**Alessandra Gonzalez  
  
Ejercicio 1:**

**Escribir un método que dada una matriz de enteros devuelva la suma de los elementos que contiene la matriz.**

public class Main

{

public static void main(String[] args) {

int matriz[][] =

{

{4, 1, 3},

{6, 4, 1}, //2x3

};

System.out.print(sumarMatriz(matriz));

}

public static int sumarMatriz(int matriz[][]){

int suma=0;

for(int fila=0; fila < matriz.length ; fila++){

for(int columna=0; columna < matriz[0].length; columna++){

suma = suma + matriz[fila][columna];

}

}

return suma;

}

}

**Ejercicio 2:**

**Escribir un método que dada una matriz de enteros devuelva la cantidad de elementos positivos que contiene la matriz.**

public class Main

{

public static void main(String[] args) {

int matriz[][] =

{

{1,2,3},

{4,3,1},

{1,1,-1}, //3x3

};

int pos = contarPositivos(matriz);

System.out.print("La cantidad de positivos son: "+ pos);

}

public static int contarPositivos(int matriz[][]){

int positivos=0;

for(int fila=0; fila < matriz.length ; fila++){

for(int columna=0; columna < matriz[0].length; columna++){

if(matriz[fila][columna] > 0){

positivos = positivos + 1;

}

}

}

return positivos;

}

}

**Ejercicio 3:**

**Escribir un método que dada una matriz de enteros y un entero que representa un índice de fila, devuelva la columna que contiene el máximo elemento de esa fila.**

public class Main

{

public static void main(String[] args) {

int matriz[][] =

{

{2, 3},

{1, -5}

};

int fila = 0;

int posicion = mayorColumna(matriz, fila);

System.out.print("El mayor se encuentra en la posicion: "+ posicion);

}

//devuelva la columna que contiene el máximo elemento de esa fila.

public static int mayorColumna(int matriz[][], int f){

int mayor = matriz[f][0];

int col = 0;

for(int columna=1; columna < matriz[f].length ; columna++){

if(mayor < matriz[f][columna]){

mayor = matriz[f][columna];

col = columna;

}

}

return col;

}

}

**Ejercicio 4:**

**Escribir un método que dada una matriz de enteros y un entero que representa un índice de columna, devuelva la fila que contiene el máximo elemento de esa columna.**

public class Main

{

public static void main(String[] args) {

int matriz[][] =

{

{1,2,30},

{40,5,6},

{4,8,9}

};

int columna=0;

System.out.print("La fila donde se encuentra el mayor: "+filaDelMayor(matriz, columna));

}

public static int filaDelMayor(int m[][], int colum){

int f=0;

int mayor=m[0][colum];

for(int fila=0; fila<m.length; fila++){

if(m[fila][colum] > mayor){

mayor = m[fila][colum];

f= fila;

}

}

return f;

}

}

**Ejercicio 5:**

**Escribir un método que dada una matriz de enteros devuelva la fila que contiene el máximo elemento de la matriz.**

public class Main

{

public static void main(String[] args) {

int matriz[][] = {

{5, 6, 90},

{11, -5, 0},

{33, 1, 2}

};

int fila = filaDelMaximo(matriz);

System.out.print("La fila donde se encuentra el mayor es: "+ fila);

}

public static int filaDelMaximo(int mat[][]){

int maximo= mat[0][0];

int f=0;

for(int fila=0; fila < mat.length ; fila++){

for(int columna=0; columna < mat[0].length; columna++){

if(maximo < mat[fila][columna]){

maximo = mat[fila][columna];

f= fila;

}

}

}

return f;

}

}

**Ejercicio 6:**

**Escribir un método que dada una matriz de enteros devuelva la columna que contiene el máximo elemento de la matriz.**

public class Main

{

public static void main(String[] args) {

int matriz[][] = {

{5, 6, 90},

{11, -5, 0},

{33, 1, 2}

};

int columna = columnaDelMaximo(matriz);

System.out.print("La columna donde se encuentra el mayor es: "+ columna);

}

public static int columnaDelMaximo(int mat[][]){

int maximo= mat[0][0];

int c=0;

for(int fila=0; fila < mat.length ; fila++){

for(int columna=0; columna < mat[0].length; columna++){

if(maximo < mat[fila][columna]){

maximo = mat[fila][columna];

c= columna;

}

}

}

return c;

}

}

**Ejercicio 7:**

**Escribir un método que dada una matriz de enteros devuelva la posición que contiene el máximo elemento de la matriz.**

public class Main

{

public static void main(String[] args)

