

"Porque uno es esclavo de sus hazañas pero dueño de sus intentos...." – K. J. –

Apellido, Nombre: ..... Padron: .....

Mail: ..... Entrego ..... hojas.

1	2	3	4	5	Nota

Antes de empezar a resolver el examen lea las siguientes aclaraciones:

- Complete sus datos en esta hoja. Firme, numere e inicialice con nombre, apellido y padrón todas sus hojas.
- Léalo **todo** a conciencia, y haga preguntas sobre lo que no entiende en el espacio designado para ello.
- La parte práctica de los ejercicios debe ser realizada en el lenguaje C, respetando las buenas prácticas de programación.
- Recomendamos fuertemente realizar un análisis de **cada** ejercicio.
- Para aprobar es necesario tener bien, al menos, el 60% de todo el examen.
- Los ejercicios 1 y 2 no pueden estar mal.
- Cada ejercicio vale 2 puntos.

## Ejercicios

1. La **Oficina de Aurores** es una oficina en el **Departamento de Operaciones Mágicas Especiales**, en el segundo piso del **Ministerio de Magia**. Un aurores es un empleado del **Ministerio** cuya misión es perseguir y detener magos tenebrosos. De acuerdo con **Minerva McGonagall** (profesora de **Hogwarts**), la **Oficina de Aurores** solo admite nuevos reclutas con 2 **EXTASIS** (Exámenes Terribles de Alta Sabiduría e Invocaciones Secretas) aprobados y cree que los **EXTASIS** más apropiados para alguien que aspira a ser recluta son los de **Transformaciones** y **Defensa Contra las Artes Oscuras**, y que sus puntajes deben ser entre 75 y 100 en cada examen. Se tienen dos vectores de alumnos con registros del tipo **alumno\_t**, uno de los que aprobaron **Transformaciones** y otro de los que aprobaron **Defensa Contra las Artes Oscuras** (ambos con nota entre 60 y 100 puntos, ordenados ascendentemente por nombre).

```
typedef struct alumno {
    char nombre[MAX_NOMBRE];
    int nota;
} alumno_t;
```

- a. Se necesita crear un tercer vector con los alumnos que **NO** pueden ser reclutados por la **Oficina de Aurores**, según los requisitos de **Minerva McGonagall**. En este tercer vector en el campo **nota** debe ir el promedio de las notas que tiene el alumno en **Transformaciones** y **Defensa Contra las Artes Oscuras**.
  - b. Explique uno de los métodos de ordenamiento que conoce. ¿Cuál es el mejor?
2. El tren que parte de la plataforma  $9^{3/4}$  lleva los mejores dulces de todos. El carrito que los contiene está dividido en cuadrados en donde están los registros de los caramelos: nombre del mismo, la acidez y el dulzor en porcentaje.

```
typedef struct dulce_magico {
    char nombre_dulce[MAX_NOMBRE];
    int acidez;
    int dulzor;
} dulce_magico_t;
```

Ron adora los dulces mágicos, pero esta vez desea encontrar uno en particular, la **Rana de Chocolate**. El vector está ordenado descendientemente por nombre del dulce y no hay nombres repetidos.

- a. Pide una función que reciba como parámetro un vector de dulces y el tope, y devuelva la posición en donde se encuentra el caramelo tan deseado por Ron o -1 si no está.

3. **Hogwarts** es un castillo muy muy muy muy muy grande con muchas habitaciones para que todos los estudiantes de magia vivan ahí durante su cursada. El castillo tiene 10 pisos y en cada piso hay 20 habitaciones. **Argus Filch**, el celador, tiene un mapa cargado al sistema, en forma de matriz, donde cada celda es una habitación que tiene la siguiente información:

```
typedef struct habitacion{  
    int numero_habitacion;  
    char nombre_alumno[MAX_ALUMNO];  
    bool necesita_limpieza;  
    int hora;  
} habitacion_t;
```

Un duendecillo de **Cornualles**, ser de color azul y muy revoltoso, entra todos los días al sistema a fijarse si ese día son más de la mitad las habitaciones que necesitan limpieza entre las 7 y las 10 (horario que él tiene disponible para hacer sus travesuras) así puede entrar a ellas y dejar pequeños artefactos escondidos para sus bromas pesadas, solo entrará a las habitaciones de un piso, si en todas ellas puede entrar, en caso de que haya una a la que no puede, no entrará a ninguna de ese piso.

- a. El duendecillo necesita que nosotros creamos un algoritmo que devuelva la cantidad de pisos a los que puede entrar, cumpliendo con la siguiente firma:

```
int cantidad_pisos(habitacion_t hogwartz[ALTO][ANCHO]);
```

4. **Harry** está en el **Callejón Diagon** buscando una lechuza para regalarle a **Ron** ya que hace tiempo no puede comunicarse con sus padres. Las lechuzas, para los magos, sirven más para llevar y traer paquetes que como mascotas.
- a. Cree un registro **lechuza\_t** con los campos que crea necesarios para representar una lechuza.
5. ¿Cuáles son los componentes principales de la recursividad? Grafique un ejemplo donde se muestre como es el flujo de un algoritmo recursivo. ¿Qué ejemplos conoce?