

## **ARRAYS DE VARIAS DIMENSIONES**

**Ejercicio U4\_B4A\_E1:** Volver a escribir el ejemplo anterior utilizando el atributo length, lo que permite utilizar de nuevo un bucle anidado en otro y hacer un código más compacto.

El atributo length nos permite condensar de nuevo el código utilizando un bucle anidado en otro.

```
class Unidad4{
    public static void main(String[] args){
        int[][] arrayDosD=new int[3][];
        arrayDosD[0]=new int[4];
        arrayDosD[1]=new int[2];
        arrayDosD[2]=new int[3];
        for(int i=0;i<arrayDosD.length;i++){
            System.out.println("cargamos e imprimimos arrayDosD["+i+"]. Observa que su tamaño es "+arrayDosD[i].length);
            for(int j=0;j<arrayDosD[i].length;j++){
                arrayDosD[i][j]=i*j + i + j*2; //cargamos la matriz
                System.out.println("arrayDosD["+ i + "]["+ j +"]="+ arrayDosD[i][j]);
            }
        }
    }
}
```

### **Ejercicio U4\_B4A\_E2:**

```
class Unidad4 {
    public static void main(String[] args){

        int[][] matriz= {
            {11,12,13,14},
            {21,22},
            {31,32,33}
        };

        for(int i=0;i<matriz.length;i++){
            for(int j=0;j<matriz[i].length;j++){
                System.out.println("matriz["+i+"]"+"["+j+"]= "+matriz[i][j]);
            }
        }
    }
}
```

### **Ejercicio U4\_B4A\_E3:**

```
class Unidad4 {
    public static void main(String[] args){

        int[][] matriz= {{11,12,13,14},{21,22},{31,32,33}};

        for(int[] arrayInt:matriz){
            for(int i:arrayInt){
                System.out.print(i+ " ");
            }
        }
    }
}
```

```

    }
    System.out.println("");
}

}
}

```

Observa que no puedo referirme a los índices de un elemento, es decir, no puedo escribir como en el ejercicio anterior

```
System.out.println("arrayDosD["+ i + "]["+ j +"]="+ arrayDosD[i][j]);
```

A no ser que declare variables para llevar la cuenta y entonces no compensa esta sintaxis.

### Ejercicio U4\_B4A\_E4:

Observa:

- al imprimir la matriz como no me hace falta hacer ninguna operación con los índices, se utiliza el for mejorado que es más fácil!
- A pesar de lo comentado en el ejemplo del break, aquí como es un ejercicio de matrices cuadradas "quizá" es más claro trabajar con una variable tam para especificar el tamaño de la matriz.
- Las matrices bidimensionales casi siempre implican un bucle anidado pero no siempre. Observa que el intercambio de valores de diagonales se hace en bucle simple. Observa también que el punto central es común a las dos diagonales pero no pasa nada con intercambiarlo consigo mismo pues deja el resultado invariante.

La relación de índices para conseguir la nueva matriz es la siguiente

Para cada fila i  
 $matriz[i][i] \leftrightarrow matriz[i][tam-1-i];$

```

import java.util.Random;
class Unidad4{
    static void imprimirMatriz(int[][] m){
        for(int[] fila:m){
            for(int elemento:fila){
                System.out.print(elemento+"\t");
            }
            System.out.println();
        }
        System.out.println("\n");
    }
}

public static void main (String[]args){

    Random r= new Random();
    int[][] matriz;
    int tam=Integer.parseInt(args[0]);

    //Creamos array y se rellenan aleatoriamente
    matriz= new int[tam][tam];
    for(int fila=0;fila<tam;fila++){
        for(int columna=0;columna<tam;columna++){
            matriz[fila][columna]=r.nextInt(100);
        }
    }
    //listamos matriz para comprobar
    imprimirMatriz(matriz);
    //Intercambio de diagonales
    for(int i=0;i<tam;i++){
        int temp=matriz[i][i];
        matriz[i][i]=matriz[i][tam-1-i];
        matriz[i][tam-1-i]=temp;
    }
}

```

```

        //imprimimos resultado
        imprimirMatriz(matriz);
    }
}

```

### Ejercicio U4\_B4A\_E5:

La relación de índices es diferente para cada diagonal, además hay que tener en cuenta que el proceso debe parar al llegar a la mitad de la matriz ya que si no vuelvo a cambiar los valores y dejaría la matriz como estaba.

