

UNIDAD NUEVE

ACTIVIDADES

(arrays. Ordenacion, Busqueda e insercion)

1. Implementar los métodos de ordenación, búsqueda e inserción (con distintas interfaces), e incluirlos en clase de utilidades o en la clase array ya implementada.
2. Modificar el algoritmo de búsqueda secuencial para el caso en que el array esté ordenado.
3. Una empresa discográfica realiza una encuesta sobre los temas de actualidad. Dispone de un total de 100 éxitos a los que se puede votar. Cada persona encuestada podrá dar una lista de sus 5 éxitos favoritos. El programa deberá dar el número de éxito con los puntos que ha obtenido, imprimiendo una lista que los contenga, ordenada de mayor a menor número de votos. No se imprimirán los que no obtengan votos.
4. En una liga de fútbol participan N equipos. En esta liga, cada vez que un equipo gana un partido, recibe 3 puntos, cuando empata, 1 punto, y cuando pierde, ningún punto.

Los resultados de los distintos partidos se introducen por teclado tras cada jornada, en una entrada que contiene nombre del equipo local (*local*) y sus goles (*golesLocal*), y nombre del equipo visitante (*visitante*) y sus goles (*golesVisitante*).

Se pretende procesar todos los resultados y almacenarlos en tablas con los datos *nombreEquipo*, *partidosJugados*, *golesMarcados*, *golesRecibidos* y *puntos*. Las tablas están generadas inicialmente con random y ordenadas.

Realiza un programa que lea los resultados de cada jornada y actualice las tablas. A continuación deberá ordenarse la tabla por puntuación de mayor a menor y, por último, presentarla en pantalla. La ordenación deberá ser congruente que significa que, en caso de igualdad en los puntos, debe mantenerse el orden previo.

5. Una compañía de confección de vestidos quiere una lista de los diferentes diseños pedidos durante la semana. Para preparar la lista, se dispone de las fichas de peticiones de ventas que contiene los siguientes datos: código de diseño (*codDiseño*) y cantidad pedida (*cantidad*).

Tenemos un array (*diseños*), que se inicializará al declararlo, no ordenado que contiene los 10 códigos de diseños (todos de tres dígitos enteros positivos) y sus precios.

Realizar un programa para imprimir una lista ordenada por *codDiseño* con los diferentes diseños pedidos en la semana, junto con la cantidad total pedida de ellos, su correspondiente precio unitario y el coste total de cada uno.

Especificaciones:

- a. Todos los campos son del mismo tipo de datos.
- b. El programa debe verificar que el código de diseño de entrada sea correcto, es decir se encuentre en el array *diseños*.
- c. La información se procesa por teclado y el proceso acaba cuando a la pregunta ¿Quedan más fichas?, se responda "NO" (¡Ojo! esto es una cadena).
- d. Puede haber fichas referidas al mismo vestido y puede haber diseños que no se hayan pedido esa semana.