

Arrays

Nota: Cada uno de los ejercicios de esta relación ha de desarrollarse en un archivo independiente cuyo nombre ha de ser de la forma “Ejercicio3???.java”, donde ‘??’ es el número del ejercicio. El desarrollo de estos ejercicios ha de basarse en la estructura básica vista en el tema 0:

```
'''Ejercicio3???.java'''  
  
class Ejercicio3?? {  
  
    public static void main ( String [] args ) {  
  
        // <Secuencia de instrucciones>  
    }  
}
```

donde <Secuencia de instrucciones> es la secuencia de instrucciones que implementa el programa descrito en el enunciado de cada ejercicio.

Ejercicio 1:

Desarrollar un programa que escriba en pantalla línea a línea todos los datos proporcionados en la entrada.

Ejercicio 2:

Desarrollar un programa que escriba en pantalla todos los datos proporcionados en la entrada en forma de vector, utilizando '[' y ']' como delimitadores y ',' como separador. De esta forma:

```
> java Ejercicio302 1 2 3 4 5  
[1,2,3,4,5]
```

Nota: Para la realización de los siguientes ejercicios será necesario utilizar la librería `Lectura.jar` que se encuentra en la página de la asignatura. A esta librería se han añadido métodos de lectura de vectores: `Lectura.vectorEntero()`, `Lectura.vectorReal()` y `Lectura.vectorDoble()`. Estos métodos devuelven un vector de elementos de tipo `int`, `float` y `double` respectivamente. El tamaño de este vector está determinado por la cantidad de datos que se proporcionan. Los datos se han de proporcionar uno detrás de otro separados por espacios en blanco.

Ejercicio 3:

Desarrollar un programa en el que se pida al usuario un vector de números enteros e indique en pantalla si los elementos de dicho vector están ordenados de menor a mayor o no.

Ejercicio 4:

Desarrollar un programa en el que se pida al usuario un vector de números enteros e indique en pantalla la media aritmética de todos sus elementos.

Ejercicio 5:

Desarrollar un programa en el que se pidan al usuario un vector de números enteros e indique en pantalla si dicho vector es capicúa, es decir, la secuencia de sus elementos es igual vista de delante hacia atrás y de detrás hacia delante.

Ejercicio 6:

Desarrollar un programa en el que se pida al usuario un vector de números enteros e indique en pantalla cuantos de dichos elementos son números impares. Para comprobar si un número es impar se puede utilizar la condición `(n % 2) == 1`.

Ejercicio 7:

Desarrollar un programa en el que se pida al usuario un vector de números enteros e indique en pantalla el número de ocurrencias de elementos repetidos. Por ejemplo el vector `[1,2,3,1,2,1]` tiene tres ocurrencias de elementos repetidos (dos 1 y un 2).

Ejercicio 8:

Desarrollar un programa en el que se pidan al usuario dos vectores de números enteros e indique en pantalla el producto escalar de los mismos. Si los vectores proporcionados no tienen la misma longitud, se ha de escribir en pantalla un mensaje indicativo.

Ejercicio 9:

Desarrollar un programa en el que se pidan al usuario dos vectores de números enteros e indique en pantalla si son proporcionales o no. (Dos vectores son proporcionales si uno de ellos es el resultado de multiplicar por

un mismo factor todos los elementos del otro). Si los vectores proporcionados no tienen la misma longitud, se ha de escribir en pantalla un mensaje indicativo.

Ejercicio 10:

Desarrollar un programa en el que se pida al usuario dos vectores de números enteros, v_1 y v_2 , e indique en pantalla si para toda posición i , el elemento i -ésimo de v_1 es menor o igual que el elemento i -ésimo de v_2 . Si los vectores no tienen la misma longitud, se ha de escribir en pantalla un mensaje indicativo.

Ejercicio 11:

Desarrollar un programa en el que se pidan al usuario dos vectores de números enteros v_1 y v_2 , se construya el vector resultado de “concatenar” los vectores v_1 y v_2 , es decir, poner los elementos de v_2 a continuación de los de v_1 y, finalmente, se escriban en pantalla todos los elementos de la concatenación.

