Arreglos

Vectores o arreglos unidimensionales

Elizabeth León Guzmán, Ph.D.
eleonguz@unal.edu.co

Jonatan Gómez Perdomo, Ph. D.
jgomezpe@unal.edu.co

Arles Rodríguez, Ph.D.
aerodriguezp@unal.edu.co

Camilo Cubides, Ph.D. (c)
eccubidesg@unal.edu.co

Carlos Andres Sierra, M.Sc.
casierrav@unal.edu.co

Research Group on Data Mining – Grupo de Investigación en Minería de Datos – (Midas)
Research Group on Artificial Life – Grupo de Investigación en Vida Artificial – (Alife)
Computer and System Department
Engineering School
Universidad Nacional de Colombia

Agenda

Agenda

1. Arreglos (Vectores)
   - Conceptos
   - Arreglos en Java
   - Funciones para utilizar arreglos
2. Matrices
   - Conceptos
   - Matrices en Java
   - Ejemplos
3. Ejercicios y problemas

Definición

Un **arreglo** o **vector** es una secuencia de *n* elementos del mismo tipo, en donde el valor de *n* se fija al crear el arreglo. A los elementos de un arreglo se les llama **componentes** del arreglo.

Ejemplo

- `[]`: Arreglo vacio.
- `[1, 0, 7, -2, 8]`: Un arreglo de enteros de 5 componentes.
- `["Radio", "Lado"]`: Un arreglo de cadenas de 2 componentes.
- `[1.3, 2.4, -3.0, 4.5]`: Un arreglo de reales de 4 componentes.
- `[true, false, false]`: Un arreglo de booleanos de 3 componentes.

Notación I

En general, un arreglo v se puede representar de la siguiente forma

$$v = [v_0, v_1, v_2, \ldots, v_{n-1}]$$

donde el arreglo está constituido por n componentes de un conjunto genérico $\mathcal{V}$. Si $v \in \mathcal{V}^n$, entonces el arreglo se dice que es n-dimensional o de tamaño n.

Ejemplo

El siguiente arreglo de tamaño 6 pertenece a $\mathbb{Z}^6$ y tiene una notación particular la cual es $0^6 = [0, 0, 0, 0, 0, 0]$

Notación II

En un vector, para referirse a una componente en particular, a ésta se le dice que es la *componente* en la posición i o la i-ésima componente, esto significa que el objeto es la componente ubicada en la posición i, se denota por la expresión v_i y se puede ubicar dentro del vector como se muestra a continuación

$$\begin{array}{c}\text{componente } i\text{-ésima}\\ \downarrow \\ {}[v_0, \ldots, v_i, \ldots, v_{n-1}]\end{array}$$

Notación III

Ejemplo

Para el vector

$$v = \left[-1,\ \tfrac{3}{4},\ -0.25,\ -\tfrac{1}{5},\ \sqrt{2},\ 0.0,\ \pi,\ \sqrt[3]{5},\ 0.\overline{9} \right]$$

de $\mathbb{R}^9$ se tiene que sus componentes son:

- $v_0 = -1$.
- $v_1 = \frac{3}{4}$.
- $v_2 = -0.25$.
- $v_3 = -\frac{1}{5}$.
- $v_4 = \sqrt{2}$.
- $v_5 = 0.0$.
- $v_6 = \pi$.
- $v_7 = \sqrt[3]{5}$.
- $v_8 = 0.\overline{9}$.

El conjunto de los arreglos

A partir de la notación de producto generalizado de un conjunto $\mathcal{V}$

$$\mathcal{V}^n = \underbrace{\mathcal{V} \times \mathcal{V} \times \mathcal{V} \times \cdots \times \mathcal{V}}_{n\text{-veces}}$$

se puede obtener el conjunto de los arreglos (vectores) $\mathcal{V}^*$, el cual se define como la unión de todos los productos cartesianos del conjunto $\mathcal{V}$, de la siguiente manera

$$\mathcal{V}^* = \bigcup_{n \in \mathbb{N}} \mathcal{V}^n$$

Agenda

1. Arreglos (Vectores)
   - Conceptos
   - Arreglos en Java
   - Funciones para utilizar arreglos
2. Matrices
   - Conceptos
   - Matrices en Java
   - Ejemplos
3. Ejercicios y problemas

