

7.4 Ejercicios

Arrays de una dimensión



Ejercicio 1

Define un *array* de 12 números enteros con nombre `num` y asigna los valores según la tabla que se muestra a continuación. Muestra el contenido de todos los elementos del *array*. ¿Qué sucede con los valores de los elementos que no han sido inicializados?

Índice	0	1	2	3	4	5	6	7	8	9	10	11
Valor	39	-2			0		14		5	120		



Ejercicio 2

Define un *array* de 10 caracteres con nombre `simbolo` y asigna valores a los elementos según la tabla que se muestra a continuación. Muestra el contenido de todos los elementos del *array*. ¿Qué sucede con los valores de los elementos que no han sido inicializados?

Índice	0	1	2	3	4	5	6	7	8	9
Valor	'a'	'x'			'@'		' '	'+'	'Q'	



Ejercicio 3

Escribe un programa que lea 10 números por teclado y que luego los muestre en orden inverso, es decir, el primero que se introduce es el último en mostrarse y viceversa.



Ejercicio 4

Define tres *arrays* de 20 números enteros cada una, con nombres `numero`, `cuadrado` y `cubo`. Carga el *array* `numero` con valores aleatorios entre 0 y 100. En el *array* `cuadrado` se deben almacenar los cuadrados de los valores que hay en el *array* `numero`. En el *array* `cubo` se deben almacenar los cubos de los valores que hay en `numero`. A continuación, muestra el contenido de los tres *arrays* dispuesto en tres columnas.



Ejercicio 5

Escribe un programa que pida 10 números por teclado y que luego muestre los números introducidos junto con las palabras “máximo” y “mínimo” al lado del máximo y del mínimo respectivamente.



Ejercicio 6

Escribe un programa que lea 15 números por teclado y que los almacene en un *array*. Rota los elementos de ese *array*, es decir, el elemento de la posición 0 debe pasar a la posición 1, el de la 1 a la 2, etc. El número que se encuentra en la última posición debe pasar a la posición 0. Finalmente, muestra el contenido del *array*.



Ejercicio 7

Escribe un programa que genere 100 números aleatorios del 0 al 20 y que los muestre por pantalla separados por espacios. El programa pedirá entonces por teclado dos valores y a continuación cambiará todas las ocurrencias del primer valor por el segundo en la lista generada anteriormente. Los números que se han cambiado deben aparecer entrecomillados.



Ejercicio 8

Realiza un programa que pida la temperatura media que ha hecho en cada mes de un determinado año y que muestre a continuación un diagrama de barras horizontales con esos datos. Las barras del diagrama se pueden dibujar a base de asteriscos o cualquier otro carácter.



Ejercicio 9

Realiza un programa que pida 8 números enteros y que luego muestre esos números junto con la palabra “par” o “impar” según proceda.



Ejercicio 10

Escribe un programa que genere 20 números enteros aleatorios entre 0 y 100 y que los almacene en un *array*. El programa debe ser capaz de pasar todos los números pares a las primeras posiciones del *array* (del 0 en adelante) y todos los números impares a las celdas restantes. Utiliza *arrays* auxiliares si es necesario.



Ejercicio 11

Realiza un programa que pida 10 números por teclado y que los almacene en un *array*. A continuación se mostrará el contenido de ese *array* junto al índice (0 - 9) utilizando para ello una tabla. Seguidamente el programa pasará los primos a las primeras posiciones, desplazando el resto de números (los que no son primos) de tal forma que no se pierda ninguno. Al final se debe mostrar el *array* resultante.

Por ejemplo:

Índice	0	1	2	3	4	5	6	7	8	9
Valor	20	5	7	4	32	9	2	14	11	6

Array inicial

Índice	0	1	2	3	4	5	6	7	8	9
Valor	5	7	2	11	20	4	32	9	14	6

Array final



Ejercicio 12

Realiza un programa que pida 10 números por teclado y que los almacene en un *array*. A continuación se mostrará el contenido de ese *array* junto al índice (0 - 9). Seguidamente el programa pedirá dos posiciones a las que llamaremos "inicial" y "final". Se debe comprobar que inicial es menor que final y que ambos números están entre 0 y 9. El programa deberá colocar el número de la posición inicial en la posición final, rotando el resto de números para que no se pierda ninguno. Al final se debe mostrar el *array* resultante.

Por ejemplo, para inicial = 3 y final = 7:

Índice	0	1	2	3	4	5	6	7	8	9
Valor	20	5	7	4	32	9	2	14	11	6

Array inicial

Índice	0	1	2	3	4	5	6	7	8	9
Valor	6	20	5	7	32	9	2	4	14	11

Array final