al cuarto hay un mapa.

Este mapa está representado como una matriz de enteros donde cada posición indica cuánto daño hace la trampa en esa posición de la grilla.

Viendo el mapa Kuzco se encuentra en la parte de arriba y tiene que llegar hasta la parte de abajo. Solo puede atravesar el cuarto en línea recta pasando por alguna de las columnas.

Se pide:

Hacer una función que reciba una matriz de enteros (el mapa) y la vida actual de Kuzco y devuelva `true` si puede pasar por alguna de las columnas sin morir o `false` si no puede pasar por ninguna.

Ejemplo

Por ejemplo si el mapa fuera:

```
5 | 2 | 1 | 1
2 | 1 | 3 | 2
3 | 10 | 8 | 4
8 | 3 | 1 | 2
```

Si Kuzco tuviera 5 de vida la función debería devolver `false` porque cualquiera de los caminos lo mataría.

Si Kuzco tuviera 10 de vida entonces la función debería devolver `true` porque hay una de las columnas que no lo mata (la columna de la derecha).

Si Kuzco tuviera 100 de vida podría pasar por cualquier columna así que también se devolvería `true`.

Ejercicio 5

Hay notas que a Lisa todavía le cuestan tocar, y quiere detectar cuánto tiempo duran esas notas en una partitura.

Dado un vector de `nota_t` que representa una partitura se pide crear una **función recursiva** que devuelva el promedio de la duración (en milisegundos) de las notas "RE" que cumplan:

- Sean corcheas, no sean silencios y tengan una intensidad mayor a 5.
- o que la nota anterior sea una nota "LA" semicorchea.

Una `nota_t` se representa con el siguiente struct:

```
typedef struct nota{
    int cantidad_victorias;
    char sonido[MAX_SONIDO]; // "LA", "SI", "DO", "RE", "MI", "FA", "SOL".
    char figura_musical; // (R)edonda, (C)orchea, (S)emicorchea
    bool es_silencio;
    int duracion_ms;
    int intensidad; // del 1 al 10
} nota_t;
```