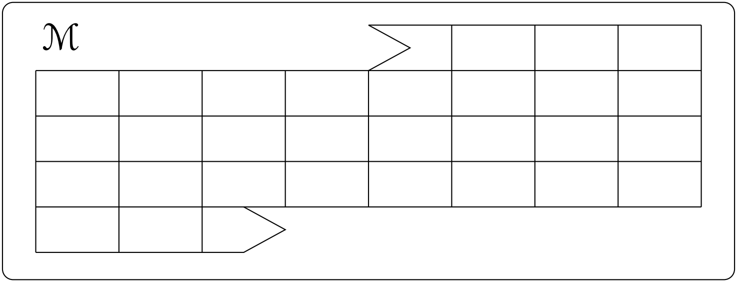
Definición de Arreglos en Java

Si se quiere expresar en Java que $\mathtt{x} \in \mathbb{T}[]$ esto se escribe como $\mathbb{T}$`[] x;`
Para reservar el espacio de memoria para todo el arreglo de tipo $\mathbb{T}$, se escribe `x = new` $\mathbb{T}$`[n]`.
De esta manera, para crear un arreglo `x` de tamaño n y de tipo $\mathbb{T}$ se utiliza la instrucción

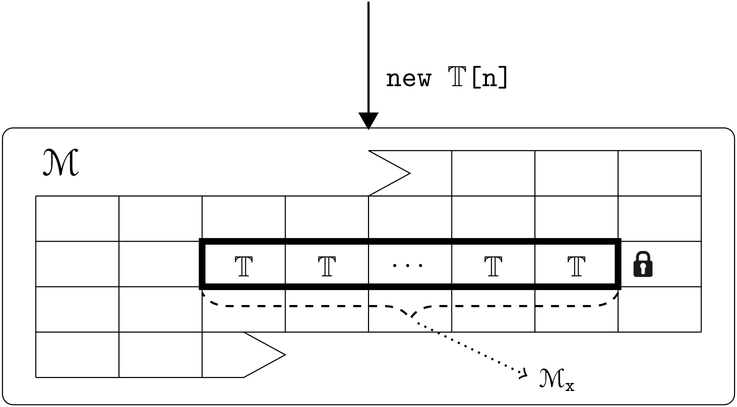
$$\boxed{\mathbb{T}\texttt{[] x = new } \mathbb{T}\texttt{[n];}}$$

Manejo de la Memoria I

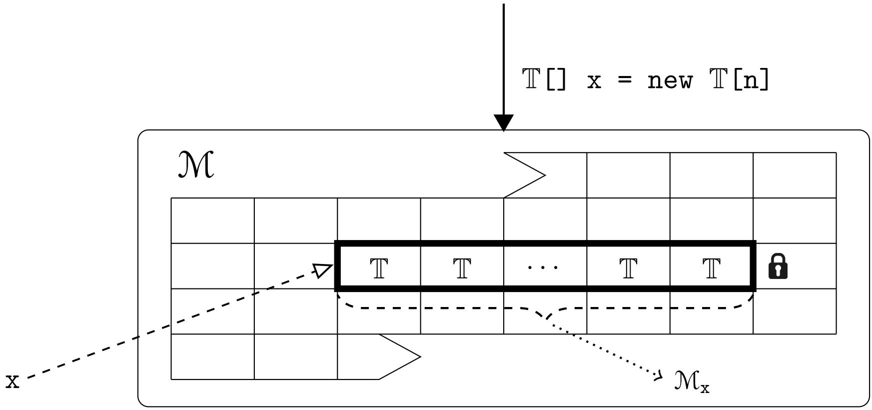
Con la instrucción $\mathbb{T}$`[] x = new` $\mathbb{T}$`[n]` se reserva una porción de la memoria $\mathcal{M}$ que es utilizada para almacenar el arreglo y que es bloqueada para que sólo se pueda utilizar para guardar valores en el arreglo. A la porción de espacio de memoria que ocupa un arreglo `x` lo notaremos como $\mathcal{M}_{\mathtt{x}}$. Gráficamente esto proceso se puede ver de la siguiente manera:



Manejo de la Memoria II



Manejo de la Memoria III



Arreglos y subíndices I

Al definir un arreglo, se construye un conjunto de variables del mismo tipo, variables que se encuentran subindicadas, esto es, se accede a ellas por medio de un índice que específica una componente particular del arreglo.

En lenguajes como C++ y Java, es necesario tener en cuenta que la primera componente del arreglo está ubicada en la posición 0, y para un arreglo de tamaño n se tiene que la última componente del arreglo estará ubicada en la posición $n - 1$.

