

Apellido, Nombre:

Padrón:

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

Parte teórica

Ejercicio 1

Kuzco está aprendiendo a programar y le está costando entender uno de los métodos de ordenamiento, necesita nuestra ayuda.

1. Explicar con tus palabras el algoritmo de **selección** para ordenar un vector.
2. Ejemplificar ordenando este vector de enteros `[3, 5, 2, 8, 0, 7]` de manera ascendente.

Ejercicio 2

Kuzco le trató de explicar un poco de manejo de punteros a Pacha. Parece que algunas de las cosas que le dijo eran un poco inchequeables así que nos pidió ayuda para verificarlas.

Suponiendo que el vector del punto anterior está guardado dentro de una variable llamada `precios`. Indique si las siguientes afirmaciones son verdaderas o falsas, justificando en ambos casos:

- a. Hacer `*precios` debería dar un error porque estoy accediendo a la posición de un vector y no un entero.
- b. `*(precios+1)` es lo mismo que `precios[0]+1`.
- c. Si quiero reservar memoria para guardar todo ese vector en memoria dinámica tengo que hacer `malloc(sizeof(int))` ya que es un vector de enteros.

Parte práctica

Ejercicio 1

Mientras Kuzco estaba distraído aprendiendo a programar, Izma estaba ideando un plan malvado. Para que el plan fuera exitoso, necesitaba tener una idea de la potencia de sus pociones más fuertes. Una poción está representada con el struct `pocion_t`.

```
typedef struct pocion{
    char nombre[MAX_NOMBRE];
    int potencia;
    bool de_rapida_accion;
```

```
} pocion_t
```

Para este plan a Izma solo le interesan las pociones fuertes. Para que una poción cuente como fuerte tiene que tener **potencia mayor a 5** y tiene que **ser de rápida acción**. Izma necesita saber la potencia promedio de las pociones fuertes

Se pide

Dado un vector de `pocion_t` que contiene **todas las pociones** y su tope hacer una función que devuelva la potencia promedio pero que solo cuente a las pociones fuertes.

Ejercicio 2

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 cuanto 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 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`.