

EJERCICIO 1

Muriel (la humana de *Coraje*) fue secuestrada por los *Topos Rabiosos*. Los *Topos Rabiosos* escaparon con *Muriel* a upa, haciendo agujeros en el terreno (como suelen hacer los topos). Para despistar un poco a *Coraje*, los topos salen a la superficie en distintas posiciones del terreno. Sin embargo, *Coraje* dedujo que en cada pozo por el que pasan, los *Topos Rabiosos* dejan las **coordenadas de la próxima salida** a la superficie. El terreno puede verse como una **matriz de pozos**:

```
typedef struct coordenada {
    int fil;
    int col;
} coordenada_t;

typedef struct pozo {
    int profundidad;
    coordenada_t proximo_pozo;
} pozo_t;
```

Variante 1

Sabiendo que los *Topos Rabiosos* entraron al terreno por el pozo ubicado en la posición **(0,0)** y que *Coraje* encontrará a los *Topos Rabiosos* (y con esto rescatará a Muriel) cuando **el pozo en el que está tenga como coordenadas siguientes a si mismo**.

1. Crear una función que devuelva **cuántos movimientos** tuvo que hacer *Coraje* para salvar a *Muriel*.
2. Explique qué son y para qué sirven las buenas prácticas. ¿En qué partes del punto 1 las utiliza?

Variante 2

Coraje tiene un palpito y es que los *Topos Rabiosos* entraron al terreno por la posición **(0,0)**. Sabiendo que entre pozo y pozo pasa **5 minutos debajo de la tierra**, que **no puede estar más de 60 minutos bajo tierra por día** (es que sufre mucho en la oscuridad) y encontrará a los topos **cuando el pozo en el que está tenga como coordenadas siguientes a si mismo**.

1. Crear una función que devuelva **true** si rescata a *Muriel* o **false** si no.
2. Explique qué son y para qué sirven las buenas prácticas. ¿En qué partes del punto 1 las utiliza?

Variante 3

Coraje tiene un palpito y es que los *Topos Rabiosos* entraron al terreno por la posición **(0,0)**. Sabiendo que entre pozo y pozo corre **100 metros debajo de la tierra**, que **no puede correr más de 2 kilómetros por día** (no es un perro atlético) y encontrará a los topos **cuando el pozo en el que está tenga como coordenadas siguientes a si mismo**.

1. Crear una función que devuelva **true** si rescata a *Muriel* o **false** si no.
2. Explique qué son y para qué sirven las buenas prácticas. ¿En qué partes del punto 1 las utiliza?

Variante 4

Sabiendo que entre pozo y pozo pasa **corre 50 metros debajo de la tierra**, que los *Topos Rabiosos* entraron al terreno por el pozo ubicado en la posición **(0,0)** y que *Coraje* encontrará a los *Topos Rabiosos* (y con esto rescatará a Muriel) cuando **el pozo en el que está tenga como coordenadas siguientes a si mismo**.

1. Crear una función que devuelva **cuántos metros** tuvo que correr *Coraje* para salvar a *Muriel*.
2. Explique qué son y para qué sirven las buenas prácticas. ¿En qué partes del punto 1 las utiliza?

Variante 5

Coraje tiene un palpito y es que los *Topos Rabiosos* entraron al terreno por la posición **(0,0)**, por lo que tomo sus bengalas y acudió a su rescate. Sabiendo que entre pozo y pozo usa **3 bengalas debajo de la tierra**, que **no puede estar en un pozo sin iluminación** (es que sufre mucho en la oscuridad) y encontrará a los topos **cuando el pozo en el que está tenga como coordenadas siguientes a si mismo**.

1. Crear una función que reciba la cantidad de bengalas totales y devuelva **true** si rescata a *Muriel* o **false** si no.
2. Explique qué son y para qué sirven las buenas prácticas. ¿En qué partes del punto 1 las utiliza?

EJERCICIO 2

Es el día de las mascotas en el colegio. *Mandy* llevará a su perro *Saliva*, y *Billy* a *Puro Hueso* en reemplazo de su hamster *Señor Cariñoso* que se le duele un poco la panza. Cuando a *Puro Hueso* le llegó el momento de brillar en la clase, *Saliva* le robó su pierna huesuda provocando un destello de risas en los compañeros de *Billy* y *Mandy*. Con una simple ojeada, *Puro Hueso* pudo recordar a todos los niños y su distribución en el aula. Al ser humillado por la risa de ellos, decidió incluirlos en su lista de cosecha de almas. Se tiene una **matriz de niños** y su estructura es como la siguiente:

```
typedef struct ninio {
    char nombre[MAX_NOMBRE];
    int edad;
    long dni;
    bool se_rio;
    bool es_amigo_billy;
} ninio_t;
```

Variante 1

1. Crear una función o procedimiento que cargue en un vector de niños, todos aquellos que se rieron de *Puro Hueso* y **su nombre no tiene más de 4 letras r** (se le complica la pronunciación).
2. Crear un procedimiento que ordene el vector de niños por **edad de forma descendente**.

