

-
- 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 x 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.