

"Son nuestras decisiones las que muestran lo que podemos llegar a ser. Mucho más que nuestras propias habilidades." - A. Dumbledore -

Apellido, Nombre: Fraccaro, Agustina Padron: 1031.99

Mail: agufraccaro@hotmail.com Entrego 4 hojas.
watto

1	2	3	4	5	Nota
B	M	B	B	B	7

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. **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.
 - b. Durante el entrenamiento de lechuzas, **Ron** se encuentra con otros magos con lechuzas, el entrenador les pide que se ordenen según la velocidad de ellas. Si no lo hizo en el punto anterior, suponga que el registro `lechuza_t` tiene un campo `velocidad` y explique como ordenaría el vector de lechuzas. No es necesario escribir el código del algoritmo.

2. **Azkaban** es una fortaleza en una isla que sirve a la comunidad mágica como una prisión para criminales magos. Usando ciertos encantamientos, **Azkaban** está escondida del mundo **muggle**. Recientemente cambió la cúpula de la prisión. La nueva dirección mandó a hacer un relevamiento de la distribución de la *peligrosidad* de los magos. **Azkaban** puede verse como una matriz, donde cada fila es un piso (tiene 10) y cada elemento de cada fila, es una celda (tiene 20 celdas por piso). Cada celda tiene la siguiente información:

```
typedef struct celda {
    bool ocupada;
    int peligrosidad; // de 1 a 5 siendo 5 el más peligroso.
} celda_t;
```

Se quiere determinar, cuál es el piso más peligroso (sumando las *peligrosidades*) y si existe algún piso que tenga todas las celdas vacías.

La dirección creó la siguiente firma:

```
int piso_mas_peligroso_de_azkaban(celda_t azkaban[PISOS][CELDAS], bool *hay_piso_vacio);
```

- a. Implementar la función `piso_mas_peligroso_de_azkaban` cumpliendo con lo pedido.
 - b. ¿Cómo cambiaría su algoritmo si se agregan pisos a la prisión?
3. Un **boggart** es un no-ser mortal de forma cambiante capaz de transformarse en la imagen de lo que más teme su espectador. Se desconoce cómo es un **boggart** antes de transformarse en el miedo de quien lo enfrenta. Se lo puede localizar ya que hace sonar, sacude y/o araña los objetos cercanos al lugar en el que se esconde.

Lupin está encargado de las estadísticas este año y quiere saber cuál es el **promedio de edad** de los alumnos que tienen como mayor miedo a... **Mendzsmort**.

```
typedef struct alumno {
    char nombre[MAX_NOMBRE];
    int edad;
    int anio_que_cursa;
    char casa;
} alumno_t;

typedef struct transformacion {
    char miedo[MAX_MIEDO];
    alumno_t espectador;
} transformacion_t;
```

- a. Implementar una función que reciba un vector con las transformaciones realizadas por el **boggart** y la cantidad de transformaciones realizadas y devuelva, como retorno de la función, el promedio de edad de aquellos cuyo mayor miedo es... **Mendzsmort**.
 - b. Si **Lupin** quisiera saber el promedio de edad de los que *pertenecen a una casa en particular* y se asustan con **Mendzsmort**, ¿qué cambiaría en su algoritmo?
4. Una vuelta del **Giratiempo** retrocede el tiempo en una hora... Así es como debería funcionar, pero **Ron** abrazó a **Hermione** y el **Giratiempo** se cayó, el vidrio de astilló, y ahora funciona como quiere... **Neville**, muchacho muy inteligente, determinó que el algoritmo al que responde ahora es el dado por **horas_que_atrasa_giratiempo**, que recibe las vueltas que se le dan y devuelve las horas que atrasa.

```
int horas_que_atrasa_giratiempo (int vueltas){
    if (vueltas == 1)
        return 1;
    else
        return vueltas + horas_que_atrasa_giratiempo(vueltas - 1);
}

int main () {
    int vueltas = 6;
    int horas = horas_que_atrasa_giratiempo (vueltas);
    printf("Volverás %i horas\n", horas);
    return 0;
}
```

- a. Determinar qué imprime el programa por pantalla, es decir, ¿cuántas horas atrasa el **Giratiempo**, si se le dan 6 vueltas? Explique cómo llega a la solución con un diagrama o dibujo.
 - b. ¿Qué es la recursividad? ¿Cuáles son los componentes principales?
5. Este año, **Dolores Umbridge** pasó lista solo 2 veces y decidió que podrán rendir el examen de *Defensa Contra Las Artes Oscuras* sólo los que hayan asistido a ambas clases. Para esto, cuenta con dos listas ordenadas por nombre del alumno, y desea generar una sola, con aquellos alumnos que figuran en las dos.

```
typedef struct alumno {
    char nombre[MAX_NOMBRE];
    int edad;
    int anio_que_cursa;
    char casa;
} alumno_t;
```

- a. Crear un procedimiento que reciba 2 vectores de alumnos y sus tope (no son más de 100 alumnos) y devuelva en un 3er vector aquellos alumnos que figuren en ambas listas (y su tope).
- b. ¿Cuál es el valor máximo que puede tomar el tope del 3er vector? ¿Por qué?