EJERCICIOS UD. 7
Estructuras de almacenamiento

C/ Gabilondo, 23 –47007 Valladolid ☎ 983 471600/471026 FAX: 983 235866 – www.gregoriofer.com e-mail: centro@gregoriofer.com

VECTORES

EJERCICIO 1

Crea un programa **Personas** que almacene en un array los nombres de 20 personas introducidos por teclado. Diseña los siguientes métodos:

- *imprimePersonas*: visualiza por pantalla los elementos del array, una en cada fila.
- *pares*: visualiza por pantalla los elementos del array que ocupan las posiciones pares. Cada elemento debe ir en una fila.

EJERCICIO 2

Crea un programa **Datos** que almacene en un array 10 números enteros. Imprime por pantalla los elementos que ocupan las posiciones pares y su suma utilizando un método llamado *sumaPares*.

EJERCICIO 3

Crea un programa **Datos2** que visualice los elementos pares que ocupan las posiciones impares del array creado en el ejercicio anterior, su cuenta y su suma. Además de imprimir las posiciones que ocupan dichos elementos.

EJERCICIO 4

Crea un programa **Frases** que almacene en un array unidimensional 5 frases que se introducen por teclado. Diseña los siguientes métodos:

C/ Gabilondo, 23 –47007 Valladolid ☎ 983 471600/471026 FAX: 983 235866 – www.gregoriofer.com e-mail: centro@gregoriofer.com

- *imprimeFrases*: imprime por pantalla el contenido del array.
- *mayorFrase:* imprime por pantalla la frase de mayor longitud y la posición que ocupa en el array.
- *menorFrase*: imprime la frase más pequeña y la posición que ocupa.

EJERCICIO 5

Escribe un programa **ListaAleatoria** que cree e imprima por pantalla un array de 10 elementos con números aleatorios comprendidos entre 1 y 10, de tal forma que no se repita ninguno.

EJERCICIO 6

Crea un programa **Capicua** que compruebe si un número es capicúa utilizando un array.

ARRAYS DE OBJETOS

EJERCICIO 7

Realiza un programa que permita gestionar una agenda de contactos telefónicos, los cuales se almacenarán en un array de tamaño 100.

Cada contacto de la agenda será un objeto de tipo **Contacto**, con los atributos, id (único), "nombre" y "tf". Ten en cuenta que no se podrán crear contactos sin nombre.

La agenda permitirá realizar las siguientes operaciones:

- Añadir un nuevo contacto. Si ya existe un contacto con ese nombre ó teléfono, se le informará al usuario antes de su almacenamiento.

C/ Gabilondo, 23 –47007 Valladolid ☎ 983 471600/471026 FAX: 983 235866 – www.gregoriofer.com e-mail: centro@gregoriofer.com

- Buscar un contacto por nombre ó teléfono. Si existen varios contactos coincidentes, se mostrarán todos.
- Modificar los datos de un contacto (previa búsqueda).
- Eliminar un contacto (previa búsqueda). Antes de su eliminación se pedirá confirmación al usuario.
- Mostrar un listado de todos los contactos almacenados en la agenda, ordenado por nombre o teléfono.
- Vaciar la agenda.

MATRICES

EJERCICIO 8

Escribe un programa **Matriz1** que genere un array bidimensional 5x5, de tal forma que sus filas pares sean múltiplos de 2 y las impares sean múltiplos de 3. Además diseña los siguientes métodos:

- *imprimirMatriz*: muestra por pantalla la matriz creada.
- *sumaMatriz*: muestra la suma de todos sus elementos.
- *diagonal*: imprime los elementos de su diagonal principal.

EJERCICIO 9

Crea un programa **MatrizTraspuesta** que genere una tabla 4x5 cuyos valores sean aleatorios entre 1 y 100. A partir de ella crea su traspuesta. Se deben mostrar por pantalla ambas tablas. Utiliza los métodos necesarios para ello.

C/ Gabilondo, 23 –47007 Valladolid ☎ 983 471600/471026 FAX: 983 235866 – www.gregoriofer.com e-mail: centro@gregoriofer.com

NOTA: La traspuesta de una matriz es aquella matriz que se consigue cambiando las filas por columnas o viceversa.