{

int matriz[][] =

{

{5, 6, 90},

{11, 5, 0},

{33, 1, 2}

};

int posicion = posicionDelMaximo(matriz);

System.out.print("La posicion donde se encuentra el mayor en la matriz es: "+ posicion);

}

public static int posicionDelMaximo(int mat[][])

{

int posMaxi= mat[0][0];

int posicion=0;

for(int fila=0; fila < mat.length ; fila++)

{

for(int columna=0; columna < mat[0].length; columna++)

{

if(posMaxi < mat[fila][columna]){

posMaxi = mat[posicion];

posicion= posMaxi;

}

}

}

return posicion;

}

}

**Ejercicio 8:**

**Escribir un método que dada una matriz de enteros y un entero, devuelva la posición de la matriz en la que se encuentra ese entero, o un valor que indique que el entero no se encuentra en la matriz**.

public class Main

{

public static void main(String[] args) {

int matriz[][] = {

{5, 6, 90},

{11, -5, 100},

{333, 1, 2}

};

int entero=333;

int posicion[] = posicionDelMaximo(matriz,entero);

if(posicion[0] == -1){

System.out.println("El número ingresado no se encuentra en la matris");

}

else{

for(int i=0; i<posicion.length; i++){

System.out.println("La posicion donde se encuentra el numero ingresado es: "+ posicion[i]);

}

}

}

public static int[] posicionDelMaximo(int mat[][], int num){

//int maximo= mat[0][0];

int p[]=new int[2];

p[0]=-1;

for(int fila=0; fila < mat.length ; fila++){

for(int columna=0; columna < mat[0].length; columna++){

if(num == mat[fila][columna]){

p[0]=fila;

p[1]=columna;

}

}

}

return p;

}

}

**Ejercicio 9:**

**Escribir un método que dada una matriz de enteros y un entero que representa un índice de fila, devuelva la suma de los elementos de esa fila.**  
public class Main

{

public static void main(String[] args) {

int matriz[][] = {

{5, 6, 90},

{11, -5, 100},

{333, 1, 2}

};

int fila2=1;

int suma = sumarMatriz1(matriz, fila2);

System.out.println("La suma de los enteros de la fila "+fila2+" es: "+suma);

}

public static int sumarMatriz1(int mat[][], int fila2){

int sumar=0;

for(int columna=0; columna < mat[0].length; columna++){

sumar = sumar + mat[fila2][columna];

}

return sumar;

}

}

**Ejercicio 10:**

**Escribir un método que dada una matriz de enteros y un entero que representa un índice de columna, devuelva la suma de los elementos de esa columna.**  
public class Main

{

public static void main(String[] args) {

int matriz[][] = {

{5, 6, 90},

{11, -5, 100},

{333, 1, 2}

};

int columna=0;

int suma = sumarMatriz1(matriz, columna);

System.out.println("La suma de los enteros de la columna "+columna+" es: "+suma);

}

public static int sumarMatriz1(int mat[][], int colum){

int sumar=0;

for(int fila=0; fila < mat[0].length; fila++){

sumar = sumar + mat[fila][colum];

}

return sumar;

}

}

**Ejercicio 11:**

**Escribir un método que dada una matriz de enteros y dos enteros que representan índices de fila, intercambie las filas correspondientes de la matriz.**

Public class Main{

public static void main(String[] args) {

//Matriz de ejemplo

int[][] matriz = {

{1, 2, 3},

{0, 0, 0},

{7, 8, 9}

};

//intercambiamos las filas

int fila1 = 0;

int fila2 = 1;

// para intercambiar las filas

int m[][] = intercambioFilas(matriz, fila1, fila2);

for (int fila= 0; fila< m.length; fila++){

for ( int columnas=0; columnas< m[0].length; columnas++){

System.out.print(m[fila][columnas]);

}

System.out.println(" ");

}

}

public static int[][] intercambioFilas (int[][] matriz,int fila1,int fila2){

int aux;

//recorre las columnas

for(int columna=0; columna< matriz[0].length; columna++)

{

//izquierda contenedor(donde quiero guardar)y el = es que quiero guardar

aux = matriz[fila1][columna];

matriz[fila1][columna] = matriz[fila2][columna];

matriz[fila2][columna] = aux;

}

return matriz;

