

AEDD - Guía Práctica 6: Arreglos Bidimensionales (Matrices)***Ejercicios propuestos***

Problema 1: Una empresa posee 5 servidores y desea analizar el tráfico de correo electrónico entre ellos. Para esto almacenará los datos en una matriz correo de 5x5 elementos. El elemento correo[i][j] indicará el número de kb de correo enviados desde el servidor i hasta el servidor j.

Obtener la siguiente información:

- a) Total de correo (en kb) enviado por cada uno de los cinco servidores.
- b) Número del servidor que ha enviado mayor cantidad de kb en correos.
- c) Muestre los servidores que recibieron correo por encima de una cota M establecida (recibida como dato).

Problema 2: Una empresa de aerolíneas nacional posee aviones de 4 categorías y viaja a 50 localidades de zonas turísticas del país. Se dispone de un registro de todos los pasajes vendidos en una temporada en una matriz de pasajes vendidos indexada por código de categoría (0, 1, 2, 3) y código de destino (0 a 49). Además se dispone de un vector ordenado por código de destino que indica el nombre de cada localidad.

Se pide informar :

- a) La cantidad de pasajeros por localidad.
- b) La cantidad de pasajeros por categoría.
- c) Las localidades a las que no se vendió ningún pasaje.
- d) La localidad a la que viajó la menor cantidad de pasajeros.
- e) La cantidad de localidades a las cuales viajaron más de 1000 pasajeros.
- f) Indicar si existe o no alguna categoría de aviones para las que no se vendió ningún pasaje.

Problema 3: Escribir la función `es_triangular_superior(A)` que indique si la matriz recibida como argumento es triangular superior o no. Una matriz a es triangular superior si todos los elementos que están bajo la diagonal principal tienen el valor cero.

Por ejemplo, la siguiente matriz es triangular superior:

```
A  ([[1, 0, 40 ],
     [ 0, 6,  0 ],
     [ 0, 0,  1 ]])
es_triangular_superior(A) →  True
```

Problema 4: Escribir la función `es_diagonal(A)` que indique si la matriz recibida como argumento es diagonal o no. Una matriz A es diagonal si todos los elementos que no están en la diagonal principal tienen el valor cero. Por ejemplo, la siguiente matriz es diagonal:

```
A  ([[1, 0, 0 ],
      [ 0, 6, 0 ],
      [ 0, 0, 1 ]])
es_diagonal(A) → True
```

Problema 5: Un cuadrado mágico es una disposición de números naturales en una tabla cuadrada, de forma tal que las sumas de cada columna, de cada fila y de cada diagonal son iguales. Los cuadrados mágicos más populares son aquellos que tienen los números consecutivos desde el 1 hasta n^2 , donde n es el número de filas y de columnas. Escribir una función que reciba un arreglo e indique si se trata o no de un cuadrado mágico.

Ejemplos:

```
es_magico (arreglo ([[3, 1, 5],
                     [4, 7, 2],
                     [9, 8, 6]])) → False
```

```
es_magico (arreglo ([[2, 7, 6],
                     [9, 5, 1],
                     [4, 3, 8]])) → True
```

Problema 6: Diseñar y codificar una función `JUGADA` que reciba una matriz V de 80×40 que contiene información sobre el estado de cada casillero en una grilla o tablero de juego. Los posibles estados son 0 (vacía), 1 (con ficha roja), 2 (con ficha verde).

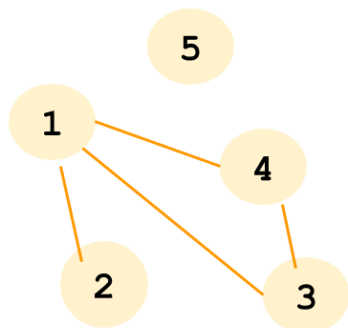
La jugada consiste en cambiar las fichas según los siguientes criterios:

- Donde hay 3 fichas verdes seguidas en una misma fila, dejar una ficha verde en el medio y vaciar las de los costados.
- Si una ficha verde está rodeada de fichas rojas, cambiar la verde por ficha roja y vaciar las rojas.

Problema 7: Escribir un programa que llene con una secuencia ordenada de números iniciada en 1, una matriz de afuera, comenzando desde el elemento (0,0) hacia adentro en forma de espiral.

1	2	3	4	5	6
20	21	22	23	24	7
19	32	33	34	25	8
18	31	36	35	26	9
17	30	29	28	27	10
16	15	14	13	12	11

Problema 8: Sea A ($N \times N$) una matriz que representa un *grafo no dirigido* (matriz de adyacencia) con el conjunto de nodos rotulados $NOD = \{ 1, 2, 3, \dots, N \}$.



