



1. Plantee e implemente un programa en lenguaje C, que utilizando exclusivamente subprogramas, solicite números enteros por teclado y vaya almacenándolos en un array de 10 elementos, mostrando a continuación por pantalla el contenido del array creado.
2. Plantee e implemente un programa en lenguaje C, que utilizando exclusivamente subprogramas, solicite nombres de animales por teclado y vaya rellenando un array de 7 elementos, mostrándolos posteriormente por pantalla apropiadamente.
3. Plantee e implemente un programa en lenguaje C, que utilizando exclusivamente subprogramas, calcule la suma de los elementos de un array de números reales proporcionados por el usuario, considerando como máximo 30 elementos.
4. Plantee e implemente un programa en lenguaje C que almacene en memoria las horas de estudio que dedica un alumno a una asignatura determinada durante una semana. Solicite los datos apropiadamente al usuario y calcule y muestre la media de horas diarias empleadas en esa asignatura, asimismo indique el día en el que ha estado más aplicado (mayor número de horas estudiando) y más vago (menor número de horas estudiando).
5. Plantee e implemente un programa en lenguaje C para almacenar las temperaturas (en °C) registradas en una ciudad durante un día hora a hora (0-23). Introduzca las temperaturas a partir de los datos proporcionados por el usuario, visualice ordenadamente las temperaturas registradas y calcule y muestre por pantalla la temperatura máxima, mínima y media alcanzadas.
6. Plantee e implemente un programa en lenguaje C que, partiendo de la estructura creada en el ejercicio anterior, cree un sistema para almacenar las temperaturas (en °C) registradas en una ciudad durante una semana (hora a hora cada día) Introduzca las temperaturas solicitando los datos apropiadamente al usuario. Calcule y muestre por pantalla las temperaturas medias diarias, las temperaturas medias de cada hora y la temperatura media semanal.

7. Plantee e implemente un programa en lenguaje C que solicite al usuario su presupuesto semanal (en euros) y sus gastos diarios en distintos conceptos (por ejemplo transporte, alimentación y ocio) durante una semana. Calcule y muestre por pantalla el gasto total de la semana, el gasto semanal y el gasto medio diario en los distintos conceptos y el % del presupuesto semanal que gastó cada día.
8. Plantee e implemente un programa en lenguaje C que, mediante búsqueda secuencial, permita encontrar el menor y el mayor índice de donde se encuentra un valor dado X en un array de números enteros. El programa debe permitir al usuario introducir apropiadamente el array de números enteros y solicitarle el valor que desea encontrar.

EJEMPLO DE EJECUCIÓN

Introduzca los elementos del array separados por espacios, <enter> para terminar:

1 2 3 5 3 2 3 4 5 6 3 1 2 3

Introduzca el número que desea buscar: 3

El número 3 aparece por primera vez en la posición 3 y por última vez en la posición 14.

Introduzca el número que desea buscar: 4

El número 4 aparece por primera y única vez en la posición 8.

Introduzca el número que desea buscar: 9

El número 9 no aparece en el array.

9. Plantee e implemente un programa en lenguaje C que construya un array a partir de números enteros proporcionados por el usuario (máximo 20 elementos) y que a continuación sustituya los números pares por su cuadrado y los números impares por la parte entera de la raíz cuadrada de su valor absoluto. Muestre por pantalla tanto el array original como el resultante de las operaciones indicadas.
10. Plantee e implemente una función en lenguaje C que reciba dos arrays de enteros desordenados de 50 elementos cada uno y devuelva un array ordenado con los elementos de los dos anteriores.
11. Plantee e implemente una función recursiva en lenguaje C que devuelva la suma de los elementos de un array de enteros.

12. Plantee e implemente una función en lenguaje C que indique si un array de enteros es capicúa.
13. Plantee e implemente un programa en lenguaje C que permita introducir apropiadamente una matriz $N \times N$ de números enteros ($0 < N < 20$), la visualice apropiadamente por pantalla (formato matricial) y muestre el elemento mayor y el elemento menor de dicha matriz y sus posiciones.
14. Plantee e implemente un programa en lenguaje C que permita introducir apropiadamente una matriz $N \times N$ de enteros y que, a petición del usuario, intercambie las filas o columnas i y j de dicha matriz. Muestre por pantalla en formato matricial tanto la matriz original introducida como la matriz resultante del intercambio de filas o columnas solicitado por el usuario.
15. Plantee e implemente un programa en lenguaje C que permita introducir apropiadamente una matriz $N \times N$ de números reales y que muestre por pantalla en formato matricial tanto la matriz original introducida como su matriz traspuesta. Realice lo mismo considerando como tipo base de la matriz números enteros.
16. Plantee e implemente un programa en lenguaje C que permita introducir un cuadrado ($N \times N$) por teclado y determine si es mágico. Un cuadrado mágico es una matriz $N \times N$ de números enteros del 1 al N^2 en la que la suma de sus filas, sus columnas y sus diagonales principales coincide.
17. Plantee e implemente un programa en lenguaje C que mediante un menú apropiado permita al usuario introducir dos matrices $N \times N$ de números enteros ($0 < N < 10$) y seleccionar la operación (suma, resta o producto) que desea realizar con ellas. Muestre los resultados de la operación seleccionada por pantalla en formato matricial.
18. Plantee e implemente un programa en lenguaje C que permita introducir apropiadamente una matriz $N \times N$ de números enteros ($0 < N < 10$), la visualice apropiadamente por pantalla (formato matricial) y muestre la posición de un elemento *punto de silla*, si existe. Una matriz presenta un *punto de silla* si alguna posición de la misma es al mismo tiempo el menor valor de su fila y el mayor valor de su columna.

