

## Práctica 6 - Matrices

1. Hacer un programa que dada una matriz de enteros de tamaño  $5 \times 10$  que se encuentra precargada, invierta el orden del contenido por fila. Este intercambio no se debe realizar de manera explícita, hay que hacer un método que incluya una iteración de intercambio.
2. Hacer un programa que dada una matriz de enteros de tamaño  $5 \times 10$  que se encuentra precargada, obtenga la cantidad de números pares que tiene y la imprima.

### **CORRIMIENTOS**

3. Hacer un programa que dada una matriz de enteros de tamaño  $5 \times 10$  que se encuentra precargada, solicite al usuario una posición fila, columna y realice un corrimiento a izquierda.
4. Hacer un programa que dada una matriz de enteros de tamaño  $5 \times 10$  que se encuentra precargada, solicite al usuario un número entero y una posición fila, columna. Con estos datos tendrá que realizar un corrimiento a derecha (se pierde el último valor en dicha fila) y colocar el número en la matriz en la posición fila, columna indicada.
5. Hacer un programa que dada una matriz de enteros de tamaño  $5 \times 10$  que se encuentra precargada, solicite al usuario un número entero y elimine la primera ocurrencia de número en la matriz (un número igual) si existe. Para ello tendrá que buscar la posición y si está, realizar un corrimiento a izquierda y no continuar buscando.
6. Hacer un programa que dada una matriz de enteros de tamaño  $5 \times 10$  que se encuentra precargada, solicite al usuario un número entero y elimine todas las ocurrencias de número en la matriz si existe. Mientras exista (en cada iteración tiene que buscar la posición fila y columna) tendrá que usar dicha posición para realizar un corrimiento a izquierda (quedarán tantas copias de la última posición de cada fila como cantidad de ocurrencias del número). Tener en cuenta, como se hizo en arreglos, si el elemento a eliminar es el último de alguna fila de la matriz.

### **ORDENAMIENTO**

7. Hacer un programa que dada una matriz de enteros ordenada creciente por filas de tamaño  $4 \times 5$  que se encuentra precargada, solicite al usuario un número entero y una fila, y luego inserte el número en la matriz en la fila indicada manteniendo su orden.
8. Hacer un programa que dada una matriz de enteros ordenada

creciente por filas de tamaño 4\*5 que se encuentra precargada, solicite al usuario un número entero y una fila, y elimine la primera ocurrencia de número en la fila indicada (un número igual) si existe.

9. Hacer un programa que dada una matriz de enteros de tamaño 4\*5 que se encuentra precargada, solicite al usuario el ingreso de una fila y dos números enteros (columnas de la matriz), y ordene de forma creciente la matriz en la fila indicada entre las dos posiciones columnas ingresadas.

### **SECUENCIAS**

Se tiene una matriz de enteros de tamaño 4\*20 de secuencias de números entre 1 y 9 (por cada fila), separadas por 0. La matriz está precargada, y además cada fila empieza y termina con uno o más separadores 0. Además, se tiene una matriz de caracteres de tamaño 4\*20 de secuencias de caracteres letras minúsculas entre 'a' y 'z' (por cada fila), separadas por ' ' (espacios). La matriz está precargada, y además cada fila empieza y termina con uno o más separadores ' '. Considere para los siguientes ejercicios estos dos tipos de matriz.

10. Hacer un programa que dada la matriz de secuencias de enteros definida y precargada, permita obtener a través de métodos la posición de inicio y la posición de fin de la secuencia ubicada a partir de una posición entera y una fila, ambas ingresadas por el usuario. Finalmente, si existen imprima por pantalla ambas posiciones obtenidas.
11. Hacer un programa que dada la matriz de secuencias de enteros definida y precargada permita encontrar por cada fila la posición de inicio y fin de la secuencia cuya suma de valores sea mayor.
12. Hacer un programa que dada la matriz de secuencias de caracteres definida y precargada, permita encontrar por cada fila la posición de inicio y fin de la anteúltima secuencia (considerar comenzar a buscarla a partir de la última posición de la fila).
13. Hacer un programa que dada la matriz de secuencias de enteros definida y precargada, y un número entero ingresado por el usuario, elimine de cada fila las secuencias de tamaño igual al número ingresado.
14. Hacer un programa que dada la matriz de secuencias de caracteres definida y precargada, elimine de cada fila todas las ocurrencias de una secuencia patrón dada por un arreglo de caracteres de tamaño igual al tamaño de columnas de la matriz (sólo tiene esa secuencia con separadores al inicio y al final). Al

eliminar en cada fila se pierden los valores haciendo los corrimientos.

15. Hacer un programa que dada la matriz de secuencias de caracteres definida y precargada elimine todas las secuencias que tienen orden descendente entre sus elementos.
16. Hay dos matrices MAT1 y MAT2 de secuencias de caracteres letras separados por espacios de tamaño MAXF x MAXC que están precargadas. Ambas matrices están precargadas y cada fila empieza y termina con caracteres espacios. Además se tiene el siguiente método:

Un método que retorna el índice inicial de la secuencia de mayor tamaño de un arreglo de secuencias (de caracteres letras minúsculas separados por espacios) de tamaño MAXFIL.

