

<p align="center">GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL INFORMÁTICA - CURSO 1º - 2022-2023 1ª CONVOCATORIA</p>			
Nombre:	Apellidos:	DNI:	

Ejercicio 1 (1.2 puntos): realice las siguientes operaciones. Tenga en cuenta que:

- Sólo se puede operar entre números que estén en la misma base.
- **Indique el resultado usando el menor número de bits posible.**
- Cualquier conversión cuyo cálculo no esté indicado, **no se tomará como válida**.

Operación	Resultado			
$010101_{(Ca2)} - 0100111_{(S-M)}$	(10)	(S-M)	(Ca2)	(Hex-Ca2)
$A7_{(Hex-Ca2)} - B4_{(Hex-S-M)}$	(10)	(S-M)	(Ca2)	(Hex-S-M)
$5_{(10)} \times 1110110_{(Ca2)}$	(10)	(S-M)	(Ca2)	(Hex-Ca2)

Zona de cálculos

Ejercicio 2 (1 punto): calcule la frecuencia de reloj que debería tener un procesador de 4 núcleos cuyo CPI es de 2 para que ejecute un programa dos veces más lento que otro procesador moncore cuyo CPI es 4 con un período de reloj de 2 ns. Tenga en cuenta que el porcentaje de instrucciones paralelizables es de un 74%.

Ejercicio 3 (0.7 puntos): Un procesador posee el tamaño de bus de direcciones de 32 bits y está conectado a una memoria principal de 8GB*. Indique el tamaño mínimo del bus de datos que debería tener el procesador para poder direccionar la memoria al completo. ¿Qué ocurriría si ampliáramos 1GB más a la memoria?

* 1 GB = 2^{30} B

GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL INFORMÁTICA - CURSO 1º - 2022-2023 1ª CONVOCATORIA			
Nombre:	Apellidos:	DNI:	

Ejercicio 4 (0.6 puntos): Responda de forma justificada, clara y breve las siguientes cuestiones:

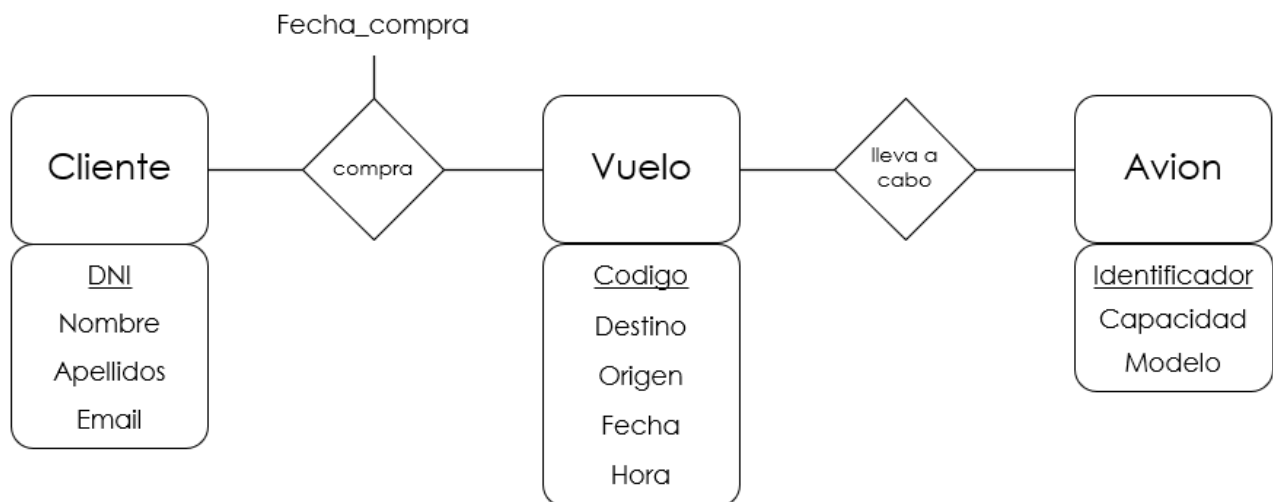
- a) Represente un esquema de la jerarquía de memoria.

- b) Defina qué es el sistema de E/S de un computador e indique de qué se compone.

- c) Defina qué es un modelo de red y nombre uno.

- d) Explique qué es el planificador de tareas e indique al menos una función del mismo.

Ejercicio 5 (0.5 puntos): Dado el siguiente modelo E/R:



a) Realice las siguientes consultas SQL:

- Muestre el nombre del avión con mayor capacidad.

- Muestre toda la información de los clientes cuyo DNI acabe en “Y” y cuyo email contenga “@” o acabe en “.com”.

- Muestre el número de vuelos cuyo destino sea Barcelona en todo el mes de agosto. Nota: las fechas se escriben en formato “día/mes/año”.

b) Indique una ocurrencia de “Compra”.

GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL
INFORMÁTICA - CURSO 1º - 2022-2023
1ª CONVOCATORIA

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

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.

