



Universidad de Costa Rica
Escuela de Ciencias de la Computación e Informática

Práctica de matrices

1. Programe un método llamado imprimir matriz que recibirá por parámetro una matriz de números enteros. (Deberá ser capaz de imprimir matrices de cualquier tamaño).
2. Cree un método capaz de elevar celdas de una matriz al cuadrado. El método deberá recibir una matriz de tamaño variable y elevar cada una de sus celdas al cuadrado.

El método deberá recibir una matriz por parámetro y retornar otra con los valores elevados a cuadrado. Por ejemplo:

Recibe:

1	2
3	4

Retorna:

1	4
9	16

3. Cree un método llamado sumarMatrices que intentará sumar una matrizA con una matrizB. Tome en cuenta que debe validar los tamaños de las matrices, en caso contrario no se podrán sumar y retornará los resultados. (Sumar matrices quiere decir sumar cada entrada de la matrizA con su entrada correspondiente en la matrizB)
4. Programe un método que reciba por parámetro una matriz de números enteros y retorne un arreglo de dos campos que representará la fila y la columna (respectivamente) en donde se encuentra el menor de los elementos del mismo.

Por ejemplo, si se recibe por parámetro la siguiente matriz:

14	9	8	7
21	11	1	23
11	21	19	12

El método retornará un arreglo de la forma:

1	2
---	---

Que representa la celda (en formato de fila y columna) en donde se encuentra ubicado el mayor de los elementos de la matriz (en este caso, el 1).

5. Programe un método que cree una submatriz a partir de una matriz recibida por parámetro. El método deberá recibir por parámetro cuatro números enteros que corresponderán a los límites de la submatriz que se desea crear (desde que fila y columna, hasta que fila y columna inclusive se desea crear la matriz). El método deberá retornar la submatriz creada.

Por ejemplo, en caso de recibir la siguiente matriz:

1	2	3	4	5
6	7	8	9	0
2	3	4	5	6
7	8	9	0	1
4	5	6	7	8

Junto con los parámetros desde fila: 1, columna: 2, hasta fila: 4, columna 3, el método retornará:

8	9
4	5
9	0
6	7

Cabe resaltar, que el método deberá validar que los límites recibidos por parámetro sean válidos y evitar que el programa se caiga. En caso de recibir posiciones inválidas, el método deberá retornar null.

Implemente una clase de prueba capaz de corroborar el funcionamiento correcto de cada uno de los métodos de la clase anterior.