}

}

**Ejercicio 12:**

**Escribir un método que dada una matriz de enteros y dos enteros que representan índices de columna, intercambie las columnas correspondientes de la matriz.**

public class Main {

public static void main(String[] args) {

int[][] matriz = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

int columna1 = 0;

int columna2 = 2;

if (columna1 >= 0 && columna1 < matriz[0].length && columna2 >= 0 && columna2 < matriz[0].length) {

int[][] m = intercambioColumnas(matriz, columna1, columna2);

for (int fila = 0; fila < m.length; fila++) {

for (int columna = 0; columna < m[0].length; columna++) {

System.out.print(m[fila][columna] + " ");

}

System.out.println();

}

} else {

System.out.println("Los índices de columna proporcionados son inválidos.");

}

}

public static int[][] intercambioColumnas(int[][] matriz, int columna1, int columna2) {

int aux;

for (int fila = 0; fila < matriz.length; fila++) {

aux = matriz[fila][columna1];

matriz[fila][columna1] = matriz[fila][columna2];

matriz[fila][columna2] = aux;

}

return matriz;

}

}

**Ejercicio 13:**

**Escribir un método que dada una matriz de enteros rote la matriz a derecha. Se define rotar la matriz a derecha como la operación que mueve cada uno de los elementos de la matriz una posición a la derecha. En caso de que un elemento al ser movido a la derecha no tenga lugar en la fila en la que se encontraba originalmente, el mismo debe ser puesto en la primera posición de la siguiente fila. En caso que el elemento que no tiene lugar en la fila se encontrara en la última fila, el mismo debe ser llevado a la primera posición de la primera fila.**

public class Main

{

public static void main(String[] args) {

//Matriz de ejemplo

int m[][] = {

{1, 2, 3},

{4, 6, 2},

{7, 8, 9}

};

m = matrizDerecha(m);

for(int fila=0; fila < m.length; fila++){

for(int columna=0; columna < m[0].length; columna++){

System.out.print(m[fila][columna] + " ,");

}

System.out.println(" ");

}

}

public static int[][] matrizDerecha(int matriz[][]){

int save1 = matriz[0][0], siguiente;

for(int fila=0; fila < matriz.length; fila++){

for(int columna=0; columna < matriz[0].length; columna++){

if(matriz.length-1 == fila && matriz[0].length-1 == columna){

matriz[0][0] = save1; // 9

}

else{

if(columna == matriz[0].length-1){

siguiente = matriz[fila+1][0]; // 4

matriz[fila+1][0] = save1; // 3

save1 = siguiente;

} else{

siguiente = matriz[fila][columna+1]; //

matriz[fila][columna+1] = save1;

save1=siguiente;

}

}

}

}

return matriz;

}

}

**Ejercicio 14:**

**Escribir un método que dada una matriz de enteros rote la matriz a izquierda. Se define rotar la matriz a izquierda como la operación que mueve cada uno de los elementos de la matriz una posición a la izquierda. En caso de que un elemento al ser movido a la izquierda no tenga lugar en la fila en la que se encontraba originalmente, el mismo debe ser puesto en la última posición de la fila anterior. En caso que el elemento que no tiene lugar en la fila se encontrara en la primera fila, el mismo debe ser llevado a la última posición de la última fila.**public class Main

{

public static void main(String[] args) {

//Matriz de ejemplo

int m[][] = {

{1, 2, 3},

{4, 6, 2},

{7, 8, 9}

};

m = matrizDerecha(m);

for(int fila=0; fila < m.length; fila++){

for(int columna=0; columna < m[0].length; columna++){

System.out.print(m[fila][columna] + " ,");

}

System.out.println(" ");

}

}

public static int[][] matrizDerecha(int matriz[][]){

int save1 = matriz[2][2];

int anterior;

for(int fila=2; fila >= 0; fila--){

for(int columna=2; columna >= 0; columna--){

if(0 == fila && 0 == columna){

matriz[2][2] = save1;

}

else{

if(columna == 0){

anterior = matriz[fila-1][columna+2];

matriz[fila-1][columna+2] = save1;

save1 = anterior;

} else{

anterior = matriz[fila][columna-1];

matriz[fila][columna-1] = save1;

save1=anterior;

}

}

}

}

return matriz;

}

}

**Ejercicio 15:**

**Escribir un método que dadas dos matrices de dobles devuelva una nueva matriz que**

**represente la suma de las matrices que se le pasaron.**

public class Main{

public static void main(String[] args) {

double[][] matriz1 = {{1.5, 2.0, 3.0}, {4.0, 5.5, 6.0}};

double[][] matriz2 = {{0.5, 1.0, 2.0}, {3.0, 2.5, 1.0}};

double[][] resultado = sumarMatrices(matriz1, matriz2);

