

GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL
INFORMÁTICA - CURSO 1º - 2023-2024
PRIMERA CONVOCATORIA OFICIAL

Nombre:	Apellidos:	DNI:	Grupo:
---------	------------	------	--------

NOTA ACLARATORIA: en los ejercicios en los que se pide escribir el cuerpo de una función, salvo que en el enunciado se especifique claramente que dicha función debe usar el teclado y/o la pantalla, su código no debe contener llamadas a `scanf()` o `printf()` o funciones análogas, mucho menos para pedir parámetros por teclado o devolver valores por pantalla. Cualquier uso indebido de las funciones de manejo de teclado y pantalla en este sentido será penalizado con la suspensión del ejercicio completo, independientemente de que el resto de él sea correcto.

----- PARTE DE PROGRAMACIÓN -----

Problema 1 (5 puntos): Se desean gestionar las sesiones de un cine a lo largo de un año. En este cine, se registran una serie de sesiones que son definidas por el día, la hora y los minutos en la que se emiten, la película que se proyecta y la sala en la que tienen lugar. Para ello se proponen las siguientes estructuras y variables globales que recogen toda la información:

```
struct Sala
{
    int num;                // Número de la sala (identificador).
    int asientos[100][100]; // Asientos del cine. Si un asiento está ocupado
                           // se almacena un 1 en su fila y columna correspondiente;
                           // 0 si está libre.
    int nF;                 // Número de filas.
    int nC;                 // Número de columnas.
};

struct Sesion
{
    int dia;                // Día de la sesión.
    int hora;               // Hora a la que empieza la sesión.
    int min;                // Minutos en los que empieza la sesión.
    char pelicula[50];      // Película que se emite.
    struct Sala sala;       // Sala de la sesión.
    float precio;           // Precio por entrada.
};

struct Cine
{
    int id;                 // Identificador del cine.
    struct Sesion sesiones[100]; // Sesiones registradas.
    int numSesiones;        // Número de sesiones registradas.
};

struct Cine c; // Variable con información del cine.
```


GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL
INFORMÁTICA - CURSO 1º - 2023-2024
PRIMERA CONVOCATORIA OFICIAL

Nombre:	Apellidos:	DNI:	Grupo:
---------	------------	------	--------

a) (0.75 puntos) Escriba un programa que reserve un asiento para una sesión específica. Se deberá recoger por teclado: la posición que ocupa la sesión deseada dentro del vector sesiones, y la fila y columna del asiento que se desea reservar. Si el asiento está previamente ocupado, no podrá llevarse a cabo la reserva y se deberá mostrar por pantalla “Error. Asiento reservado”.

```
int main()
{
```

b) (1,75 puntos) Escriba un programa que calcule la recaudación total de una determinada película (cuyo nombre debe recogerse por teclado) y lo muestre por pantalla. Debe tener en cuenta sólo los asientos que están ocupados, y no los que están libres.

```
int main()  
{
```

GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL
INFORMÁTICA - CURSO 1º - 2023-2024
PRIMERA CONVOCATORIA OFICIAL

Nombre:	Apellidos:	DNI:	Grupo:
---------	------------	------	--------

c) (2,5 puntos) Escriba un programa que muestre por pantalla las sesiones que tengan al menos n asientos libres, donde n es un valor entero que debe recogerse previamente por teclado, para un determinado día (también debe recogerse por teclado). La información que debe mostrarse para cada sesión son la hora, minuto y película proyectada.

La información de las sesiones debe mostrarse de la siguiente forma (ejemplos):

17:30 – Wonka

22:00 – Napoleón

...

```
int main()
{
```


GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL

INFORMÁTICA - CURSO 1º - 2023-2024

PRIMERA CONVOCATORIA OFICIAL

Nombre:

Apellidos:

DNI:

Grupo:

Problema 2 (3 puntos): Realice un programa que verifique si la matriz 'mat' cumple las siguientes condiciones:

1.- Todos los elementos de la matriz deben estar comprendidos entre 0 y 10.

2.- La media aritmética de los elementos de cada fila de la matriz debe ser mayor o igual a 5 (todas las filas deben cumplir la condición).

En el caso de que no se cumplan algunas de las condiciones anteriores, deberá establecer a 0 los elementos de la diagonal principal de la matriz (esto es, la diagonal que discurre desde la esquina superior izquierda hacia la inferior derecha).

Por ejemplo, con una matriz de 3x3:

Original:

2	9	7
12	6	5
5	2	2

→

Resultante:

0	9	7
12	0	5
5	2	0

En el ejemplo, la matriz original no cumple las condiciones pues:

- Un elemento de la matriz es 12 (mayor que 10).
- La media aritmética de la fila [5,2,2] es 3 (menor que 5)

Puesto que no cumple las condiciones (con no cumplir una de ellas es suficiente) la resultante tiene la diagonal a 0.

```
int main()
{
    int mat[20][20]={ {2,...,6},...};
```


GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL			
INFORMÁTICA - CURSO 1º - 2023-2024			
PRIMERA CONVOCATORIA OFICIAL			
Nombre:	Apellidos:	DNI:	

