

GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL
INFORMÁTICA - CURSO 1º - 2023-2024
2º CONVOCATORIA

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 desea gestionar un curso académico. En dicho curso, se registran asignaturas que son definidas por el nombre, créditos, estudiantes matriculados y sus correspondientes calificaciones. Para ello se proponen las siguientes estructuras y variables globales que recogen toda la información:

```
struct Estudiante {
    int dni;
    char nombre[50];
};

struct Asignatura {
    char nombre[50];
    float credits;
    struct Estudiante estudiantes[100]; // Estudiantes matriculados en la asignatura
    float calificaciones[100]; // Calificación correspondiente a cada uno de los
    // estudiantes siguiendo el mismo orden que en el vector anterior
    int numEstudiantes; // Número de estudiantes matriculados
};

struct Curso {
    int nombre;
    struct <Asignatura> asig[50];
};

struct Curso c; // Variable con información del curso
```


GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL
INFORMÁTICA - CURSO 1º - 2023-2024
2º CONVOCATORIA

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

a) (1,75 puntos) Implemente una función, llamada `calculaNotaMedia`, que recibe el DNI de un estudiante (`dni`). La función devuelve la nota media del expediente académico de dicho estudiante. Para ello, se deben sumar los créditos obtenidos por el estudiante multiplicados cada uno de ellos por el valor de las calificaciones que correspondan, y dividida por el número de créditos totales obtenidos por el estudiante.

```
float calculaNotaMedia (int dni)
{
```

b) (1,5 puntos) Implemente una función que muestre por pantalla los nombres de las asignaturas donde todos los estudiantes matriculados consten como aprobados.

```
void muestraEstadisticasAprobados ()  
{
```

GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL
INFORMÁTICA - CURSO 1º - 2023-2024
2º CONVOCATORIA

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

c) (1,75 puntos) Implemente una función que recibe como parámetro el nombre de una asignatura y muestre por pantalla el nombre del estudiante matriculado con mayor calificación.

```
void muestraMejorEstudiante (char nombreAsig[])  
{
```


GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL
INFORMÁTICA - CURSO 1º - 2023-2024
2º CONVOCATORIA

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

Problema 3 (1 punto): Analice el siguiente código y escriba cómo quedaría el vector cada vez que se haga una modificación. Indique qué se mostraría por pantalla al finalizar el programa.

Nota: Indique todo el procedimiento necesario que haya llevado a cabo para dar con la solución.

```
int main()
{
    int v[10] = {3,5,0,4,1,1,2,0,3,1};
    int i, j;

    for(i=0; i<2; i++)
    {
        for(j=3; j>1; j--)
        {
            if(i>0 && v[j]<=2)
            {
                v[j+2] = v[j+4];
                printf("[1]");
            }
            else if(i>=1 || j<3)
            {
                v[j*2-i] = 0;
                printf("[2]");
            }
            else
            {
                v[v[j]] = i + j;
                printf("[3]");
            }
        }
        printf("\n");
    }
}
```


GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL
INFORMÁTICA - CURSO 1º - 2023-2024
2º CONVOCATORIA

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

v	3	5	0	4	1	1	2	0	3	1

Se muestra por pantalla:

GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL
INFORMÁTICA - CURSO 1º - 2023-2024
2ª CONVOCATORIA

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

Problema 2 (2 puntos): Suponga que tenemos una matriz de 11 filas y 20 columnas cuyas 10 primeras filas están rellenas con valores enteros sin signo. Realice un programa que rellene la última fila de la siguiente forma:

- La última fila de cada columna PAR debe rellenarse con la SUMA de los valores del resto de filas de esa columna.
- La última fila de cada columna IMPAR debe rellenarse con el valor MÁXIMO de los valores del resto de filas de esa columna.

Véase el siguiente ejemplo para una matriz más pequeña (5 filas y 6 columnas)

5	7	1	8	4	6
9	9	5	3	1	2
1	4	7	6	3	7
3	8	1	6	3	5
18	9	14	8	11	7

En este ejemplo:

- En la columna 1 de la última fila, al ser columna impar, se ha escrito un 9 al ser el máximo de los valores 7,9,4,8
- En la columna 2 de la última fila, al ser columna par, ha escrito un 14 que es la suma de los valores de esa columna: $1+5+7+1=14$
- Este comportamiento se mantiene para el resto columnas

```
unsigned int valores[11][20]; // = {{8,4,7,3,6,2,9,6,4,3,...},...};
```

```
void main()  
{
```

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()	