19. Plantee e implemente un programa en lenguaje C que rellene tres arrays con 500 elementos enteros de forma aleatoria y posteriormente muestre por pantalla todos los números conseguidos de forma ordenada de menor a mayor.
20. Plantee e implemente un programa en lenguaje C que permita almacenar en memoria el nombre y las notas de una asignatura de una clase de 15 alumnos. Muestre por pantalla el nombre y nota de los alumnos que han obtenido la peor y la mejor calificación. Además muestre a continuación el listado con el nombre y nota de todos los alumnos.
21. Diseñe la estructura de datos que permita almacenar en memoria la lista de tareas a realizar durante los días del año. Se debe de poder controlar la fecha, la descripción de tareas a realizar cada día con su hora y minutos. Tenga en cuenta que cada día no podrá tener más de 20 tareas.
22. Plantee e implemente una función en lenguaje C que reciba dos arrays de reales desordenados de 30 elementos cada uno y devuelva una array ordenado con los elementos de los dos anteriores.
23. Plantee e implemente una función en lenguaje C que reciba dos arrays de tipo base carácter, sin finalizar en '\0', es decir, no son cadenas; ambos desordenados y con 100 elementos cada uno y devuelva un array ordenado con los elementos de los dos anteriores.
24. Diseñe la estructura para almacenar en memoria la información de los trabajadores de un taller de 20 empleados. Si el empleado es oficinista tendrá un número de teléfono; si es conductor tendrá asignado el número de matrícula de un vehículo; y si es técnico tendrá asignado el código de barras de un ordenador. Se debe conocer el nombre, la fecha de nacimiento, tipo de puesto de trabajo y el nivel de estudios de todos los trabajadores.
25. Plantee e implemente un programa en lenguaje C que lea una frase de teclado (secuencia de caracteres hasta fin de línea), y determine la frecuencia de aparición de cada vocal respecto al total de caracteres de la frase.
26. Plantee e implemente un programa en lenguaje C que lea una frase (como máximo de 100 caracteres) y proporcione al usuario la posibilidad de sustituir alguna palabra de las introducidas por un sinónimo. Muestre por pantalla la frase original y la frase después de llevar a cabo la sustitución.

27. Plantee e implemente un programa en lenguaje C que lea una frase (como máximo de 100 caracteres) y que muestre a continuación cada palabra de la misma seguida del número de letras que la componen. Para simplificar suponga que las palabras pueden separarse únicamente por espacios, comas o puntos.

EJEMPLO DE EJECUCIÓN

Introduzca una frase de como máximo 100 caracteres:

Buenos días, vamos a comenzar. Encended los equipos.

Buenos	6
días	4
vamos	5
a	1
comenzar	8
Encended	8
los	3
equipos	7

28. Plantee e implemente un programa en lenguaje C que lea una frase (secuencia de caracteres hasta fin de línea), sustituya todas las secuencias de dos o más espacios por un único espacio y que muestre por pantalla la frase obtenida.
29. Diseñe una función que devuelva al revés la cadena recibida como parámetro.
30. Diseñe una función que sume dos números de 2500 cifras cada uno.
31. Implemente la función booleana *VAL (tCadena cadena)* la cual devuelve TRUE si puede convertir la cadena de 9 caracteres a un valor real y FALSE en caso contrario.
32. Implemente una función que reciba una matrícula, la edad y el nombre de un conductor y devuelva TRUE si la suma de los números de la matrícula es igual a la edad del conductor y las letras de la matrícula coinciden con las iniciales del nombre y los apellidos del propietario del vehículo.
33. Implemente un programa que lea una frase, carácter a carácter del teclado y que calcule y muestre el número de letras, dígitos y otros caracteres.

34. Implemente un programa que lea un mínimo de 20 palabras del teclado hasta que aparezca el carácter <ENTER> y guarde en un array las 20 primeras leídas. Por cada palabra guardada debe figurar en el array la siguiente información:
- Si la palabra está en mayúsculas
 - Número de vocales que la componen
 - Número de veces que la palabra aparece en el texto
 - Frecuencia de los caracteres que componen cada palabra
35. Implemente un programa que guarde en un vector las capitales europeas y su población. Dicho programa debe permitir mostrar las capitales existentes en el array de forma ordenada, tanto por nombre de capital como por número de habitantes.
36. Implemente un programa que solicite cinco números enteros ($n_i > 0$) al usuario y muestre por pantalla cada número y su descomposición factorial. Utilice una estructura de datos adecuada para almacenar a la vez los números y su descomposición.
37. Implemente un programa que lea una fecha y la almacene en una cadena que quepan sus 10 caracteres (dd/mm/aaaa) añadiendo ceros a la izquierda del día, mes y año si son necesarios) y calcule y almacene en memoria los siguientes 100 años bisiestos. El programa debe dar la opción al usuario de cambiar la fecha introducida y ver por pantalla los n años bisiestos que desee a partir de la fecha dada.
38. Implemente un programa en lenguaje C que tras leer un texto, indique el número de palabras que tienen un mínimo de tres vocales diferentes.
39. Implemente un programa en lenguaje C que permita modificar los nombres de los destinatarios de las cartas escritas y guardadas en una estructura de datos.
40. Plantee e implemente un programa en lenguaje C para simular el juego de la vida de John Conway.