Problema 3 (2 puntos): Analice el siguiente código, indique el valor de las variables que se indican más abajo en cada iteración y escriba qué se muestra en la pantalla tras su ejecución.

```
int main()
{
    int a, i=0;
    int v[10] = {2,7,3,5,6,4,8,1};
    while (i<6)
    {
        if (i%2!=0 && v[i+1]<=5)
        {
            a=v[i];
            v[i]=v[i+1];
            v[i+1] = a;
            printf("[1] ");
        }
        else if (v[i]>i || i<4)
        {
            v[i+1]++;
            v[i]=v[i+1]-i;
            printf("[2] ");
        }

        if (v[i+1]-v[i]>0)
        {
            printf("[3] ");
            i++;
        }
        i++;
        printf("\n");
    }
    return 0;
}
```

· Valor de las variables en cada iteración:

Iteración	Variable i	vector v							
1ª									
2ª									
3ª									
4ª									
5ª									
6ª									

· Se muestra por pantalla:

REFERENCIA RÁPIDA DE LENGUAJE C Y SQL

Tipos básicos de datos char short int long float double	Operadores aritméticos + - * / (de menor a mayor prioridad) % (operador módulo) () (paréntesis, para agrupar operaciones) Asignación de variables variable = expresion;	Operadores lógicos bit a bit & (AND bit a bit) (OR bit a bit) ^ (XOR bit a bit) ~ (NOT bit a bit) >> (desplazamiento a la derecha) << (desplazamiento a la izquierda)
Operadores relacionales == != < > <= >=	Operadores lógicos para condiciones && (AND lógico) (OR lógico) ! (NOT lógico)	Bucle FOR for (inicializ;condicion;actualiz) { cuerpo del bucle }
Condiciones if (condicion) { bloque si condición cierta } else { bloque si condición falsa } OPCIONAL	Condicional múltiple IF-ELSE-IF if (condicion1) { bloque1 } else if (condicion2) { bloque2 } ... else { bloque por defecto }	Condicional múltiple SWITCH-CASE switch (expresion) { case CONSTANTE1: ... break; case CONSTANTE2: ... break; default: ... }
Bucle WHILE while (condicion) { cuerpo del bucle }	Vectores (las cadenas son vectores de tipo char) Tipo nombre[DIMENSION]; Índices válidos: 0 a DIMENSION-1	Matrices Tipo nombre[FILAS][COLUMNAS]; Índices válidos: 0 a FILAS-1, 0 a COLUMNAS-1
Funciones Prototipo (antes de main()): tipo_devuelto nombre (tipo1 arg1, tipo2 arg2, , tipoN argN); Implementación (después de main()): tipo_devuelto nombre (tipo1 arg1, tipo2 arg2, , tipoN argN) { declaración de variables... bloque de código... return valor_a_devolver (si tipo_devuelto no es void) } Llamada a una función: funcion(val1, val2,...); (si funcion es void) variable = funcion(val1, val2,...); (si funcion no es void)		Constantes #define NOMBRE_CTE valor (nose termina en ;) Bucle DO-WHILE do { cuerpo del bucle } while (condicion);
Estructuras Definición: struct nom_structura { tipo1 campo1; tipo2 campo2; ... };	Estructuras Uso: struct nom_structura var; var.campo1 var.campo2 ...	Estructuras anidadas Uso: struct nom_structura_anidada var; var.campo1.campoX...

Biblioteca math.h (funciones matemáticas)

sqrt (v)	Raíz cuadrada de v
pow (x, y)	Eleva el valor x a la potencia y
abs (x)	Devuelve el valor absoluto de x

Biblioteca stdio.h (funciones de manejo de pantalla, teclado y ficheros)

printf ("cadena de formato", exp1, exp2, ...);	Escribe el mensaje o valores de exp1, exp2 según las secuencias de sustitución de la cadena de formato
scanf ("cadena de formato", &variable);	Pide por teclado valores según se especifique en su cadena de formato y la asigna a una o más variables
putchar(c);	Imprime el carácter cuyo código ASCII es c en pantalla
"%d" "%c" "%x" "%f" "%lf" "%s"	Secuencias de sustitución para imprimir, o pedir por teclado: un entero decimal, un carácter, un valor hexadecimal, un valor float, un valor double, y una cadena (en scanf, los espacios terminan la cadena)

Biblioteca string.h (funciones de manejo de cadenas)

strcpy (a,b);	Copia la cadena b a la cadena a
strcat (a,b);	Añade por la derecha el contenido de la cadena b a la cadena a
strcmp (a,b);	Compara alfabéticamente la cadena a con b. Devuelve 0 si son iguales, 1 si a>b y -1 si a<b
strlen (a)	Devuelve el número de caracteres en la cadena a

SQL

Sintaxis básica de una consulta: SELECT atributo1, atributo2,..... FROM entidad WHERE condición..... (opcional) GROUP BY atributo,..... (opcional) ORDER BY atributo,..... [DESC] (opcional)	GROUP BY da un resultado igual a DISTINCT sólo si los campos de GROUP BY son los mismos que hay en SELECT Si en GROUP BY hay menos campos que en SELECT , los campos que haya en SELECT que no estén en GROUP BY sólo pueden ser campos agregados
Agregados: COUNT() AVG() MAX() MIN() SUM()	