Se pide realizar un programa que:

- a. Contenga la definición de los encabezados de los métodos de carga de la matriz y del método mencionado en el enunciado (se supone que existen y no se requiere implementarlos).
- b. Para MAT1 y MAT2 elimine de cada secuencia el primer carácter vocal.
- c. Para MAT1 agregue al principio de cada secuencia el primer carácter de la secuencia de mayor tamaño de dicha fila. Considere agregar un espacio al final de la fila para mantener la estructura de secuencia delimitada por separadores espacio.
- d. En cada fila, si se verifica que la secuencia de mayor tamaño de la fila para MAT1 es mayor que la primera secuencia en dicha fila para MAT2, las intercambie (la que está en MAT1 pasa a MAT2 y la que está en MAT2 pasa a MAT1) sin usar estructuras auxiliares (otros arreglos o matrices).
- e. Para un valor de fila ingresado por el usuario verifique e imprima si la primera secuencia de MAT1 en dicha fila es igual a la primera secuencia de MAT2 en dicha fila.

## Bonus Track 1 - Ejercicio dominio real

1. Dado que el volumen de ventas ha aumentado, el gerente de

sistemas de la cadena de supermercados carrefive plantea la necesidad de modificar la estructura de almacenamiento de los datos. Hasta el momento se guardaban en un arreglo que tenía secuencias, para cada ticket de venta, separadas por cero. Dado que el arreglo debía ser enorme para almacenar todos los datos de ventas, se propone cambiar por una matriz en donde cada fila representa las ventas para un día. La matriz por lo tanto tendrá 31 filas y cada fila tendrá MAXCOL columnas (suficientes para almacenar todas las ventas de un día). Para poder almacenar las ventas de todo un año se podrían tener 12 matrices de las mismas características. Con estos cambios presentados, el departamento de ventas se muestra interesado en obtener más información que le sirve para asistir en la toma de decisiones en pos de mejorar la economía estimando los ingresos que tienen. Por ello solicita al gerente de sistemas que le prepare algunas rutinas para conocer:

- a. Cuántos tickets en promedio se emiten por día para un mes particular.
  - b.Cuál es el importe medio de venta para 1 día, 1 semana (considerar la primera semana del mes) y para un mes.
  - c.Cuál es el ticket de mayor importe del mes.
  - d.Cuál es el ticket con más productos vendidos en 1 día particular.
  - e. ¿Qué día del mes se vendió la mayor cantidad de productos?
2. Dado que el equipo de desarrollo es muy eficiente, el departamento de ventas le ha generado un nuevo requerimiento para que el sistema pueda emitir algunos informes extras:
  - a. dado un importe, el sistema le debe informar cuantos tickets superan dicho importe para 1 día particular y para todo el mes.
  - b. dada una cantidad de artículos, el sistema le debe informar cuántos tickets están por encima de esa cantidad para 1 día particular y para el mes entero.
  - c. dado un importe, el sistema debe informar cuantos artículos vendidos están por encima del mismo para 1 día particular.
  - d. dado un importe promedio, cuántos tickets superan en promedio a dicho importe para 1 día particular y para todo el mes.
3. Una refactorización del sistema se ha aplicado sobre la estructura que permite almacenar los tickets para poder soportar una nueva funcionalidad. En el nuevo esquema por cada día (fila de la matriz) cada secuencia tiene en primer lugar el id del producto, al lado la

cantidad vendida de ese producto para ese ticket. Ese se repite hasta que no hay más productos y el último lugar lo ocupa el importe total. Ejemplo:

0	3	24	7	1	203.56	0	99	2	87.5	0	12	8	3	5	1233.4	0
0	5	4	6	1	435.77	0	87	2	91.3	0	0	0	0	0	0	0

Para el ejemplo tenemos para un día (la primera fila) 3 tickets:

- i. en el primero para el producto de id 3 se vendieron 24, para el id 7, 1 solo y el importe de la venta fue de \$203.56
- ii. para el segundo sólo se vendieron 2 productos del id 99 por un importe de \$87.5
- iii. para el tercero se vendieron 8 del id 12 y 5 del id 3 todo por un importe de \$1233.4

Para el segundo día (la segunda fila) 2 tickets:

- iv. en el primero para el producto de id 5 se vendieron 4, para el id 6, 1 solo y el importe de la venta fue de \$435.77
- v. para el segundo se vendieron 2 artículos del id 87 por un importe de \$91.3

Con este esquema el departamento de ventas pretende conocer cuántos descuentos se aplicaron según el siguiente criterio: si la cantidad de productos iguales es mayor a 2 y es una cantidad par, el descuento fue del 10% del valor total del ticket. Tomando en cuenta estos descuentos el departamento de ventas quiere saber

- a. cuánto fue el total de descuentos aplicados en un día particular para lo cual necesita saber cuánto sumaron en pesos todos los descuentos aplicados. Para el ejemplo, en el primer día (primera fila) suma: \$152.45 (redondeado), para el segundo (segunda fila) suma: \$52.71(redondeado).
- b. cuantos productos tuvieron descuento un día particular (la cantidad de productos diferentes no la cantidad de unidades). Para el ejemplo, el primer día, fueron 3 artículos: el de id 3, id 99 e id 12. El segundo día fueron 2 artículos el de id 5 e id 87.
- c. cuánto fue el total de descuentos aplicados para todo el mes para lo cual necesita saber cuánto sumaron en pesos todos los descuentos aplicados. Para el ejemplo suman \$205.16.

