Actividad resuelta 5.1

Crear una tabla de longitud 10 que se inicializará con números aleatorios comprendidos entre 1 y 100. Mostrar la suma de todos los números aleatorios que se guardan en la tabla.

Actividad resuelta 5.2

Diseñar un programa que solicite al usuario que introduzca por teclado 5 números decimales. A continuación, mostrar los números en el mismo orden que se han introducido.

Actividad resuelta 5.3

Escribir una aplicación que solicite al usuario cuántos números desea introducir. A continuación, introducir por teclado esa cantidad de números enteros, y por último, mostrar en el orden inverso al introducido.

Actividad resuelta 5.4

Diseñar la función: `int maximo(int t[])`, que devuelva el máximo valor contenido en la tabla `t`.

Actividad resuelta 5.5

Escribir la función `int[] rellenaPares(int longitud, int fin)`, que crea y devuelve una tabla ordenada de la longitud especificada, que se encuentra rellena con números pares aleatorios comprendidos en el rango desde 2 hasta `fin` (inclusive).

Actividad propuesta 5.4

Escribe la función: `int buscar(int t[], int clave)`, que busca de forma secuencial en la tabla `t` el valor `clave`. En caso de encontrarlo, devuelve en qué posición lo encuentra; y en caso contrario, devolverá `-1`.

Actividad resuelta 5.6

Definir una función que tome como parámetros dos tablas, la primera con los 6 números de una apuesta de la primitiva, y la segunda (ordenada) con los 6 números de la combinación ganadora. La función devolverá el número de aciertos.

Actividad propuesta 5.5

Escribe en una función el comportamiento de la inserción ordenada.

Actividad resuelta 5.9

Diseñar una aplicación para gestionar un campeonato de programación, donde se introduce la puntuación (enteros) obtenidos por 5 programadores, conforme van terminando su prueba. La aplicación debe mostrar las puntuaciones ordenadas de los 5 participantes. En ocasiones, cuando finalizan los 5 participantes anteriores, se suman al campeonato programadores de exhibición, cuyos puntos se incluyen con el resto. La forma de especificar

que no intervienen más programadores de exhibición es introducir como puntuación un –1. La aplicación debe mostrar, finalmente, los puntos ordenados de todos los participantes.

Actividad resuelta 5.10

Escribir la función:

```
int[] eliminarMayores(int t[], int valor)
```

que crea y devuelve una copia de la tabla `t` donde se han eliminado todos los elementos que son mayores que `valor`.

Actividad propuesta 5.6

Crea una función que realice el borrado de un elemento de una tabla ordenada.

Actividad propuesta 5.7

El «número de la suerte» de una persona puede calcularse a partir de sus números favoritos. De entre estos, se seleccionan dos diferentes al azar, que se eliminarán de la lista, pero en su lugar se añade la media aritmética de los dos eliminados a la lista de números favoritos. El proceso se repite hasta que solo quede un número, que resultará el número de la suerte para esa persona. Para calcular bien el número de la suerte es imprescindible que la lista de números se encuentre siempre ordenada.

Escribe una aplicación que solicite al usuario sus números favoritos y calcula su número de la suerte.

Actividad propuesta 5.8

Comprueba qué produce la comparación con el operador `==` de dos tablas del mismo tipo, la misma longitud y los mismos valores.

Actividad resuelta 5.11

Desarrollar el juego «la cámara secreta», que consiste en abrir una cámara mediante su combinación secreta, que está formado por una combinación de dígitos del uno al cinco. El jugador especificará cuál es la longitud de la combinación; a mayor longitud, mayor será la dificultad del juego. La aplicación genera, de forma aleatoria, una combinación secreta que el usuario tendrá que acertar. En cada intento se muestra como pista, para cada dígito de la combinación introducido por el jugador, si es mayor, menor o igual que el correspondiente en la combinación secreta.

