

"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

- 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;
```

- Se necesita crear un tercer vector con los alumnos que 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**.
 - Explique uno de los métodos de ordenamiento que conoce y cómo ordenaría el vector descendentemente por promedio. ¿Qué ventaja tendría la **Oficina** si se ordena así?
- El tren que parte de la plataforma **93/4** 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** con un porcentaje de acidez y de dulzor, en ambos, por arriba de 20.

- a. Pide también una función que reciba como parámetro un vector de dulces y el tope, y devuelva la posición en donde se encontraba el caramelo tan deseado por Ron.

Aclaración: Existen ranas de distintos tipos de dulzor y acidez.

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, si son menos de la mitad las habitaciones a las que puede acceder en ese horario, considera que no sería lo suficientemente divertido.

- a. El duendecillo necesita que nosotros creemos un algoritmo que devuelva **true** si le conviene entrar a hacer sus travesuras ese día o **false** si no le conviene entrar, cumpliendo con la siguiente firma:

```
bool conviene_travesurear(habitacion_t hogwartz[ALTO][ANCHO]);
```

4. Los elfos domésticos son utilizados como esclavos, son poseedores de una magia distinta a la de los magos y brujas, y la utilizan para servir a sus amos (familias humanas) a los que son totalmente leales. Sirven hasta a 10 familias en toda su vida y su estatura no supera el metro. Estos elfos pueden ser liberados cuando su amo les da una prenda de vestir.
 - a. Crear una estructura que permita almacenar la información para representar a un elfo doméstico.
5. ¿Qué es la recursividad? ¿Cuáles son los componentes principales? Grafique un ejemplo donde se muestre como es el flujo de un algoritmo recursivo.