

Ejercicios arrays unidimensionales / multidimensionales

Ejercicio nº 1.

Se desea codificar en Java un programa, ejecutable, que nos permita introducir por teclado un vector con 20 valores enteros, y un número entero, el programa deberá construir dos nuevos vectores «mayores» y «menores», en los que dejará los valores del vector original mayores o menores, respectivamente, al número que se ha introducido. Finalmente, el programa mostrará los resultados al usuario, mostrando tanto el vector inicial, como los vectores resultantes. Por ejemplo:

- **Vector inicial introducido:**

13	2	23	7	1	19	43	22	19	3
----	---	----	---	---	----	----	----	----	---
- **Número introducido:** 15
- **Vectores resultantes:** menores:

13	2	7	1	3
----	---	---	---	---

 mayores:

23	19	43	22	19
----	----	----	----	----

El programa deberá repetirse mientras así lo desee el usuario.

Ejercicio nº 2.

Se desea codificar en Java un programa, ejecutable, que nos permita introducir por teclado dos vectores con 10 valores enteros, cada uno. El primer vector se encuentra en orden creciente, es decir, de menor a mayor (esto no debes comprobar, se supone bien introducido) y el segundo se encuentra en orden decreciente, es decir, de mayor a menor (tampoco lo debes comprobar), el programa deberá fusionar ambos vectores en un tercer vector, de forma que tras la fusión el vector obtenido se encuentre en orden creciente. El programa mostrará los resultados al usuario, mostrando los vectores introducidos y el resultante. Por ejemplo:

- **Vectores iniciales:**

12	18	26	30	35
----	----	----	----	----

42	25	13	12	6
----	----	----	----	---
- **Vector resultante:**

6	12	12	13	18	25	26	30	35	42
---	----	----	----	----	----	----	----	----	----

El programa deberá repetirse mientras así lo desee el usuario.

Ejercicio nº 3.

Se desea codificar en Java un programa, ejecutable, que nos permita introducir por teclado los valores enteros de una matriz de 5 filas y 5 columnas. El programa deberá indicar los pozos que tiene la matriz introducida. Se considera pozo a aquellos valores de la matriz que tanto por el norte, sur, este y oeste tienen un valor superior al suyo, o bien están al borde de la matriz. El programa mostrará los resultados al usuario, mostrando los valores de la matriz, así como las posiciones de los pozos, indicando fila y columna. Por ejemplo:

- **Matriz introducida:**

12	18	26	30	35
23	45	7	12	15
9	15	19	3	45
18	6	3	8	12
9	23	1	9	31

- **Pozos de la matriz:** Fila 1 / columna 1
 Fila 2 / columna 3
 Fila 3 / columna 1
 Fila 3 / columna 4
 Fila 5 / columna 1
 Fila 5 / columna 3

El programa deberá repetirse mientras así lo desee el usuario.

Ejercicio nº 4.

Se desea codificar en Java un programa, ejecutable, que nos permita introducir por teclado 10 números enteros y positivos. El programa, a continuación, pedirá al usuario que introduzca un número, seguidamente el programa obtendrá un nuevo vector con los valores del vector inicial que tengan los mismos dígitos que el número que se ha introducido. Por ejemplo:

- **Vector inicial introducido:**

147	25	365	42	128	956	100	43	563	653
-----	----	-----	----	-----	-----	-----	----	-----	-----
- **Número introducido:** 356
- **Vector resultante:**

365	563	653
-----	-----	-----

El programa deberá repetirse mientras así lo desee el usuario.

Ejercicio nº 5.

Un colegio electoral se ha constituido con 7 mesas electorales (numeradas desde la mesa 1 a la mesa 7), en cada una de las mesas electorales han ejercido su derecho al voto los ciudadanos inscritos. Al final de la jornada electoral, el presidente de la mesa le entrega al responsable del colegio el resultado que se ha obtenido en su mesa. Por ejemplo:

- **Mesa nº 3:**

658	23	129	418	3	315
Partido1	Partido2	Partido3	Partido4	Partido5	Partido6

Cuando finaliza todo el recuento del colegio electoral, el responsable del colegio deberá pasar a la Junta Electoral el resultado total obtenido por los 6 partidos que se presentaban a las elecciones, así como el partido ganador en su colegio. Codificar un programa en Java que nos permita realizar esta tarea.

Ejercicio nº 6.

La Ley D'Hont es un sistema electoral de cálculo proporcional que divide el número de votos emitidos para cada partido entre el número de cargos electos con los que cuenta cada circunscripción.

Para entenderlo mejor utilizaremos un ejemplo. Supongamos que en una circunscripción se presentan 4 partidos políticos y quieren conseguir los 5 cargos de diputados disponibles en la circunscripción, para ello los votos obtenidos por cada uno de los partidos se irán dividiendo entre 1, 2, 3, Hasta el total de los diputados en juego

Partidos	Votos	÷1	÷2	÷3	÷4	÷5	Partidos	Votos	÷1	÷2	÷3	÷4	÷5
A	5.000 Votos	5.000	2.500	1.666,6	1.250	1.000	A	5.000 Votos	5.000	2.500	1.666,6	1.250	1.000
B	4.000 Votos	4.000	2.000	1.333,3	1.000	800	B	4.000 Votos	4.000	2.000	1.333,3	1.000	800
C	3.500 Votos	3.500	1.750	1.166,6	875	700	C	3.500 Votos	3.500	1.750	1.166,6	875	700
D	1.000 Votos	1.000	500	333,3	250	200	D	1.000 Votos	1.000	500	333,3	250	200

Se divide el número de votos de cada partido por 1, 2, 3, 4 y 5

De todos los resultados obtenidos, los cinco diputados se asignan a las cinco cifras más altas, independientemente del partido que sean. Eso sí, para poder aspirar a obtener representación, hay que lograr un mínimo del 3% del total de los votos. En este caso el reparto quedará como muestra la tabla superior, es decir, el Partido A obtiene dos representantes, el Partido B, también obtiene 2 representantes y el partido C obtiene un representante, mientras que el Partido D no obtiene representación.

Realizar un programa en Java que nos permita introducir los resultados obtenidos por los 5 partidos políticos que se presentan a elecciones en una circunscripción, así como el número de diputados que se van a repartir, el programa deberá mostrarnos el reparto, así como el número de diputados conseguido por cada Partido