Problema 1 (3 puntos): La Consejería de Sostenibilidad, Medio Ambiente y Economía Azul nos ha encargado el desarrollo de una aplicación para monitorizar el crecimiento anual de los árboles de un determinado parque natural. Las estructuras y variables encargadas de almacenar la información de cada árbol son las siguientes:

```
struct Coordenada // Coordenadas geográficas en grados decimales de cada árbol
{
    float latit; // Latitud: posición Norte-Sur (Mayor latitud hacia el Norte)
    float longit; // Longitud: posición Oeste-Este (Mayor longitud hacia el Este)
};

struct Arbol
{
    char tipo[25]; // Tipo de árbol: Pino, encina, roble...
    int crecim[1000]; // Centímetros que ha crecido en cada año de vida, en el
                    // que la posición 0 indica lo que creció su primer año.
    int edad; // Años de vida del árbol. Si -1, el árbol está muerto.
    struct Coordenada ubic; // Coordenadas GPS que sitúan el árbol en el parque.
};

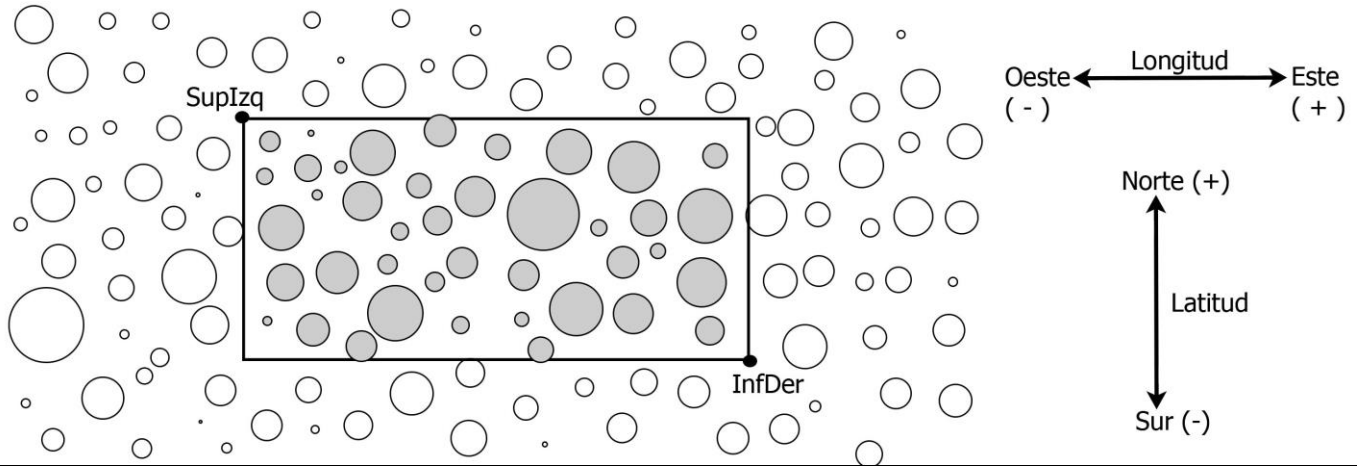
struct Arbol arboleda[1000]; // Información de cada árbol.
int numArboles; // Número de árboles registrados en el parque
```


a) (1.5 punto) Implemente una función que busque en el vector el primer árbol que sea del tipo especificado por el parámetro *tipoArb* con una altura igual o superior a la indicada por el parámetro *alturaMin*. La función devolverá la posición en la que se encuentra dicho árbol en el vector o -1 si no encuentra ninguno.

Nota: Para obtener la altura de un árbol, deberá sumar el crecimiento que ha registrado en cada año de su vida.

int buscarArbol (char tipoArb[25], int alturaMin)

b) (1.5 punto) Realice una función que devuelva el número de árboles vivos que hay en una determinada zona del parque. La zona está determinada por un rectángulo definido por 2 coordenadas *SupIzq* e *InfDer* recibidas por parámetros, donde la coordenada *SupIzq* indica el vértice superior izquierdo del rectángulo e *InfDer* el vértice inferior derecho.



```
int numArbolesZona(struct Coordenada SupIzq, struct Coordenada InfDer)
```


GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL
INFORMÁTICA - CURSO 1º - 2022-2023
1ª CONVOCATORIA

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

Problema 2 (2 puntos): Realice un programa que calcule la moda (el valor que más se repite) por cada fila de una matriz de 50x100. Suponga que los valores de la matriz sólo pueden estar entre el 0 y el 2. Suponga también que la matriz ya se encuentra declarada e inicializada en el programa.

Ejemplo. Dada la siguiente matriz:

1	0	0	2	.	1	1	1	1
2	2	0	2		0	2	1	2
0	0	0	1		1	2	0	0
...					...			
0	0	2	0		0	2	1	1

La impresión de pantalla comenzaría:

En la fila 0, la moda es 1

En la fila 1, la moda es 2

En la fila 2, la moda es 0

...

```
int main()
```

```
{
```

```
    int m[50][100] = {{1,0,0...}, ... , {...2,1,0}};
```


GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL
INFORMÁTICA - CURSO 1º - 2022-2023
1ª CONVOCATORIA

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

Problema 3 (1 punto): Analice el siguiente código y escriba qué se mostraría en consola tras su ejecución.

Nota: aquel resultado que no esté justificado, no se tomará como válido, aunque sea correcto.

```
int funcion(int a, int b);

int main()
{
    int i, y=0, res;
    int x[8] = {1, 2, 3, 4, 5, 6, 7, 8};

    for (i=0; i<8; i=i+2){
        res = funcion(x[i], i);
        y = y + res;
    }
    printf("Fin - %d\n", y);
    return 0;
}

int funcion(int a, int b){
    int res;
    res = a * b;
    printf("%d\n", res);
    return res;
}
```

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()	