

## Ejercicios UT6 –Estructuras de datos: Arrays.

### Programas de manipulación de datos simples en vectores.

1. Realice un programa que defina un array llamado “*numeros*” de diez enteros. A continuación, inicialice el array con valores aleatorios (entre 1 y 100) y finalmente muestre en pantalla cada elemento del vector junto con su cuadrado y su cubo.
2. Desarrolle un programa que solicite diez números enteros por teclado, los almacene en un array y muestre la media aritmética, el valor máximo, el valor mínimo y las posiciones que ocupan estos dos últimos valores en el array.
3. Desarrolle un programa que declare un array de diez elementos enteros y pida números para rellenarlo. La entrada de datos finalizará cuando se llenen todas las posiciones del array o se introduzca un número negativo. Finalmente, el programa debe imprimir los valores que han sido introducidos, así como la suma y el producto de todos ellos.
4. Escriba un programa que genere veinte números enteros aleatorios entre 0 y 99 y los almacene en un array. El programa debe crear un nuevo array con los números primos que existan entre esos 20 números aleatorios. Finalmente, el programa debe mostrar los dos arrays por pantalla.
5. Desarrolle un programa que cree un array de caracteres que contenga de la ‘A’ a la ‘Z’ (solo las mayúsculas). Después, solicite posiciones del array por teclado y si la posición es correcta, la letra asociada se añadirá a una cadena que se mostrara al final. La solicitud de datos de entrada finalizará cuando el usuario introduzca el valor -1. Por ejemplo, si introducen los números siguientes:
  - 0 (Añadirá la ‘A’)
  - 5 (Añadirá la ‘F’)
  - 25 (Añadirá la ‘Z’)
  - 50 (Error, inserte otro número.)
  - -1 (Fin)
  - Cadena resultante: AFZ
6. Escriba un programa que invierta el orden de los valores de un array. Por invertir el orden de los valores de un array, se entiende que el último valor pasa a ser el primero, el penúltimo el segundo y así sucesivamente.
7. Realice un programa que pida al usuario que introduzca un texto y cree una palabra con las iniciales de las palabras del texto introducido. Por ejemplo: texto introducido = “Hay osos libres arriba”, la palabra que tendría que pintar el programa por pantalla sería: “Hola”.
8. Desarrolle un programa que genere un array aleatorio de diez números enteros y construya otro array cuyos elementos se obtengan como la diferencia entre dos términos consecutivos del array inicial.

### Programas de búsqueda y clasificación.

9. Elabore un programa que solicite una fecha en formato dd/mm/aaaa y verifique que la fecha es correcta. Para ello, utilice un array que almacene los días que tiene cada mes.
10. Desarrolle un programa que permita validar un DNI introducido por teclado. Consulte en Internet el procedimiento que se utiliza para conocer qué letra le corresponde a un número. Almacene la tabla de correspondencias que encontrará en un array.
11. Escriba un programa que busque un valor dentro de un array de enteros y borre todas sus ocurrencias reduciendo consecuentemente la longitud del array. Es decir, no vale con poner cero en el lugar donde se encuentre el valor. El programa debe mostrar por pantalla el array antes de borrar las ocurrencias del número y el array después. El array de enteros estará **“hardcodeado”** y será: 1, 5, 9, 3, 45, 23, 45, 12, 87, 9, 6, 5.
12. Realice un programa que pida al usuario N números enteros y los inserte de forma ordenada en un vector de N elementos, siendo N un valor que se solicitará al principio. El programa deberá permitir la búsqueda de elementos dentro del vector mediante “búsqueda dicotómica”. Para ello (y mientras el usuario no decida salir) se pedirá el número que se desea saber si existe en el vector y se devolverá, en ese caso, su posición o se informará de que no existe. Para verificarlo, muestre el contenido del vector por pantalla.

### Programas de manipulación de objetos en vectores.

13. Desarrolle un programa que permita almacenar en un vector los nombres y las edades de los alumnos de un curso. El programa comenzará solicitando los datos de los alumnos finalizando el proceso de lectura de datos cuando se introduzca como nombre un asterisco (\*). Tras la lectura de los datos, el programa mostrará la información siguiente: Relación de alumnos mayores de edad y relación de alumno o alumnos más jóvenes (los que tienen menor edad).
14. De una empresa de transporte se quiere guardar el nombre de los conductores que tiene y los kilómetros que conducen cada día de la semana. Elabore un programa que permita crear la plantilla con un número N de conductores y desde un menú de opciones permita dar de alta conductores, añadir kilómetros al historial de un conductor, dar de baja conductores, modificar el nombre de un conductor y listar la información de todos los conductores. El programa también debe mostrar el total de kilómetros realizados por todos los conductores de la plantilla.

## Programas de manipulación de datos en matrices.

15. Desarrolle un programa que lea los elementos de una matriz de 4 filas y 5 columnas de números double y genere un array de longitud 4 en el que cada elemento contenga el sumatorio de los elementos de cada fila. El programa debe mostrar la matriz original y el array en este formato:

1	2	3	4	5		15
9	8	7	6	5	—	35
2	3	4	1	8	—	18
5	6	3	4	2		20

16. Diseñar el algoritmo correspondiente a un programa, que crea una tabla bidimensional de longitud 5x15 y nombre 'marco' y carga la tabla con dos únicos valores 0 y 1, donde el valor uno ocupará las posiciones o elementos que delimitan la tabla, es decir, las más externas, mientras que el resto de los elementos contendrán el valor 0.

```
111111111111111
100000000000001
100000000000001
100000000000001
111111111111111
```

17. Se necesita un programa que permita jugar de forma sencilla al tres en raya (dos jugadores A y B). El programa almacenará el estado del tablero de acuerdo con el siguiente código: 0 en una casilla significa que está libre; 1 representa que hay una ficha del jugador A y 2 representa que hay una ficha del jugador B. Realice los métodos necesarios para:

- Mostrar el estado actual del tablero.
- Solicitar la posición de la ficha a introducir y guardarla en el tablero.
- Comprobar si el tablero está lleno.
- Verificar si la jugada actual proporciona un tres en raya.