Actividad resuelta 5.12

Crear una tabla bidimensional de longitud 5 × 5 y rellenarla de la siguiente forma: el elemento de la posición [n][m] debe contener el valor 10 × n + m. Después se debe mostrar su contenido.

5.1. Una tabla puede almacenar datos de distintos tipos, como por ejemplo enteros, booleanos, reales, etcétera:

a) Cierto, las tablas siempre pueden almacenar datos de distintos tipos.
b) Falso, las tablas solo pueden almacenar datos de un único tipo.
c) Puede almacenar datos de distintos tipos siempre que estos sean numéricos.
d) Puede almacenar datos de distintos tipos siempre que la longitud de los datos sea idéntica.

5.2. En Java, la numeración de los índices que determina la identificación de cada elemento de una tabla comienza en:

a) Cero.
b) Uno.
c) Depende del tipo de dato de la tabla.
d) Es configurable por el usuario.

5.3. Si en una tabla de 10 elementos utilizamos el elemento con índice 11 (que se encuentra fuera de rango):

a) Al salir del rango de la longitud, Java redimensiona la tabla de forma automática.
b) No es posible y produce un error.
c) Las tablas tienen un comportamiento circular y utilizar el índice 11 es idéntico a utilizar el índice 1.
d) Ninguna de las anteriores respuestas es cierta.

5.4. ¿Qué método de la clase Arrays permite realizar una búsqueda dicotómica en una tabla?

a) `Arrays.search()`.
b) `Arrays.find()`.
c) `Arrays.binarySearch()`.
d) Cualquiera de los métodos anteriores realiza una búsqueda.

5.5. Con respecto a las tablas, el operador `new`:

a) Destruye, crea y redimensiona tablas.
b) Destruye y crea tablas.
c) Crea tablas.
d) Destruye las tablas.

5.6. La forma de invocar al recolector de basura es:

a) Mediante `System.garbageCollector()`.
b) Mediante el operador `new`.
c) Mediante `Arrays.garbageCollector()`.
d) Ninguna de las anteriores respuestas es correcta.

5.7. La forma de conocer la longitud de una tabla `t` es mediante:

a) `t.size`.
b) `t.elements`.
c) `t.length`.
d) `Arrays.size(t)`.

5.8. La comparación del contenido (los elementos) de dos tablas se realiza utilizando:

a) `Arrays.compare()`.
b) El operador `==`.
c) `Arrays.equals()`.
d) `Arrays.same()`.

5.9. ¿Qué condición tiene que cumplir una tabla para que podamos realizar búsquedas dicotómicas en ella?

a) Que esté ordenada.
b) Que esté ordenada y sea una tabla de enteros.
c) Que no esté ordenada.
d) No importa si la tabla está ordenada, lo realmente importante es que sea de algún tipo numérico.

5.10. ¿Cuál es la principal diferencia entre `Arrays.copyOf()` y `System.arraycopy()`?

a) No existe diferencia alguna, ambos métodos son idénticos.
b) `Arrays.copyOf()` copia mientras `System.arraycopy()` copia y compara.
c) `Arrays.copyOf()` copia entre tablas existentes mientras `System.arraycopy()` crea una nueva tabla y copia en ella.
d) `Arrays.copyOf()` crea una nueva tabla y copia en ella mientras `System.arraycopy()` solo copia entre tablas ya creadas.

5.11. Realiza la función: `int[] buscarTodos(int t[], int clave)`, que crea y devuelve una tabla con todos los índices de los elementos donde se encuentra la clave de búsqueda. En el caso de que `clave` no se encuentre en la tabla `t`, la función devolverá una tabla vacía.

5.12. Escribe la función `void desordenar(int t[])`, que cambia de forma aleatoria los elementos contenidos en la tabla `t`. Si la tabla estuviera ordenada, dejaría de estarlo.

5.13. Modifica la Actividad de aplicación 5.12 para que la función no modifique la tabla que se pasa como parámetro y, en su lugar, cree y devuelva una copia de la tabla donde se han desordenado los valores de los elementos.