En caso de que se quiera acceder a una componente negativa o a una mayor o igual a la n-ésima en Java el programa será interrumpido mostrando al usuario una excepción de tipo `java.lang.ArrayIndexOutOfBoundsException`.

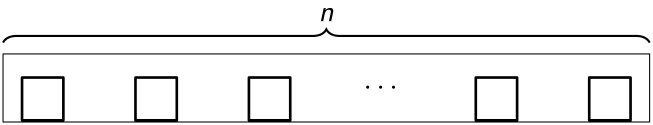
Arreglos y subíndices II

A partir de lo anterior se tiene que dado un arreglo $\mathtt{x} \in \mathbb{T}[]$, para acceder en C++ o Java a la variable almacenada en la componente i se utiliza la equivalencia

$$\mathtt{x}_i \equiv \mathtt{x[i-1]}$$

Arreglos y subíndices III

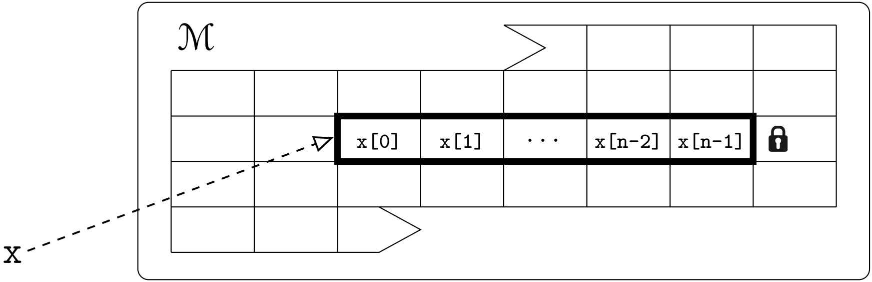
es decir, para un vector $x = (x_1, x_2, x_3, \ldots, x_{n-1}, x_n)$ la representación del arreglo en Java es la que se presenta en la siguiente figura.



$$
\begin{array}{ccccccccccc}
\uparrow & & \uparrow & & \uparrow & & \cdots & & \uparrow & & \uparrow \\
\big[\ \texttt{x[0]} & , & \texttt{x[1]} & , & \texttt{x[2]} & , & \ldots & , & \texttt{x[n-2]} & , & \texttt{x[n-1]}\ \big] \\
\uparrow & & \uparrow & & \uparrow & & \cdots & & \uparrow & & \uparrow \\
(\ x_1 & , & x_2 & , & x_3 & , & \ldots & , & x_{n-1} & , & x_n\ )
\end{array}
$$

Arreglos y subíndices IV

con lo cual, la representación en la memoria resultará ser



Arreglos en otros lenguajes

En lenguajes como SciLab y MatLab, la primera componente del arreglo está ubicada en la posición 1, y para un arreglo de tamaño n se tiene que la última componente del arreglo estará ubicada en la posición n, por lo que las posiciones se manejan como en notación matemática de vectores, es decir, para acceder en SciLab o MatLab a la variable almacenada en la componente i se utiliza la equivalencia

$$\mathtt{x}_i \equiv \mathtt{x[i]}$$

Agenda

1. Arreglos (Vectores)
   - Conceptos
   - Arreglos en Java
   - Funciones para utilizar arreglos
2. Matrices
   - Conceptos
   - Matrices en Java
   - Ejemplos
3. Ejercicios y problemas

Lectura de un arreglo

Se pueden combinar las operaciones de lectura de un tipo de dato primitivo con la instrucción `for` para crear una función que reciba el Scanner (de donde se lee) y el tamaño del arreglo a leer y lea el arreglo.

Ejemplo

En este ejemplo leeremos un arreglo de enteros:

```
public static int lee_arreglo_enteros(Scanner sc, int n){
  int[] x = new int[n];
  for( int i=0; i<n; i++ ){
    System.out.println("Componente "+i+"-ésima?");
    x[i] = sc.nextInt();
  }
  return x;
}
```

Escritura de un arreglo

Se puede usar la instrucción `for` para crear una función que reciba el arreglo a escribir y lo escriba.