0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
0	0	0	0	0

- A partir de un grafo, cargar su matriz de adyacencia.
- Determinar si un nodo es pendiente (nodo que tiene 1 sola arista, es decir que se encuentra conectado sólo a 1 nodo).
- Determinar si tiene nodos pendientes.
- Determinar si un nodo es aislado (nodo que no está conectado a ningún otro nodo).
- Determinar la cantidad de nodos aislados.
- Determinar si un grafo es completo (todo par de vértices (a, b) debe tener una arista que los une).

Problema 9: Una empresa distribuidora posee 4 sucursales y comercializa 25 artículos. Para analizar el desempeño de cada sucursal se ingresan los datos correspondientes a las ventas efectuadas en cierto período de la siguiente manera:

código sucursal, código artículo, cantidad unidades vendidas.

La carga de datos finaliza cuando el código de sucursal es igual a 0 (cero).

Informar:

- Las cantidades de unidades vendidas por la empresa de cada artículo.
- El total de unidades vendidas por la sucursal 3, sumando todos los artículos.
- La cantidad vendida por la sucursal 1 del artículo 6.
- La sucursal que vendió más unidades del artículo 8.

Emplee funciones para resolver cada caso.

Problema 10: Considerar el mismo enunciado del problema anterior. Incorpore -además- los datos de los precios unitarios de cada uno de los 25 artículos que comercializa la empresa (solicitar la información por teclado al comenzar la ejecución).

Informar:

- La recaudación de cada sucursal.

- b) La recaudación de la empresa.
- c) La sucursal que obtuvo mayor recaudación.

Problema 11: Se leen los datos de una matriz o tabla de 5x12 que contiene los valores en mm de lluvias producidas en 5 departamentos de la provincia de Santa Fe durante los 12 meses del año. Se sabe que en algunos departamentos existen datos faltantes, por lo que se ha completado con -1 la información faltante.

Escribir un programa C++ que a través de 2 funciones:

- a) Complete los datos de lluvia faltantes con el promedio anual de precipitación en el departamento correspondiente. Cada promedio debe obtenerse con los datos reales existentes (no contar los -1).
- b) Amplíe la matriz agregando una nueva columna con los totales de mm caídos en todo el año en cada departamento.

Informar:

- a) La matriz original de datos.
- b) La matriz modificada.
- c) Los totales anuales de lluvia por departamento.

Problema 12: Escribir un programa que, teniendo una matriz de $n \times n$, inicialice el valor de sus elementos en forma de espiral desde el elemento central hacia fuera.

```

25  24  23  22  21
10   9   8   7  20
11   2   1   6  19
12   3   4   5  18
13  14  15  16  17
    
```

Problema 13: Dadas dos matrices A y B de tamaño $N \times N$ y del mismo tipo de elementos, diseñar subalgoritmos para determinar si todos los números de la diagonal secundaria de A:

- a) Están en alguna celda de la segunda.
- b) Están en la diagonal principal de la segunda en cualquier orden.
- c) Están en la diagonal secundaria de la segunda en el mismo orden.

Problema 14: Dada la siguiente imagen en una matriz de caracteres ASCII, se provee el código con la matriz ya cargada y sus dimensiones.



A partir de esta matriz se propone:

Realizar una función Recortar que dado un punto (dos valores enteros) y dos dimensiones (dos valores enteros, ancho y alto), se muestre en pantalla el “recorte” indicado por el punto y las dimensiones.

Realizar una función QuitarRecuadro que dado un número entero muestre en pantalla la imagen resultante de remover esa cantidad de caracteres de los bordes de la matriz.

Realizar una función PorcentajeBlancoDiagPrincipal. Debe mostrar en pantalla el porcentaje de celdas en blanco (caracteres espacio) que hay por encima de la diagonal principal.

Realizar una función ContarCaracteres que dada la matriz imagen, genere un listado de todos los caracteres usados para conformar la misma, y muestre en pantalla la cantidad de dichos caracteres utilizados. No olvidar los espacios en blanco: también son caracteres.

Realizar una función ImprimirEnColor que dada la matriz imagen, un arreglo de caracteres y un arreglo de enteros (colores); imprima en pantalla la imagen cambiando el color de los caracteres indicados. Nota: Los arreglos pasados por parámetro deben tener el mismo tamaño:

```
char caracteres[]={ '@', '#', '%' };  
int colores[]={ 1,14,6 };  
ImprimirEnColor(image,caracteres,colores,3);
```