18. Una empresa informática le solicita que desarrolle una versión simplificada del juego de los barcos. Para su diseño le dan las siguientes especificaciones:

- a. El tablero es una cuadrícula de 8x8 casillas.
- b. Al inicio del juego se solicitarán las coordenadas de cada uno de los barcos. Por simplicidad se admitirá que hay 8 barcos, que cada uno está colocado en una fila, y que el número de casillas que ocupa cada barco es 2. Por tanto, para situar un barco en el tablero bastará con dar una coordenada admitiéndose que la segunda casilla ocupada por el barco se corresponde con la casilla que está situada a la derecha de la indicada. Así, si las coordenadas dadas por el usuario son (3,4) las casillas ocupadas por el barco serán (3,4) y (3,5). Para

- posicionar un barco en el tablero el sistema debe comprobar que las coordenadas introducidas son válidas.
- c. Una vez posicionados los 8 barcos en el tablero comienza el juego. El juego constará de 8 rondas. En cada ronda, el jugador dará las coordenadas de una casilla:
    - i. Si la casilla está ocupada un barco, el barco se considerará "hundido" y, por tanto, las 2 casillas ocupadas por este barco serán casillas "libres" en las siguientes rondas. Además, al hundir un barco, el sistema otorgará al usuario una ronda de juego adicional. Es decir, las rondas en las que el usuario hunde un barco no serán descontadas del total de rondas disponibles.
    - ii. Si la casilla está libre, no es una casilla válida o está ocupada por un barco ya hundido, se considerará que el usuario ha consumido una ronda.
  - d. El juego finaliza cuando se han hundido todos los barcos o se han agotado todas las rondas.
  - e. Una vez finalizado el juego se indicará por pantalla qué jugador ha ganado y la distribución inicial y final de los barcos en los tableros.

### Programas completos.

19. Desarrolle un programa para ser usado por los camareros de un restaurante a la hora de tomar nota de las comandas de su única mesa (es un restaurante muy íntimo). Lo primero que va a introducir el camarero es el número de comensales que ocupan la mesa. A continuación, preguntará a cada comensal su elección, las cuales constan de un primer plato, un segundo plato y un postre. El camarero podrá anotar también observaciones por plato y observaciones por comensal. El programa, finalmente imprimirá por pantalla los menús de cada mesa comenzando por el primer plato, luego el segundo y finalmente el postre.
20. Modifique la aplicación anterior para que el restaurante pierda intimidad en beneficio de la cuenta de resultados, es decir, que tendrá un número de mesas N, siendo este valor un número configurable al principio del programa.
21. Realiza un programa para acertar el nombre de una persona. Dicho nombre estará **"hardcodeado"** (no hay que pedir al usuario que introduzca los datos, sino que los escribimos directamente en el código). El programa pedirá al usuario repetidamente que introduzca un nombre hasta que acierte. A cada fallo, el programa le dará una pista que consiste en mostrar por pantalla las letras del nombre hasta la posición igual al número de fallos que se lleven. Por ejemplo: Nombre = "Eustaquio"
  - 1ª Introducción: "Ana" -> fallo -> 1ª pista: "E"
  - 2ª Introducción: "Enrique" -> fallo -> 2ª pista: "Eu"
  - 3ª Introducción: "Eulalia" -> fallo -> 3ª pista: "Eus"
  - 4ª Introducción: "Eusebio" -> fallo -> 4ª pista: "Eust"
  - 5ª Introducción: "Eustaquio" -> acierto -> fin del programa
22. La resonancia magnética es una técnica de diagnóstico por imagen en la que se obtiene información de las características de diferentes puntos del cuerpo a partir de su respuesta a un campo magnético variable.

Supongamos que tenemos una matriz tridimensional de puntos que representa una imagen adquirida por este método, en el que cada punto tiene un valor entre 0 y 255, correspondiente a un distinto nivel de gris. La imagen será un cubo de  $100 * 100 * 100$  puntos.

Se quiere escribir un programa que permita detectar aspectos en la imagen que pueden ser indicadores de una enfermedad.

Para ello se pide:

- Escribir un método que, para un determinado punto en la imagen (identificado por sus coordenadas  $x, y, z$ ), detecte si es sospechoso. Se consideran sospechosos aquellos puntos para los que TODOS los puntos adyacentes tengan un valor entre 20 y 40 (esto incluye los puntos pertenecientes al mismo plano, al plano inferior y al plano superior). **Nota:** A la hora de analizar la imagen, no se analizarán los puntos de ninguno de los planos exteriores del cubo.
- Escribir un método que, para un determinado plano de la imagen, busque líneas sospechosas. Se considera una línea sospechosa toda línea HORIZONTAL (paralela al eje  $x$ ) en la que haya al menos tres puntos sospechosos consecutivos.
- Escribir un programa que emplee los métodos anteriormente descritos para identificar en una imagen si hay alguna línea sospechosa. Los valores de todos los elementos de la matriz se supondrán ya cargados en memoria (no es necesario leerlos). En caso de que existan líneas sospechosas en la imagen, el programa mostrará en qué plano hay más líneas sospechosas (servirá para saber en qué lugar es mejor hacer una biopsia).