Ejemplo

En este ejemplo escribiremos un arreglo de reales:

```
public static void escribe_arreglo_reales(double[] x){
  int n = x.length;
  for( int i=0; i<n; i++ ){
    System.out.println("x["+i+"]="+x[i]);
  }
}
```

Suma de las componentes de un arreglo I

Si se tiene un arreglo de números (por ejemplo de reales), se puede definir la función `suma_arreglo_reales` para calcular la suma de todos ellos:

Ejemplo

```
public static double suma_arreglo_reales(double[] A){
  double s = 0;
  for( int i=0; i<A.length; i++){
    s += A[i];
  }
  return s;
}
```

Suma de las componentes de un arreglo II

Se puede usar la instrucción `for` y ver el arreglo como una colección (tema que se verá más adelante) para usar la versión iterable y definir la función `suma_arreglo_reales` para calcular la suma de todos ellos:

Ejemplo

```
public static double suma_arreglo_reales(double[] A){
  double s = 0;
  for( double x:A ){
    s += x;
  }
  return s;
}
```

Posición del máximo de un arreglo

Si se tiene un arreglo de números, se puede definir la función pos_maximo para calcular la posición donde se encuentra el máximo de ellos (el arreglo debe tener al menos una componente):

Ejemplo

```
public static int pos_maximo(int[] A){
   int m = 0;
   for(int i=0; i<A.length; i++){
      if ( A[i] > A[m] )
         m = i;
   }
   return m;
}
```

Agenda

Agenda

1. Arreglos (Vectores)
   - Conceptos
   - Arreglos en Java
   - Funciones para utilizar arreglos
2. Matrices
   - Conceptos
   - Matrices en Java
   - Ejemplos
3. Ejercicios y problemas

Definición

Una **matriz** es un arreglo rectangular de elementos del mismo tipo. En este curso usaremos arreglos de arreglos para representar matrices.

Ejemplo

La siguiente estructura rectangular de números enteros denotada por la expresión X es una matriz

$$X = \begin{bmatrix} 1 & 3 & 7 & -2 & 8 \\ 9 & 11 & 5 & 6 & 4 \\ 6 & -2 & -1 & 1 & 1 \end{bmatrix}$$

Notación I

En general, una matriz X se puede representar de la siguiente manera

$$X = \begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} & \cdots & x_{0,(m-1)} \\ x_{1,0} & x_{1,1} & x_{1,2} & \cdots & x_{1,(m-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{(n-1),0} & x_{(n-1),1} & x_{(n-1),2} & \cdots & x_{(n-1),(m-1)} \end{bmatrix}$$

donde la matriz está compuesta por n **filas** y m **columnas**, a esta matriz se le dice que es de **tamaño** $n \times m$.

Notación II

A los objetos de la matriz se les llaman **componentes** o **entradas** de la matriz, y para referirse a una componente en particular, a ésta se le dice que es la **componente** en la posición (i, j), esto significa que la componente ubicada en la fila i y en la columna j, se denota por la expresión $X_{i,j}$ y se puede ubicar dentro de la matriz como se muestra a continuación

$$\begin{array}{cc}
 & \text{columna } j \\
 & \downarrow \\
\text{fila } i \longrightarrow &
\begin{bmatrix}
x_{0,0} & \cdots & x_{0,j} & \cdots & x_{0,(m-1)} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
x_{i,0} & \cdots & \boxed{x_{i,j}} & \cdots & x_{i,(m-1)} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
x_{(n-1),0} & \cdots & x_{(n-1),j} & \cdots & x_{(n-1),(m-1)}
\end{bmatrix}
\end{array}$$

Notación III

Para la matriz

$$X = \begin{bmatrix} 2 & -1 & \frac{3}{4} & -0.1 & e \\ -\frac{1}{5} & \sqrt{2} & 4 & 0.0 & 6 \\ 3.14 & \pi & \frac{1}{2} & \sqrt{3} & 3 \\ \sqrt[3]{5} & -10 & 0 & -5 & 0.\overline{9} \end{bmatrix}$$

de tamaño 4×5 se tiene que sus componentes son:

- $X_{0,0} = 2$
- $X_{0,1} = -1$
- $X_{0,2} = \frac{3}{4}$
- $X_{0,3} = -0.1$
- $X_{0,4} = e$
- $X_{1,0} = -\frac{1}{5}$
- $X_{1,1} = \sqrt{2}$
- $X_{1,2} = 4$
- $X_{1,3} = 0.0$
- $X_{1,4} = 6$
- $X_{2,0} = 3.14$
- $X_{2,1} = \pi$
- $X_{2,2} = \frac{1}{2}$
- $X_{2,3} = \sqrt{3}$
- $X_{2,4} = 3$
- $X_{3,0} = \sqrt[3]{5}$
- $X_{3,1} = -10$
- $X_{3,2} = 0$
- $X_{3,3} = -5$
- $X_{3,4} = 0.\overline{9}$

