

GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL
INFORMÁTICA - CURSO 1º - 2020-2021
PRIMERA CONVOCATORIA

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

NOTA ACLARATORIA EXTRA: 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.

Problema 1 (4 puntos): Una tienda dedicada a la venta de música online precisa una aplicación para la gestión de las reseñas (valoraciones) que los usuarios han realizado de las canciones que tiene disponible en su página web.

La información necesaria para realizar dicha gestión está organizada de la siguiente forma:

```
struct Cancion
{   char titCanc[30];           // Título de la canción
    int valoraciones[1000];     // Reseña de cada usuario puntuada entre 1 y 5.
    int numValoraciones;       // Número de reseñas registradas
    int descargas;             // Número de veces que ha sido descargada la canción
};

struct Disco
{   int idDisco;               // Número que identifica al disco en el sistema.
    char titDisco[30];         // Título del disco.
    struct Cancion canciones[100]; // Información de las canciones que forman el disco.
                                    // Las canciones están organizadas en el vector según
                                    // el orden de aparición en el disco, considerando
                                    // que la primera canción se encuentra en la posición 0.
    int numCanciones;          // Número de canciones que posee el disco.
};

struct Disco infoDisc[1000];   // Contiene toda la información de la tienda.
int numDiscos;                 // Número de discos actualmente registrados para la venta.
```

Realice las implementaciones que se indican en los siguientes apartados teniendo en cuenta que **deberá reutilizar, cuando sea posible, las funciones que se han solicitado en apartados anteriores,** independientemente de que las haya desarrollado o no. Por ejemplo, al implementar la función del apartado b) deberá reutilizar la función del apartado a) si ésta resulta útil para implementar la función del apartado b) aunque el alumno no haya realizado la función de dicho apartado a).

a) (*1 punto*) Realice una función que reciba el identificador de un disco (*idDisco*) y que devuelva la posición en la que se encuentra dentro del vector *infoDisc*. La función devolverá -1 si no hay disco con ese identificador.

```
int getPosDisco(int idDisc)
```

b) (1.5 punto) Realice una función que reciba por parámetros una canción mediante el identificador del disco (*idDisco*) al que pertenece y la posición que ocupa en el disco (comenzando en 0), y devuelva la valoración media que ha obtenido esa canción expresada con números decimales.

En caso de no encontrar el disco indicado, de no existir ese número de canción en el disco o no tener valoraciones, la función devolverá -1;

```
float getValoracion (int idDisc, int posCanc)
```

c) (1.5 punto) Realice una función que imprima por pantalla el título de la canción mejor valorada de todas y devuelva la valoración obtenida.

```
float theBest()
```

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	Condición múltiple IF-ELSE-IF if (condicion1) { bloque1 } else if (condicion2) { bloque2 } ... else { bloque por defecto }	Condición 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 (no se 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 [DISTINCT] tabla1.campo1, tabla2.campo2,..... FROM tabla1, tabla2, tabla3..... WHERE condición..... (opcional) GROUP BY tabla.campo,..... (opcional) ORDER BY tabla.campo,..... [DESC] (opcional)	tabla1,tabla2.... deben ser entidades, o relaciones de muchos a muchos. 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()	

(Intencinadamente en blanco)

(Intencinadamente en blanco)

GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL
INFORMÁTICA - CURSO 1º - 2020-2021
PRIMERA CONVOCATORIA

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

Problema 2 (2 punto): Escriba el cuerpo de una función que toma como argumentos un vector v de n números enteros. La función debe calcular la media de todos los valores de v , posteriormente, buscar en v el valor más cercano a dicha media, y devolverlo. El valor vc más cercano a la media (m) es aquel que hace que el valor absoluto de la diferencia con m sea mínimo:

vc es tal que $|m - vc| = \min |m - v_i| \quad \forall i \in \{0, \dots, n - 1\}$. *NOTA: v_i es lo que en C escribimos $v[i]$*

Si hay varios valores que equidistan de m , la función deberá devolver el primero que encuentre, además de mostrar por pantalla cuántos valores equidistantes hay.

Ejemplo: si v vale $\{2, -3\}$, n vale, por tanto, 2, y m vale 2.5, la función devolverá el valor 2 y mostrará por pantalla un 2 (debido a que existen dos valores equidistantes a m).

Puede usar la función `abs(x)`, ya definida, que devuelve el valor absoluto del argumento x que se le envíe. El prototipo de la función es: `int busca_mas_cercano (int v[], int n);`