

Tema 6.- Ejercicios propuestos

1. Realiza una función que devuelva el elemento máximo de un vector que recibe como parámetro de entrada.
2. Diseñar un algoritmo en pseudocódigo que, dado un vector de enteros, determine si el valor de algún elemento coincide con la suma de todos los que están a su izquierda en el vector.
3. Realiza una función que devuelva la matriz producto de otras dos que recibe como parámetros de entrada.
4. Realiza un procedimiento que genere y escriba por pantalla la traspuesta de una matriz que recibe como parámetro de entrada.
5. Realiza una función que devuelva la matriz resultante de permutar las filas i y j de una matriz. Suponemos que se recibe como parámetro i , j y la matriz.
6. Diseña un algoritmo en pseudocódigo que determine si una matriz de tamaño $N \times M$ es un cuadrado mágico. Se considera que una matriz es un cuadrado mágico si es cuadrada y las filas, columnas y diagonales principal y secundaria suman lo mismo.
7. Realiza una función que devuelva verdadero si una matriz que recibe como parámetro de entrada es un cuadrado mágico y falso en caso contrario.
8. Se dice que una matriz de dos dimensiones posee un punto de silla en una cierta posición, si el valor de dicha posición es mayor que el de todas las de su fila, y menor que el de todas las de su columna, o viceversa (si el valor de dicha posición es menor que el de todas las de su fila, y mayor que el de todas las de su columna). Diseña un algoritmo en pseudocódigo que lea una matriz de $N \times M$ elementos enteros (N y M conocidos), y muestre por pantalla cuántos puntos de silla contiene la matriz y las posiciones que ocupan.
9. Realiza una función que invierta una cadena de caracteres y devuelva la cadena invertida. Suponemos que podemos acceder directamente a los elementos de la cadena, con la notación `cadena[indice]`, para acceder al carácter que se encuentra en la posición `indice` dentro de la cadena.
10. Realiza una función que devuelva verdadero si una cadena que recibe como parámetro es un palíndromo y falso si no lo es. Una cadena de caracteres es un palíndromo si puede leerse igual de derecha a izquierda que de izquierda a

derecha, salvo por los espacios en blanco, mayúsculas y minúsculas. Realiza una versión iterativa y otra recursiva suponiendo que podemos acceder directamente a los elementos de la cadena, con la notación `cadena[indice]`.

11. Suponiendo que **sólo** disponemos de las siguientes funciones de manejo de cadenas de caracteres:

- entero **longitud** (E cadena: cad)
Devuelve la longitud de la cadena de entrada ó 0 si la cadena de entrada es vacía.
- cadena **subcadena** (E cadena: cad_fuente, E entero: pos, E entero: long)
Devuelve la subcadena de cad_fuente que comienza en la posición pos y tiene longitud long. Si $\text{pos} > \text{longitud}(\text{cad_fuente})$ o $\text{long} \leq 0$ ó $\text{pos} \leq 0$ entonces se devuelve la cadena vacía.
- entero **posicion** (E cadena : cad, E cadena: subcad)
Devuelve la posición de la cadena cad en la que comienza la secuencia de caracteres de la cadena subcad. Devuelve 0 si subcad es una cadena vacía o no aparece en la cadena cad. En caso que subcad aparezca más de una vez dentro de cad, se devuelve la primera ocurrencia.
- cadena **concatena** (E cadena: cadena1, E cadena: cadena2)
Devuelve la cadena resultado de concatenar las cadenas cadena1 y cadena2 en este orden.

Diseña en pseudocódigo las funciones siguientes:

- a) cadena **insertar** (E cadena : cad, E cadena: subcad , E entero : pos)
Devuelve la cadena resultado de insertar en cad, a partir de la posición pos, la cadena subcad.
- b) cadena **sustituir** (E cadena: cad, E cadena: cad_orig, E cadena: cad_sust)
Devuelve la cadena resultado de sustituir en cad la primera ocurrencia de la subcadena cad_orig por cad_sust.
- c) entero **ocurrencias** (E cadena: cad, E cadena: subcad)
Devuelve el número de veces que aparece la cadena subcad dentro de la cadena cad.
- d) cadena **sustituir_todas**(E cadena: cad, E cadena: cad_orig, E cadena: cad_sust)
Devuelve la cadena resultado de sustituir en cad todas las ocurrencias de la subcadena cad_orig por cad_sust.

12. Una empresa posee 5 gasolineras (que podemos numerar de 1 a 5), cada una con 10 vendedores (numerados de 1 a 10), y vende 4 tipos diferentes de combustible: normal, súper, sin plomo y gas-oil (codificados de 1 a 4).

Diseña un algoritmo en pseudocódigo que lea los datos de cada una de las ventas efectuadas en las gasolineras: número de vendedor, número de gasolinera, tipo de combustible, e importe de la venta. El final de los datos de entrada se indicará con un número de vendedor igual a 0.

A partir de estos datos, debes obtener un listado con el importe total de las ventas realizadas en cada gasolinera y de las realizadas por cada vendedor.

Además se desea conocer qué tipo o tipos de combustible son los que más dinero han facturado.

NOTA: Cada vendedor puede tener más de una venta.

13. En la planta de Nefrología de un hospital se trata un máximo de 100 pacientes renales que se encuentran en su domicilio. Cada uno de estos pacientes debe medir diariamente su peso (en Kg), temperatura corporal (en °C), su tensión arterial (“mínima” y “máxima”) (en mmHg) y número de pulsaciones por minuto; y periódicamente envía esta información al hospital.

Realiza en pseudocódigo:

- a) la estructura de datos necesaria para almacenar para cada paciente el nombre y apellidos, D.N.I., número de teléfono, y las medidas que ha tomado en todos los días del mes,
- b) un procedimiento para leer la información sobre todos los pacientes,
- c) un procedimiento que lea el D.N.I. de un paciente y muestre su número de pulsaciones medio, peso medio y temperatura media a lo largo del mes,
- d) un procedimiento que muestre el nombre de aquellos pacientes que han tenido en algún día del mes una tensión máxima superior a 160 mmHg o una tensión mínima superior a 100 mmHg.