Notación IV

A partir del concepto de arreglo, se puede ahora definir el *conjunto de las matrices* $\mathbb{T}^{**}$ como la unión de todos los productos cartesianos del conjunto de los arreglos del conjunto $\mathbb{T}$, de la siguiente manera

$$\mathbb{T}^{**} = \bigcup_{n \in \mathbb{N}} \left(\bigcup_{m \in \mathbb{N}} \mathbb{T}^m \right)^n$$

Un elemento genérico del conjunto $\mathbb{T}^{**}$ es de la forma $(\mathbb{T}^m)^n$, donde n es el número de filas y m es el número de columnas. Para abreviar, de aquí en adelante se utilizará la notación

$$(\mathbb{T}^m)^n \Leftrightarrow \mathbb{T}^{n \times m}.$$

Agenda

Matriz=Arreglo de Arreglos

En Java, una matriz de tamaño $n \times m$ es un arreglo de n arreglos (llamados **filas**) de m elementos cada uno, teniendo que todos los arreglos son del mismo tipo:

$$n\left\{ \underbrace{\begin{bmatrix} x_{0,0} & x_{0,1} & \cdots & x_{0,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{i,0} & x_{i,1} & \cdots & x_{i,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-1,0} & x_{n-1,1} & \cdots & x_{n-1,m-1} \end{bmatrix}}_{m} \right. = \underbrace{\begin{bmatrix} \begin{bmatrix} x_{0,0} & x_{0,1} & \cdots & x_{0,m-1} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} x_{i,0} & x_{i,1} & \cdots & x_{i,m-1} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} x_{n-1,0} & x_{n-1,1} & \cdots & x_{n-1,m-1} \end{bmatrix} \end{bmatrix}}_{1}$$

Definición de Matrices

En Java, a una matrix $\mathbb{T}^{n \times m}$ se le da espacio en memoria con la instrucción `new` $\mathbb{T}$`[n][m]`. Una variable x de tipo matriz se define con la instrucción $\mathbb{T}$`[][] x;`

Ejemplo

Los siguientes son ejemplos de definición y asignación de espacio en memoria de variables de tipo matriz:

```
char[][] c = char[10][20]; // matriz de 10x20 caracteres
int[][] A = int[5][4]; // matriz de 5x4 enteros
double[][] y = double[6][6]; // matriz de 6x6 reales
boolean[][] b = boolean[2][3]; // matriz de 2x3 booleanos
```

Agenda

Cuadrados de elementos de una matriz I

Una función general que permite construir una nueva matriz que contiene el cuadrado de cada componente de una matriz dada es

Ejemplo

$$\begin{aligned} \mathit{cuadrados\_matriz} : \mathbb{Z}^{n\times m} &\to \mathbb{Z}^{n\times m} \\ (X) &\mapsto Y, \quad \text{donde} \quad Y_{i,j} = X_{i,j}^2, \\ &\qquad\qquad\qquad \forall i = 0, 1, \ldots, n-1, \\ &\qquad\qquad\qquad \forall j = 0, 1, \ldots, m-1 \end{aligned}$$

Cuadrados de elementos de una matriz II

Una función general que permite construir una nueva matriz que contiene el cuadrado de cada componente de una matriz dada es

Ejemplo

```
public static int[][] cuadrados_matriz(int[][] X){
  int[][] Y = new int[X.length][X[0].length];
  for(int i=0; i<X.length; i++){
    for(int j=0; j<X[i].length; j++){
      Y[i][j] = X[i][j]*X[i][j];
    }
  }
  return Y;
}
```

Diagonal de una matriz cuadrada I

Una función general que permite obtener un arreglo con los elementos que están en la diagonal de una matriz cuadrada (mismo número de filas y columnas) es:

Ejemplo

$$\begin{aligned} \textit{diagonal\_matriz} : \mathbb{R}^{n\times n} &\to \mathbb{R}^{n} \\ (X) &\mapsto Y, \quad \text{donde} \quad Y_i = X_{i,i}, \\ &\qquad\qquad\qquad\quad \forall i = 0, 1, \ldots, n-1 \end{aligned}$$

