

| FINAL INFORMATICA I | | <i>Duración</i> | <i>Fecha</i> | <i>Hojas</i> |
|----------------------------|--|-----------------|-------------------|--------------|
| | | 1h 45m | 10/12/2025 | |

| <i>Nombre y Apellido</i> | <i>Nº Legajo</i> | <i>Calificación</i> | | <i>Docente Evaluador</i> | |
|--------------------------|------------------|---------------------|---------------|--------------------------|--------------|
| | | <i>número</i> | <i>letras</i> | <i>Nombre</i> | <i>Firma</i> |
| | | | | | |

*Numere las hojas entregadas y complete en la primera hoja la cantidad total de hojas entregadas.
Lea detenidamente el enunciado; su correcta interpretación forma parte de la evaluación.*

Parte Teórica

- 1) Analice el siguiente programa. Pretende ser un ejemplo sencillo de aplicación de threads. Pero una vez ejecutado, el valor de la variable contador no es el esperado. Explique por qué sucede esto y cómo se puede resolver el problema para que funcione adecuadamente. **(2,00 puntos)**

```
#include <stdio.h>
#include <pthread.h>
#include <sys/stat.h>
#include <unistd.h>
#include <sys/types.h>
#include <fcntl.h>

int contador=10;

void * f_hilo1(void *sin_usar)
{
    int i;
    for(i=0; i<100000; i++)
    {
        contador++;
    }
    pthread_exit(NULL);
}

void * f_hilo2(void *)
{
    int i;
    for(i=0; i<100000; i++)
    {
        contador--;
    }
    pthread_exit(NULL);
}

int main(void)
{
    pthread_t hilo1, hilo2;

    pthread_create(&hilo1, NULL, f_hilo1, NULL);
    pthread_create(&hilo2, NULL, f_hilo2, NULL);

    pthread_join(hilo1, NULL);
    pthread_join(hilo2, NULL);

    printf("Contador = %d", contador);

    printf("\n\n");
    return 0;
}
```

Parte Práctica

Un salón de eventos dispone de 185 asientos de distintas categorías:

| Categoría | Filas | Asientos por fila | Total de asientos x categoría |
|--------------|-------------------|-------------------|-------------------------------|
| • PREMIUM | A – B – C – D – E | 10 | 50 Asientos |
| • ESTANDAR | F – G – H – I – J | 12 | 60 asientos |
| • ECONÓMICAS | K – L – M – N – O | 15 | 75 asientos |

Los precios de cada ubicación se encuentran en el archivo binario **eventos.utn** que responde a la siguiente estructura:

```
typedef struct ubicaciones
{
    char fila, categoria[15];
    int asiento;
    float precio;
}Asientos;
```

Para el acceso al salón, el público dispone de entradas que incluyen los siguientes datos:

- Código de entrada (entero)
- Fila (Carácter)
- Número de asiento (entero)

Para el control de acceso y facturación, se solicita un programa en C que lleve a cabo las siguientes acciones:

2) Escriba la función **(2,00 puntos)**

| | |
|-----------|---|
| Prototipo | int Entradas(Nodo **inicio, char fila, int asiento, char * archivo); |
| Propósito | Construir una lista simple con los datos de las localidades ocupadas |
| Recibe | inicio: dirección de comienzo de la lista |
| | fila: Carácter que identifica la fila (de A a O) |
| | asiento: entero que identifica el asiento |
| | archivo: string con el nombre del archivo |
| Devuelve | 0: lista creada con éxito |
| | 1: error de acceso a archivo |
| | 2: error de memoria |

Notas:

- La estructura de los nodos es:

```
typedef struct reg
{
    char fila, categoria[15];
    int asiento;
    float precio;
    Nodo *next;
}Nodo;
```

- El precio y categoría de entrada se debe obtener del archivo binario **eventos.utn**, a partir de la fila y número de asiento ingresados por teclado.

3) Determinar el subtotal recaudado por cada categoría de entradas. Utilice una función. Proponga Ud. el prototipo. **(1,50 puntos)**

4) Determinar el porcentaje de ocupación de cada fila. Utilice una función. Proponga Ud. el prototipo. **(1,50 puntos)**

5) Guardar en un archivo binario los nodos correspondientes a la categoría que más recaudó. **(1,50 puntos)**

6) Eliminar la lista de la memoria dinámica. **(1,50 puntos)**

IMPORTANTE!!! Para calificación máxima, debe utilizar funciones de archivos de bajo nivel