5.14. El ayuntamiento de tu localidad te ha encargado una aplicación que ayude a realizar encuestas estadísticas para conocer el nivel adquisitivo de los habitantes del municipio. Para ello, tendrás que preguntar el sueldo a cada persona encuestada. *A priori,* no conoces el número de encuestados. Para finalizar la entrada de datos, introduce un sueldo con valor –1.

Una vez terminada la entrada de datos, muestra la siguiente información:

- Todos los sueldos introducidos ordenados de forma decreciente.
- El sueldo máximo y mínimo.
- La media de los sueldos.

5.15. Debes desarrollar una aplicación que ayude a gestionar las notas de un centro educativo. Los alumnos se organizan en grupos compuestos por 5 personas. Leer las notas (números enteros) del primer, segundo y tercer trimestre de un grupo. Debes mostrar al final la nota media del grupo en cada trimestre y la media del alumno que se encuentra en una posición dada (que el usuario introduce por teclado).

5.16. En un juego de rol el mapa puede implementarse como una matriz donde las filas y las columnas representan lugares (lugar 0, lugar 1, lugar 2, etc.) que estarán conectados. Si desde el lugar X podemos ir hacia el lugar Y, entonces la matriz en la posición [x][y] valdrá cierto; en caso contrario, valdrá falso. Escribe una función que, dada una matriz que representa el mapa y dos lugares, indique si es posible viajar desde el primer lugar al segundo (directamente o pasando por lugares intermedios).

5.17. Implementa la función: `int[] suma(int t[], int numElementos)`, que crea y devuelve una tabla con las sumas de los `numElementos` elementos consecutivos de `t`. Veamos un ejemplo, sea t = [10, 1, 5, 8, 9, 2]. Si los elementos de `t` se agrupan de 3 en 3, se harán las sumas:

10 + 1 + 5. Igual a 16.
1 + 5 + 8. Igual a 14.
5 + 8 + 9. Igual a 22.
8 + 9 + 2. Igual a 19.

Por lo tanto, la función devolverá una tabla con los resultados: [16, 14, 22, 19].

5.18. Escribe un programa que solicite los elementos de una matriz de tamaño 4 × 4. La aplicación debe decidir si la matriz introducida corresponde a una matriz mágica, que es aquella donde la suma de los elementos de cualquier fila o de cualquier columna valen lo mismo.

5.19. Diseña una aplicación para gestionar la llegada a meta de los participantes de una carrera. Cada uno de ellos dispone de un dorsal (un número entero) que los identifica. En la aplicación se introduce el número de dorsal de cada corredor cuando este llega a la meta. Para indicar que la carrera ha finalizado (han llegado todos los corredores a la meta), se introduce como dorsal el número –1.

A continuación, la aplicación solicita información extra de los corredores. En primer lugar, se introducen los dorsales de todos los corredores menores de edad; para premiarlos por su esfuerzo se les avanza un puesto en el ranking general de la carrera, es decir, es como si hubieran adelantado al corredor que llevaban delante. En segundo lugar, se introducen los dorsales de los corredores que han dado positivo en el test antidopaje, lo que provoca su expulsión inmediata. Para finalizar, se introducen los dorsales de los corredores que no han pagado su inscripción en la carrera, lo que provoca que se releguen a los últimos puestos del ranking general. La aplicación debe mostrar los dorsales de los corredores que han conseguido las medallas de oro, plata y bronce.

5.20. La fusión de dos tablas ordenadas consiste en copiar todos sus elementos (de ambas tablas) en una tercera que deberá seguir ordenada. Podemos realizar una fusión «ineficiente» copiando los elementos de ambas tablas (sin tener en cuenta el orden) en la tabla final y ordenar esta. Existe una manera óptima de realizar la fusión en la que se elige en cada momento el primer elemento no copiado de alguna de las tablas y se añade a la tabla final, que seguirá ordenada sin necesidad de ordenación alguna.

Busca información sobre el algoritmo de fusión e impleméntalo en Java.