System.out.println("La matriz suma es:");

imprimirMatriz(resultado);

}

public static double[][] sumarMatrices(double[][] matriz1, double[][] matriz2) {

int filas = matriz1.length;

int columnas = matriz1[0].length;

double[][] resultado = new double[filas][columnas];

for (int i = 0; i < filas; i++) {

for (int j = 0; j < columnas; j++) {

resultado[i][j] = matriz1[i][j] + matriz2[i][j];

}

}

return resultado;

}

public static void imprimirMatriz(double[][] matriz) {

for (int i = 0; i < matriz.length; i++) {

for (int j = 0; j < matriz[0].length; j++) {

System.out.print(matriz[i][j] + " ");

}

System.out.println();

}

}

}

**Ejercicio 16:**

**Escribir un método que dadas dos matrices de dobles devuelva una nueva matriz que**

**represente la resta de las matrices que se le pasaron.**public class Main {

public static void main(String[] args) {

int[][] matriz1 = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

int [][] matriz2 = {

{1, 1, 1},

{1, 1, 1},

{1, 1, 1}

};

//imprime las matrices

int[][] resultado = RestaMatrices(matriz1, matriz2);

imprimirMatriz(resultado);

}

//siempre se pone doble [][] paraa que sepa que es una matriz

public static int[][] RestaMatrices(int[][] matriz1, int[][] matriz2) {

int[][] resultado = new int[matriz1.length][matriz1[0].length];

// Sumar las matrices

for (int i = 0; i < resultado.length; i++) {

for (int j = 0; j < resultado[0].length; j++) {

resultado[i][j] = matriz1[i][j] - matriz2[i][j];

}

}

return resultado;

}

//utilizarla

public static void imprimirMatriz(int[][] matriz) {

for (int i = 0; i < matriz.length; i++) {

for (int j = 0; j < matriz[0].length; j++) {

System.out.print(matriz[i][j] + " ");

}

System.out.println();

}

}

}

**Ejercicio 17:**

**Escribir un método que dadas dos matrices de dobles devuelva una nueva matriz que**

**represente el producto de las matrices que se le pasaron.**

public class Main {

public static void main(String[] args) {

int[][] matriz1 = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

int [][] matriz2 = {

{2, 1, 1},

{2, 10, 2},

{2, 2, 2}

};

//imprime las matrices

int[][] resultado = MultiplicaciondeMatrices(matriz1, matriz2);

imprimirMatriz(resultado);

}

//siempre se pone doble [][] paraa que sepa que es una matriz

public static int[][] MultiplicaciondeMatrices(int[][] matriz1, int[][] matriz2) {

int[][] resultado = new int[matriz1.length][matriz1[0].length];

// Sumar las matrices

for (int i = 0; i < resultado.length; i++) {

for (int j = 0; j < resultado[0].length; j++) {

resultado[i][j] = matriz1[i][j] \* matriz2[i][j];

}

}

return resultado;

}

//utilizarla

public static void imprimirMatriz(int[][] matriz) {

for (int i = 0; i < matriz.length; i++) {

for (int j = 0; j < matriz[0].length; j++) {

System.out.print(matriz[i][j] + " ");

}

System.out.println();

}

}

}

**Ejercicio 18:**

**Escribir un método que dada una matriz de dobles transponga la matriz.**public class Main {

public static void main(String[] args) {

double[][] matrizOriginal = {{1.0, 2.0, 3.0}, {4.0, 5.0, 6.0}, {7.0, 8.0, 9.0}};

double[][] matrizTranspuesta = transponerMatriz(matrizOriginal);

for (int i = 0; i < matrizTranspuesta.length; i++) {

for (int j = 0; j < matrizTranspuesta[0].length; j++) {

System.out.print(matrizTranspuesta[i][j] + " ");

}

System.out.println();

}

}

public static double[][] transponerMatriz(double[][] matriz) {

int filas = matriz.length;

int columnas = matriz[0].length;

double[][] matrizTranspuesta = new double[columnas][filas];

for (int i = 0; i < filas; i++) {

for (int j = 0; j < columnas; j++) {

matrizTranspuesta[j][i] = matriz[i][j];

}

}

return matrizTranspuesta;

}

}

**Ejercicio 19:**

**Escribir un método que dada una matriz calcule su inversa.**