EJERCICIO 10

Crea un programa **Permutación** que trabaje con una matriz de enteros cuya dimensión será solicitada al usuario, y se rellenará con aleatorios entre 1 y 100. El programa deberá permitir realizar permutaciones a través de los siguientes métodos:

- *permutaFilas*: recibe como parámetros un array bidimensional de enteros "m", un entero "fila1" y otro entero "fila2". El método debe intercambiar las filas "fila1" y "fila2".
- *permutaColumnas*: de forma análoga intercambia columnas.

EJERCICIO 11

Crea un programa **Rombo** que rellene una matriz con la figura de un rombo. La dimensión de la matriz y el carácter de relleno serán introducidos por teclado.

BÚSQUEDA Y ORDENACIÓN

EJERCICIO 12

Escribe un programa **Ordenacion1** que cree una lista de palabras introducidas por teclado, el tamaño de la lista se pedirá al usuario. Seguidamente se imprimirá la lista por pantalla, a continuación, deberás ordenar la lista por longitud de las palabras (tomando como base el método de ordenación de la **burbuja**) y finalmente imprimir la lista ordenada.

C/ Gabilondo, 23 –47007 Valladolid ☎ 983 471600/471026 FAX: 983 235866 – www.gregoriofer.com e-mail: centro@gregoriofer.com

EJERCICIO 13

Escribe un programa **Ordenacion2** que cree una lista de N palabras introducidas por teclado (N es un entero que se pedirá al usuario). Imprime la lista., ordénala alfabéticamente (método de la **burbuja**) e imprímela de nuevo. Crea un método que devuelva el número de palabras de la lista que empiezan por un carácter que se solicita al usuario.

COLECCIONES

EJERCICIO 14

Realiza un programa que permita gestionar las notas de las diferentes asignaturas de DAM de los alumnos de una clase.
Cada alumno, dispone de 3 notas correspondientes a las 3 evaluaciones.

El programa permitirá añadir la nota a cada asignatura en cada evaluación. También mostrará un listado de las medias de la clase en cada asignatura, y un listado de los alumnos ordenados alfabéticamente por apellidos.

EJERCICIO 15

Usa una tabla hash (HashMap) para almacenar clientes y realizar las operaciones básicas.

Un Cliente tiene los siguientes atributos:

- id: numérico y único.
- Dni.
- Nombre.

C/ Gabilondo, 23 –47007 Valladolid ☎ 983 471600/471026 FAX: 983 235866 – www.gregoriofer.com e-mail: centro@gregoriofer.com

Operaciones a realizar:

- Introducir clientes en el mapa.
- Modificar los datos de un cliente (a excepciÛn de su id).
- Eliminar clientes del mapa.
- Buscar clientes por id y por dni.
- Mostrar todos los clientes del mapa.

PILAS Y COLAS

EJERCICIO 16

Crea un programa **InvertirArray** que dado un array de enteros invierta su orden utilizando una pila.

EJERCICIO 17

Realiza un programa **Parentesis** que reciba desde el teclado una cadena con un conjunto de paréntesis, "(" o ")", e indique si están bien o mal cerrados.

Ejemplos:

)	mal cerrado
()	bien cerrado
(()	mal cerrado
)(	mal cerrado
(()()(()))	bien cerrado
((()))()	mal cerrado

EJERCICIO 18

Partiendo de dos pilas de números enteros, la primera ordenada ascendentemente desde la cima hasta el fondo, y la segunda ordenada descendentemente desde la cima hasta el fondo, realiza un programa que fusione ambas pilas en una tercera ordenada descendentemente desde la cima hasta el fondo.

NOTAS:

- Los tamaños de las dos pilas iniciales, se generarán aleatoriamente entre 1 y 20.
- Igualmente, los contenidos de las pilas de partida, se generarán aleatoriamente entre 1 y 100.
- No se puede utilizar ninguna pila auxiliar para realizar la fusión de las pilas.