Variante 2

1. Crear una función o procedimiento que cargue en un vector de niños, todos aquellos que se rieron de *Puro Hueso* y no son amigos de *Billy*.
2. Crear un procedimiento que ordene el vector de niños por **nombre de forma ascendente**.

Variante 3

Suponiendo que la matriz está ordenada por nombre (por fila y columna). 1. Crear una función o procedimiento que cargue en un vector de niños, todos aquellos que se rieron de *Puro Hueso* y no son amigos de *Billy*. 2. Crear una función que determine si en el vector creado se encuentra *Mandy*.

Variante 4

1. Crear una función o procedimiento que cargue en un vector de niños, todos aquellos que se rieron de *Puro Hueso*.
2. Crear un procedimiento que ordene el vector de niños por **dni de forma ascendente**.

Variante 5

1. Crear una función o procedimiento que cargue en un vector de niños, todos aquellos que se rieron de *Puro Hueso*.
2. Crear un procedimiento que ordene el vector de niños por **edad de forma ascendente**. Si hay mas de un niño de la misma edad, se subordinaran entre ellos **por nombre de manera descendente**.

EJERCICIO 3

Un día como cualquier otro, *Bombón*, *Burbuja* y *Bellota* organizaron su colección de juguetes. Decidieron que algunos estaban un poco viejos y gastados, por lo que quieren separarlos del resto para restaurarlos y luego regalarlos. Sin embargo, recibieron una llamada urgente del *Alcalde de Saltadilla*, por lo que delegaron esta tarea al *Profesor Utonio*.

Los juguetes están representados mediante la siguiente estructura:

```
typedef struct juguete {
    char nombre[MAX_NOMBRE];
    char duenia[MAX_DUENIO];
    int antiguedad;
} juguete_t;
```

Teniendo en cuenta que se almacenan en un vector ordenado por **nombre de juguete**:

Variante 1

1. Explicar cómo podría hacer el *Profesor Utonio* para separar **un** juguete antiguo del resto. (No escribir código)
2. *Las Chicas Superpoderosas* recibieron nuevos juguetes como obsequio por su ardua tarea de salvar al mundo, pero como reciben tantos no están seguras si cada uno de estos nuevos juguetes ya lo han tenido en el pasado. Si las chicas quieren saber si un juguete nuevo ya estuvo en su colección en otro momento. ¿Seguirías recomendando la solución del ítem anterior? ¿Qué modificarías? ¿Por qué? (No escribir código)

Variante 2

1. Explicar cómo podría hacer el *Profesor Utonio* para buscar **el juguete más antiguo**. (No escribir código)
2. *Las Chicas Superpoderosas* recibieron nuevos juguetes como obsequio por su ardua tarea de salvar al mundo y fueron agregados al final, por lo que el vector ahora, no está más ordenado por nombre. Una niña, se dió cuenta que por equivocación les dio a su muñeca favorita *Minnie*, y quiere recuperarla. ¿Seguirías recomendando la solución del ítem anterior para buscar esta muñeca? ¿Qué modificarías? ¿Por qué? (No escribir código)

Variante 3

1. Explicar cómo podría hacer el *Profesor Utonio* para insertar un juguete que encontró en la habitación, y las *chicas superpoderosas* no tuvieron en cuenta, **sin alterar el orden por nombre**. (No escribir código)
2. *Las Chicas Superpoderosas* recibieron nuevos juguetes como obsequio por su ardua tarea de salvar al mundo y fueron agregados al final, por lo que el vector ahora, no está más ordenado por nombre. Una niña muy fan de ellas, no pudo ir a verlas a la ciudad, por lo que las esperó en la casa para darles un juguete más. ¿Seguirías recomendando la solución del ítem anterior para agregar este nuevo juguete? ¿Qué modificarías? ¿Por qué? (No escribir código)

EJERCICIO 4

Ed, Eddy y Eddy caminaban por el vecindario y quedaron fascinados al ver como a un adolescente le quedaba un piercing en la ceja.

Decidieron que iban a hacerse uno cada uno, pero antes, tenían que probarlo en alguien que no sienta dolor... *Tablón*.

Como toda actividad en el barrio, por insignificante que parezca se convierte en un gran evento, decidieron que el clavo a elegir saldría de una competencia y que el dueño del clavo elegido será coronado como *Rey del Barrio* por toda la semana.

Ed, Eddy y Eddy corrieron a juntar clavos y cada uno de ellos creó su propio vector de clavos:

```
typedef struct clavo {
    int largo;
    int dobladez;
    bool oxidado;
} clavo_t;
```

El ganador de la competencia se determinará de la siguiente manera:

Variante 1

- *Ed* decidió elegir entre los clavos que se encuentran en **posiciones pares** de su vector.
 - *Edd* decidió elegir entre los clavos que se encuentran en **posiciones impares** de su vector.
 - *Eddy* decidió elegir entre los clavos que se encuentran en **cualquier posición** de su vector.
 - Cada uno de ellos, elegirá el clavo **más largo**.
 - De esos 3 clavos, el elegido para usar como piercing de *Tablón* será el menos doblado.
1. Crear una función que muestre por pantalla el nombre del *Rey del Barrio* y la posición del clavo ganador.