Ejercicio 12:

Desarrollar un programa en el que se pida al usuario un número natural n (el método `Lectura.natural()` sirve para leer números naturales), se construya un vector de tamaño n tal que en la posición i -ésima almacene el factorial del número i y, finalmente, se escriban en pantalla todos los elementos de dicho vector.

Ejercicio 13:

Desarrollar un programa en el que se pida al usuario un número natural n , se construya un vector de tamaño n tal que en la posición i -ésima almacene el elemento i -ésimo de la sucesión de Fibonacci y, finalmente, se escriban en pantalla todos los elementos de dicho vector.

Ejercicio 14:

Desarrollar un programa en el que se pida al usuario un vector v_1 de números enteros, se construya un nuevo vector v_2 que almacene los elementos de v_1 eliminando las repeticiones y preservando el orden y, finalmente, se escriban en pantalla todos los elementos del vector v_2 . Para ajustar el tamaño de v_2 hay que calcular el número de ocurrencias de elementos repetidos en v_1 (ejercicio 7). El programa ha de escribir en pantalla el nuevo vector.

Ejercicio 15:

Desarrollar un programa en el que se pida al usuario dos vectores de números enteros, se construya un nuevo vector v que almacene la suma de ambos vectores y, finalmente, se escriban en pantalla todos los elementos de v . El vector suma se ha de ajustar al vector más largo proporcionado por el usuario, completando el más corto con ceros. Por ejemplo, la suma de los vectores $[1, 2, 3]$ y $[1, 2, 3, 4, 5]$ es $[2, 4, 6, 4, 5]$.

Ejercicio 16:

Desarrollar un programa en el que se pida al usuario un vector de números enteros v y se construya un nuevo vector u con los mismos elementos que v pero en el que los números negativos preceden a los positivos. Por ejemplo, si $v = [-1, 3, -2, 2, 1, -3]$ entonces u ha de ser $[-1, -2, -3, 3, 2, 1]$. El programa ha de escribir en pantalla el nuevo vector.

Ejercicio 17:

Desarrollar un programa en el que se pida al usuario un vector de números enteros v y se construye un nuevo vector u con los mismos elementos que v pero ordenados de menor a mayor. El programa ha de escribir en pantalla el nuevo vector.

Ejercicio 18:

Desarrollar un programa en el que se construya una matriz de tamaño 3×3 de números enteros a partir de los datos proporcionados por el usuario. Los datos de la matriz se pedirán con el procedimiento de lectura de vectores, **uno por fila**, desechando los elementos que se escriban de más y rellenando con ceros los que se escriban de menos. Una vez construida la matriz, el programa ha de escribir sus elementos en pantalla, una fila por línea.

Ejercicio 19:

Desarrollar un programa en el que se construya una matriz de tamaño 3×3 de números enteros a partir de los datos proporcionados por el usuario. Los datos de la matriz se pedirán con el procedimiento de lectura de vectores, **en un único vector**, desechando los elementos que se escriban de más y rellenando con ceros los que se escriban de menos. Una vez construida la matriz, el programa ha de escribir sus elementos en pantalla, una fila por línea.

Ejercicio 20:

Desarrollar un programa en el que se pidan al usuario dos números naturales positivos N y M y se construya una matriz A de tamaño $N \times M$ en la que el elemento $A[i, j]$ sea $i + j$. Una vez construida la matriz, el programa ha de escribir sus elementos en pantalla, una fila por línea.

Ejercicio 21:

Desarrollar un programa en el que se pidan al usuario dos números naturales positivos N y M y construya una matriz de dimensiones $N \times M$. Los datos de la matriz se pedirán con el procedimiento de lectura de vectores, **en un único vector**, desechando los elementos que se escriban de más y rellenando con ceros los que se escriban de menos. Una vez construida la matriz, el programa ha de escribir sus elementos en pantalla, una fila por línea.

Nota: A la librería `Lectura.jar` se ha añadido un método de lectura de matrices: `Lectura.matrizEntera()`. Este método pide al usuario las dimensiones de la matriz y un vector de números enteros, y construye una matriz de las dimensiones indicadas cuyos elementos se toman del vector proporcionado, desechando los elementos que se escriban de más y rellenando con ceros los que se escriban de menos. Es necesario volver a obtener dicha librería de la página de la asignatura.