Diagonal de una matriz cuadrada II

Una función general que permite obtener un arreglo con los elementos que están en la diagonal de una matriz cuadrada (mismo número de filas y columnas) es:

Ejemplo

```
public static double diagonal_matriz(double[][] X){
  double[] Y = new double[X.length];
  for( int i=0; i<X.length; i++){
    Y[i] = X[i][i];
  }
  return Y;
}
```

Simetría de una matriz I

Una función general que permite determinar si una matriz cuadrada de caracteres es simétrica o no, es:

Ejemplo

$$matriz\_simetrica : \mathrm{ASCII}^{n\times n} \to \mathbb{B}$$

$$(X) \mapsto \bigwedge_{\substack{i=0\\ j=i+1}}^{\substack{n-1\\ n-2}} X_{i,j} \equiv X_{j,i}$$

Simetría de una matriz II

Una función general que permite determinar si una matriz cuadrada de caracteres es simétrica o no, es:

Ejemplo

```
public static boolean matriz_simetrica(char[][] X){
  boolean bandera = true;
  for( int i=0; i<X.length-1; i++){
    for( int j=i+1; j<X.length; j++){
      bandera &= (X[i][j] == X[j][i]);
    }
  }
  return bandera;
}
```

Agenda

Problemas de arreglos I

Problemas

1. Desarrollar un algoritmo que calcule el promedio de un arreglo de reales.
2. Desarrollar un algoritmo que calcule el producto punto de dos arreglos de números enteros (reales) de igual tamaño. Sean $v = [v_0, v_1, \ldots, v_{n-1}]$ y $w = [w_0, w_1, \ldots, w_{n-1}]$ dos arreglos, el producto de v y w (notado $v \cdot w$) es el número: $v_0 * w_0 + v_1 * w_1 + \cdots + v_{n-1} * w_{n-1}$.
3. Desarrollar un algoritmo que calcule el producto directo de dos arreglos de números reales de igual tamaño. Sean $v = [v_0, v_1, \ldots, v_{n-1}]$ y $w = [w_0, w_1, \ldots, w_{n-1}]$ dos arreglos, el producto directo de v y w (notado $v * w$) es el vector: $[v_0 * w_0, v_1 * w_1, \ldots, v_{n-1} * w_{n-1}]$.

Problemas de arreglos II

Problemas

4. Desarrollar un algoritmo que determine la mediana de un arreglo de enteros. La mediana es el número que queda en la mitad del arreglo después de ser ordenado.
5. Hacer un algoritmo que deje al final de un arreglo de números todos los ceros que aparezcan en dicho arreglo.

Ejemplo

vector original: $[1, 6, 0, 7, -3, 8, 0, -2, 11]$
vector salida: $[1, 6, 7, -3, 8, -2, 11, 0, 0]$

Ejemplo

vector original: $[0, 11, 36, 10, 0, 17, -23, 81, 0, 0, 12, 11, 0]$
vector salida: $[11, 36, 10, 17, -23, 81, 12, 11, 0, 0, 0, 0, 0]$

Problemas de matrices I

Problemas

1. Desarrollar un algoritmo que permita sumar dos matrices de números reales (enteros).
2. Desarrollar un algoritmo que permita multiplicar dos matrices de números reales (enteros).
3. Desarrollar un programa que sume los elementos de una columna dada de una matriz.
4. Desarrollar un programa que sume los elementos de una fila dada de una matriz.

Problemas de matrices II

Problemas

5. Desarrollar un algoritmo que determine si una matriz es mágica. Se dice que una matriz cuadrada es mágica si la suma de cada una de sus filas, de cada una de sus columnas y de cada diagonal es igual.

Ejemplo

8	1	6
3	5	7
4	9	2

Ejemplo

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

Problemas varios I

Problemas

1. Modele mediante una función matemática y desarrolle un algoritmo en Java con ciclos que permita hallar la posición del mínimo de un vector de números reales.
2. Modele mediante una función matemática y desarrolle un algoritmo con ciclos en Java que calcule el mínimo de un vector de números reales.

Problemas varios II

Problemas

4. Modele mediante una función matemática y desarrolle un algoritmo en Java que calcule el producto por escalar de una constante con un vector de números reales. Sea $\alpha \in \mathbb{R}$ y $v = (v_1, v_2, \ldots, v_n)$, el producto por escalar de α con v (notado αv) es el vector dado por la expresión

$$\alpha v = (\alpha v_1, \alpha v_2, \ldots, \alpha v_n)$$