Variante 2

- *Ed* decidió que solo elegirá clavos cuyo **largo sea múltiplo de 2**.
 - *Edd* decidió que solo elegirá clavos cuyo **largo sea múltiplo de 3**.
 - *Eddy* decidió que solo elegirá clavos cuyo **largo sea múltiplo de 4**.
 - Cada uno de ellos, elegirá el clavo **más corto** y que **no esté oxidado**.
 - De esos 3 clavos, el elegido para usar como piercing de *Tablón* será el más doblado.
1. Crear una función que muestre por pantalla el nombre del *Rey del Barrio* y la posición del clavo ganador.

Variante 3

- Los 3 decidieron que elegirán sus clavos empezando **desde atrás** de sus vectores.
 - Cada uno de ellos, elegirá el clavo con la suma de características (**largo y dobladez**) sea mayor.
 - De esos 3 clavos, el elegido para usar como piercing de *Tablón* será el más doblado.
1. Crear una función que muestre por pantalla el nombre del *Rey del Barrio* y la posición del clavo ganador.

EJERCICIO 5

Variante 1

Muchos niños tienen amigos imaginarios pero los abandonan al crecer, por eso existe el *Hogar de Foster*, una mansión donde se hospedan los amigos imaginarios de toda la infancia que son abandonados por sus creadores al crecer y pueden vivir hasta ser adoptados por otro niño.

Mac, un niño de 8 años de gran inteligencia y corazón, y su amigo imaginario *Blooregard Q. Kazoo* (o simplemente *Bloo*) viven sus aventuras allí, ya que no quieren separarse aún.

1. Se pide crear una estructura para representar a la *Mansión Foster para Amigos Imaginarios*.

- Pista 1: No limitarse solo a la mansión, tener en cuenta que allí viven y vivieron amigos imaginarios, que tienen características especiales, tuvieron un creador y uno o más dueños... Usar, justamente, la imaginación.
- Pista 2: Les dejamos aquí una imagen de los actuales habitantes de la mansión :)



Variante 2

Mojo Jojo es un chimpancé científico de mente notable, cuyo objetivo principal es destruir a *Las Chicas Superpoderosas*, aplastar *Saltadilla* y conquistar el mundo, en ese orden. Para hacer esto, pasa gran parte de su tiempo tramando e inventando nuevas armas o tecnología.

Esto normalmente viene en forma de robots gigantes o grandes armas de energía/láser.

Todas sus ideas, las vuelve en un cuadernillo marca *AVON*, porque dice que traen paisajes hermosos.

1. Se pide crear una estructura para representar el cuadernillo de *Mojo Jojo*.

- Pista 1: No limitarse solo al cuadernillo (que tiene hojas, se gasta, etc), tener en cuenta que allí escribe muchas ideas, cada idea generalmente es destructiva, puede tener niveles de maldad, además *Mojo Jojo* vive en una montaña el invento puede o no estar ahí... Usar la imaginación!
- Pista 2: Les dejamos aquí una imagen de *Mojo Jojo* y una vez que creó un ejército de *Mojo Jojitos* :)



Variante 3

El *Laboratorio de Dexter* tiene muchísimos artefactos tecnológicos compuestos por botones, scanners, engranajes, cables, leds, láseres y/o pantallas. Pero un óptimo funcionamiento conlleva un óptimo mantenimiento, el cual delegó a sus tantos robots automatizados.

Estos deben ser meticulosamente programados para que puedan coexistir con su familia, ya que *Dee Dee* suele visitar su laboratorio para *jugar* (molestar según él), su mamá a limpiar y su papá a avisarle de que comenzó su programa favorito de tv.

1. Se pide crear una estructura para representar a las personas que merodean el *Laboratorio de Dexter*.

- Pista 1: No limitarse solo a los que habitan en la casa misma casa de él, tener en cuenta que suele recibir visita de alguna tía, primo, abuela o familiar de otra línea temporal, enemigos, amigos, animales...
- Pista 2: Les dejamos aquí una imagen de algunos de los familiares de Dexter :)



Variante 4

El *Laboratorio de Dexter* tiene muchísimos artefactos tecnológicos compuestos por botones, scanners, engranajes, cables, leds, láseres y/o pantallas. Pero un óptimo funcionamiento conlleva un óptimo mantenimiento, el cual delegó a sus tantos robots automatizados.

Estos robots también están fabricados por distintos componentes electrónicos para lograr su o sus objetivos. Suelen identificarse con algún código, algún color de chasis o con algún apodo si logra generar cariño en su creador. Algunos son tan *guauuuuuuuu!!* que hasta tienen módulo de voz y procesamiento de habla.

Dexter deja volar su creatividad cada vez más y ahora, es tu turno.

1. Se pide crear una estructura para representar a los robots creados por Dexter.

- Pista 1: No limitarse solo a las características antes mencionada, tener en cuenta que él tiene una vida fuera del laboratorio por lo que posee robots extralaboratoriales que ayudan en su vida diaria.
- Pista 2: Les dejamos aquí una imagen de algunas de sus mejores creaciones :)

