Apellido, Nombre:

.....

Mail:

.....

Padrón:

.....

Teórico			
1:	2:	3:	Nota: .....

Práctico			
1:	2:	3:	Nota: .....

Nota Final
.....

Aclaraciones:

- **Antes de comenzar a resolver el final, complete sus datos en esta hoja, y al finalizarlo, firme todas las hojas.**
- **Los ejercicios deben ser implementados en el lenguaje de programación C, respetando las buenas prácticas de programación.**
- **Para cada ejercicio práctico se recomienda fuertemente realizar un análisis y un diagrama del problema y la solución.**
- **Se deben numerar TODAS las hojas e inicializarlas con nombre, apellido, padrón y cualquier otra información que considere necesaria.**
- **La aprobación del parcial está sujeta a la aprobación de al menos el 60 % del mismo.**

Teoría

1. ¿Qué es un algoritmo?
2. Enuncie 5 buenas prácticas de programación y explíquelas con un ejemplo sencillo.
3. ¿Para qué sirve la función **fseek**? ¿En qué casos la usaría?

Práctica

1. **Dumbledore** y **Minerva** visitaron a todos y cada uno de los alumnos de **Hogwarts** para saber que materias cursarán el año que entra.
   Cada uno de ellos creó un archivo binario de acceso secuencial (**dumbledore.dat** y **minerva.dat**) ordenado alfabéticamente por alumno y cada registro contenido tiene el nombre del alumno y el id de la materia.

```c
typedef struct cursada {
    char alumno[MAX_ALUMNO];
    int id_materia;
} cursada_t;
```

   Aclaración: Cada alumno puede decidir cursar más de una materia y solo fue consultado por un profesor.

   a) Crear una función que, a partir de los dos archivos, cree un 3er archivo (**cursada.dat**). El archivo **cursada.dat** debe mantener el orden enunciado antes.

   Se cuenta además con un archivo binario de acceso directo que contiene todas las materias (**materias.dat**) que se pueden cursar en **Hogwarts**, el

mismo está ordenado por **id** que es único y correlativo (no existen huecos de id) con registros del tipo **materia_t**.

```
typedef struct materia {
    int id_materia;
    char descripcion[MAX_DESCRIPCION];
} materia_t;
```

b) Se pide generar un listado, agrupado por alumno, que liste las materias que va a cursar cada alumno. Como se muestra a continuación:

```
-------------------------------------------
Drako Malfoy
    Pociones 2
    Defensa contra las artes oscuras
-------------------------------------------
Harry Potter
    Pociones 2
    Herbologia 1
...
-------------------------------------------
```

2. **Harry** y **Ron** están jugando a las damas, dicho juego consta de un tablero de 8x8 y fichas rojas y blancas. En este caso, **Harry** usa las rojas ('R') y **Ron** las blancas ('B').

   a. Dada una matriz cuadrada y recorriéndola **una sola vez** decidir si va ganando **Harry** (devolver 'H'), **Ron** (devolver 'R') o si van empatando (devolver 'E').

3. Entrar a la *sala común de Ravenclaw* se reduce a contestar correctamente un enigma. Esta vez el enigma consiste en listar todos los números pares de un vector. Los alumnos escribieron un procedimiento que lo hace por ellos:

```
void imprimir_pares(int numeros[MAX], int tope){
    for (int i=0; i< tope; i++){
      if (numeros[i]%2 == 0)
        printf("El número %i es par.", numero[i]);
    }
}
```

Sin embargo no les funcionó para entrar, el algoritmo está bien, pero parece que tiene que ser recursivo para que se abran las puertas...

   a. Transformar el algoritmo escrito por los alumnos, para que resuelva el problema recursivamente.