Ejercicio 22:

Desarrollar un programa en el que se pida al usuario una matriz de dimensiones $N \times M$, y compruebe si la matriz es nula. (*Todos sus elementos iguales a cero*)

Ejercicio 23:

Desarrollar un programa en el que se pida al usuario una matriz de dimensiones $N \times M$, y compruebe si la matriz es positiva. (*Todos sus elementos mayores o iguales a cero*)

Ejercicio 24:

Desarrollar un programa en el que se pida al usuario una matriz de dimensiones $N \times M$, y compruebe si la matriz es negativa. (*Todos sus elementos menores o iguales a cero*)

Ejercicio 25:

Desarrollar un programa en el que se pida al usuario una matriz de dimensiones $N \times M$, y compruebe si la matriz es diagonal. (*Todos los elementos que no están en la diagonal principal son nulos*).

Ejercicio 26:

Desarrollar un programa en el que se pida al usuario una matriz de dimensiones $N \times M$, y compruebe si la matriz es triangular superior. (*Todos los elementos que están por debajo de la diagonal principal son nulos*).

Ejercicio 27:

Desarrollar un programa en el que se pida al usuario una matriz de dimensiones $N \times M$, y compruebe si la matriz es triangular inferior. (*Todos los elementos que están por encima de la diagonal principal son nulos*).

Ejercicio 28:

Desarrollar un programa en el que se pida al usuario una matriz de dimensiones $N \times M$, y compruebe si la matriz es dispersa. (*Todas las filas y todas las columnas contienen al menos un elemento nulo*).

Ejercicio 29:

Desarrollar un programa en el que se pida al usuario una matriz de dimensiones $N \times M$, y compruebe si la matriz es simétrica. (*Los elementos de la matriz (i, j) y (j, i) , si existen, son iguales*).

Ejercicio 30:

Desarrollar un programa en el que se pida al usuario una matriz de dimensiones $N \times M$, y construya una nueva matriz representando la matriz traspuesta de la inicial.

Ejercicio 31:

Desarrollar un programa en el que se pida al usuario una matriz de dimensiones $N \times M$, y construya una nueva matriz representando la matriz opuesta de la inicial.

Ejercicio 32:

Desarrollar un programa en el que se pidan al usuario una matriz de dimensiones $N \times M$ y dos índices de fila y columna i y j , y construya una nueva matriz representando la submatriz de la inicial que se obtiene eliminando la fila i y la columna j .

Ejercicio 33:

Desarrollar un programa en el que se pida al usuario una matriz de dimensiones $N \times M$, y construya un vector con la suma de todas las filas de la matriz inicial.

Ejercicio 34:

Desarrollar un programa en el que se pida al usuario una matriz de dimensiones $N \times M$, y construya un vector con la suma de todas las columnas de la matriz inicial.

Ejercicio 35:

Desarrollar un programa en el que se pidan al usuario una matriz de dimensiones $N \times M$ y un número entero r , y construya una nueva matriz representando el producto de r por la matriz inicial.

Ejercicio 36:

Desarrollar un programa en el que se pidan al usuario dos matrices de dimensiones $N \times M$, y construya una nueva matriz representando la suma de las matrices iniciales.

Ejercicio 37:

Desarrollar un programa en el que se pidan al usuario dos matrices de dimensiones $N \times M$, y construya una nueva matriz representando la resta de las matrices iniciales.

Ejercicio 38:

Desarrollar un programa en el que se pidan al usuario una matriz de dimensiones $N \times M$ y un vector v de tamaño N , y construya el vector resultado de multiplicar v por la matriz inicial.

Ejercicio 39:

Desarrollar un programa en el que se pidan al usuario una matriz de dimensiones $N \times M$ y un vector v de tamaño M , y construya el vector resultado de multiplicar la matriz inicial por v .

Ejercicio 40:

Desarrollar un programa en el que se pidan al usuario dos matrices de dimensiones $N \times M$, y construya una nueva matriz representando el producto de las matrices iniciales.