Para cada fila  $i$  hasta que  $i < \text{tam}/2$ :

En diagonal principal  $\text{matriz}[i][i] \leftrightarrow \text{matriz}[\text{tam}-1-i][\text{tam}-1-i]$ ;

En diagonal secundaria  $\text{matriz}[i][\text{tam}-1-i] \leftrightarrow \text{matriz}[\text{tam}-1-i][i]$ ;

```

import java.util.Random;
class Unidad4{
    static void imprimirMatriz(int[][] m){
        for(int[] fila:m){
            for(int elemento:fila){
                System.out.print(elemento+"\t");
            }
            System.out.println();
        }
        System.out.println("\n");
    }

    public static void main (String[]args){

        Random r= new Random();
        int[][] matriz;
        int tam=Integer.parseInt(args[0]);

        //Creamos array y se rellenan aleatoriamente
        matriz= new int[tam][tam];
        for(int fila=0;fila<tam;fila++){
            for(int columna=0;columna<tam;columna++){
                matriz[fila][columna]=r.nextInt(100);
            }
        }
        //listamos matriz para comprobar
        imprimirMatriz(matriz);
        //Intercambio de diagonales
        for(int i=0;i<tam/2;i++){
            //intercambio valores diagonal principal
            int temp=matriz[i][i];
            matriz[i][i]=matriz[tam-1-i][tam-1-i];
            matriz[tam-1-i][tam-1-i]=temp;
            //intercambio valores diagonal secundaria
            temp=matriz[i][tam-1-i];
            matriz[i][tam-1-i]=matriz[tam-1-i][i];
            matriz[tam-1-i][i]=temp;
        }

        //imprimimos resultado
        imprimirMatriz(matriz);
    }
}

```

### Ejercicio U4\_B4A\_E6:

```
static void imprimirConteos(int[][] c){
    for (int mes = 0; mes < c.length; mes++) {
        System.out.println("Mes" + (mes+1));
        System.out.println("-----");
        for (int dia = 0; dia < c[mes].length; dia++) {
            System.out.print("\tdía" + (dia+1) + ":\t");
            for(int hora=0; hora<c[mes][dia].length; hora++){
                System.out.print("\t" + c[mes][dia][hora]);
            }
            System.out.println();
        }
    }
}
```

Ojo según se coloquen los tabuladores puedo tener efectos indeseados en la impresión

### Ejercicio U4\_B4A\_E7: Sin solución. Es para entregar.

### Ejercicio U4\_B4A\_E8: Matriz identidad acepta el reto

<https://www.aceptaelreto.com/problem/statement.php?id=151&cat=14>

#### Una solución de tantas

```
import java.util.Scanner;
public class Unidad4 {
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        int tam=sc.nextInt();
        while(tam!=0){
            boolean esIdentidad=true;
            //tengo que hacer tam*tam lecturas
            for(int i =0;i<tam;i++) {
                for(int j=0;j<tam;j++){
                    int elemento=sc.nextInt();
                    if(i==j && elemento!=1 || i!=j&&elemento!=0){
                        esIdentidad=false;
                        //break; ¡MAL! HAY QUE SEGUIR LEYENDO hasta acabar matriz
                    }
                }
            }
            if(esIdentidad){
                System.out.println("SI");
            }else{
                System.out.println("NO");
            }
            //siguiente matriz
            tam=sc.nextInt();
        }
    }
}
```

A muchos alumnos le daba wrong answer porque sólo comprobaban los valores de la diagonal. Con los casos de prueba del ejemplo basta con comprobar la diagonal, pero luego el juez manda otros casos que sí precisan comprobar lo de fuera de la diagonal, por ejemplo

```
1 0 5
0 1 0
0 0 1
```

Si sólo miro la diagonal me da SÍ pero si miro todo me da NO

**Ejercicio U4\_B4A\_E9:** Acepta el reto. Matrices Triangulares id 160  
<https://www.aceptaelreto.com/problem/statement.php?id=160>

```
import java.util.Scanner;

public class Unidad4 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int dimension = sc.nextInt();
        while (dimension != 0) {
            boolean arriba=true;
            boolean abajo=true;
            int elemento=0;
            for(int i=0;i<dimension;i++){
                for(int j=0;j<dimension;j++){
                    elemento=sc.nextInt();
                    if(j>i && elemento!=0){
                        arriba=false;
                    }
                    if(j<i && elemento!=0){
                        abajo=false;
                    }
                }
            }
            System.out.println(abajo||arriba?"SI":"NO");
            dimension = sc.nextInt();
        }
    }
}
```