## Bonus Track 2 - Ejercicio análisis de código

1. Dados los siguientes códigos, analizar y detectar los errores. Justificar en cada caso.

a.

```
public class Practica_6_Bonus_1 {
    public static final int MAXFIL=10, MAXCOL=10;
    public static void main (String [] args){
        int[][] matriz = new int[MAXFIL];
        int columna=6, fila=4;
        procesar(matriz[columna], fila);
    }
    // suponemos ya implementado
    public static void procesar(int[] arr, int pos){
        ...
    }
}
```

b.

```
public class Practica_6_Bonus_2 {
    public static final int MAX=10;
    public static void main (String [] args){
        char[][] mat= new int[MAX][MAX];
        int fila=6;
        correrIzquierda(mat, fila);
        mostrarArreglo(mat[fila]); //ya implementado
    }
    public static void correrIzquierda(char[][] mat,int pos){
        for (int i = pos ; i < MAX-1; i++) {
            mat[i]=mat[i+1];
        }
    }
}
```

c.

```
public class Practica_6_Bonus_3 {
    public static final int MAX=10;
    public static void main (String [] args){
        int[][] matriz = new int[MAX,MAX];
        int numero=6;
        int pos=buscarNumero(matriz, numero);
        System.out.println("La posicion del numero: "
            + numero + " es:" + pos);
    }
    public static int buscarNumero(int[][] mat, int numero){
        int i = 0 ;
        while (mat[i] != numero)
            i++;
    }
    if ( i<MAX)
        return i;
    return -1;
}
}
```

2. Dados los siguientes enunciados y códigos, analizar y detectar los

errores. Justificar en cada caso.

- a. Hacer un programa que dada una matriz de enteros de tamaño 10x10 (precargada), imprimir por pantalla el promedio de cada una de sus columnas.

```
public class Practica_6_Bonus_4 {
    public static final int MAXFILA=10, MAXCOLUMNA=10;
    public static void main (String [] args){
        int[][] matriz = new int[MAXFILA][MAXCOLUMNA];
        imprimir_promedio_matriz(matriz);
    }
    public void imprimir_promedios_matriz (int[][] mat){
        int promedio;
        for (int col = 0 ; col < MAXCOLUMNA; col++){
            System.out.println("Promedio de la columna
            "+col+" es "+obtener_promedio_columna(mat,col));
            promedio=0;
        }
    }
    public int obtener_promedio_columna(int[][] mat,int col){
        int promedio, suma = 0;
        for (int fila = 0 ; fila < MAXFILA; fila++) {
            suma+=mat[fila][col]/MAXCOLUMNA;
        }
        return promedio;
    }
}
```

- b. Escribir un programa que elimine todas las secuencias de tamaño par. No usar estructuras auxiliares.

```
public class Practica_6_Bonus_5 {
    final static int MAXFIL = 20, MAXCOL = 20;
    public static void main (String[] args) {
        int[][] matrizA=new int[MAXFIL][MAXCOL];
        int[][] matrizB=new int[MAXFIL][MAXCOL];
        cargarMatriz(matrizA);
        matrizB=eliminarSecuencias(matrizA);
    }
    public static void cargarMatriz(int[][] mat) {
        for (int i=0; i< MAXFIL ; i++){
            cargarArreglo(mat[i]);
        }
    }
    public static void cargarArreglo(int[] arr) {
        for (int pos = 0 ; pos < MAX; pos++)
            arr[pos]=pos*2;
    }
    return arr;
}

public static int[][] eliminarSecuencias(int[][] mat){
    int ini=0,fin=-1;
    for (int i=0; i< MAXFIL ; i++){
        while (fin<MAXCOL){
            ini=buscarIni(mat[i], fin+1);
            if (ini<MAXCOL){
                fin=buscarFin(mat[i],ini);
            }
        }
    }
}
```

```

        int tam=fin-ini+1;
        if (tam%2==0){
            for (;tam>0;tam-){
                correrIzq(mat[i],ini);
            }
            fin=ini+1;
        }
        else fin++;
    }
    else fin=MAXCOL;
}

}

}

public static int obtenerIni(int[] arr, int ini) {
    while (ini<MAX && arr[ini] == 0)
        ini++;
    return ini;
}

public static int obtenerFin(int[] arr, int ini) {
    while (ini<MAX && arr[ini] !=0)
        ini++;
    return ini-1;
}

}

```