

7.4 Ejercicios

7.4.1 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.

Ejemplo 1:

```
Por favor, introduzca 10 números enteros.
Pulse la tecla INTRO después de introducir cada número.
```

```
42
66
89
100
31
7
8
16
24
23
```

```
Los números introducidos, al revés, son los siguientes:
```

```
23
24
```

```

16
8
7
31
100
89
66
42

```

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.

Ejemplo:

A continuación se muestran en tres columnas, un número aleatorio entre 0 y 100, su cuadrado y su cubo:

n	n ²	n ³
22	484	10648
18	324	5832
99	9801	970299
52	2704	140608
63	3969	250047
30	900	27000
52	2704	140608
1	1	1
67	4489	300763
60	3600	216000
94	8836	830584
31	961	29791
76	5776	438976
42	1764	74088
99	9801	970299
68	4624	314432
53	2809	148877
99	9801	970299
1	1	1
76	5776	438976

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.

Ejemplo:

Vaya introduciendo números enteros y pulsando INTRO:

```
88
62
13
95
22
3000
2000
-10
-20
19

88
62
13
95
22
3000 máximo
2000
-10
-20 mínimo
19
```

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*.

Ejemplo:

Vaya introduciendo números enteros y pulsando INTRO:

```
11
22
33
44
55
66
77
88
99
```

```
100
200
300
400
500
600
```

Array original:

```
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
-----
| 11 | 22 | 33 | 44 | 55 | 66 | 77 | 88 | 99 | 100 | 200 | 300 | 400 | 500 | 600 |
```

Array rotado a la derecha una posición:

```
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
-----
| 600 | 11 | 22 | 33 | 44 | 55 | 66 | 77 | 88 | 99 | 100 | 200 | 300 | 400 | 500 |
```

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.

Ejemplo 1:

```
5 11 13 13 17 3 4 11 20 0 20 1 2 7 13 17 0 19 18 15 19 15 7
2 6 12 1 13 9 0 9 16 7 7 4 17 10 18 1 4 11 4 7 4 16 17 14
8 9 6 16 18 2 7 15 14 7 7 0 6 6 6 19 16 20 16 19 15 3 17 4
8 17 18 4 11 12 15 16 18 13 20 1 14 10 17 16 2 2 19 2 4 13
17 8 8 10 20 10 19
```

Introduzca un número de los que se han mostrado: 13

Introduzca el valor por el que quiere sustituirlo: 88

```
5 11 "88" "88" 17 3 4 11 20 0 20 1 2 7 "88" 17 0 19 18 15 19
15 7 2 6 12 1 "88" 9 0 9 16 7 7 4 17 10 18 1 4 11 4 7 4 16
17 14 8 9 6 16 18 2 7 15 14 7 7 0 6 6 6 19 16 20 16 19 15
3 17 4 8 17 18 4 11 12 15 16 18 "88" 20 1 14 10 17 16 2 2 19
2 4 "88" 17 8 8 10 20 10 19
```

Ejemplo 2:

```
20 7 8 1 19 19 16 7 5 13 8 2 3 12 0 15 16 6 6 20 2 11 0 20
11 15 20 17 16 12 16 1 14 3 5 17 16 8 17 20 8 7 4 16 1 19 3
19 3 20 1 2 14 11 4 4 18 2 1 5 16 11 18 1 14 19 19 1 9 7 6
13 20 19 9 8 3 4 15 15 19 1 18 19 5 1 3 13 17 11 12 2 0 19
10 12 18 20 18 0
```

Introduzca un número de los que se han mostrado: 11

Introduzca el valor por el que quiere sustituirlo: 7

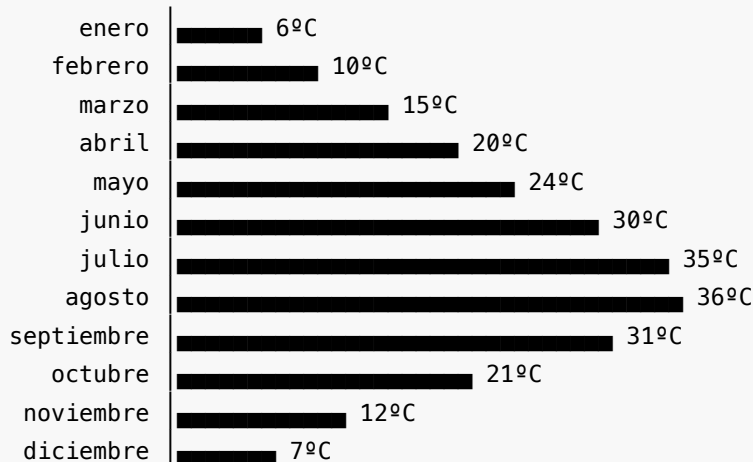
```
20 7 8 1 19 19 16 7 5 13 8 2 3 12 0 15 16 6 6 20 2 "7" 0
20 "7" 15 20 17 16 12 16 1 14 3 5 17 16 8 17 20 8 7 4 16 1
19 3 19 3 20 1 2 14 "7" 4 4 18 2 1 5 16 "7" 18 1 14 19 19 1
9 7 6 13 20 19 9 8 3 4 15 15 19 1 18 19 5 1 3 13 17 "7" 12
2 0 19 10 12 18 20 18 0
```

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.

Ejemplo 1:

```
Introduzca la temperatura media de enero: 6
Introduzca la temperatura media de febrero: 10
Introduzca la temperatura media de marzo: 15
Introduzca la temperatura media de abril: 20
Introduzca la temperatura media de mayo: 24
Introduzca la temperatura media de junio: 30
Introduzca la temperatura media de julio: 35
Introduzca la temperatura media de agosto: 36
Introduzca la temperatura media de septiembre: 31
Introduzca la temperatura media de octubre: 21
Introduzca la temperatura media de noviembre: 12
Introduzca la temperatura media de diciembre: 7
```



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.

Ejemplo:

```
Introduzca 8 números enteros, pulse INTRO después de cada número:
```

```
79
44
33
21
50
45
80
303
```

```
79 impar
44 par
33 impar
21 impar
50 par
45 impar
80 par
303 impar
```

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.

Ejemplo 1:

```
Array original:
```

```
25 65 74 24 16 45 41 84 67 11 18 95 40 71 51 15 4 99 17 16
```

```
Array con los pares al principio:
```

```
74 24 16 84 18 40 4 16 25 65 45 41 67 11 95 71 51 15 99 17
```

Ejemplo 2:

```
Array original:
```

```
56 100 91 79 37 61 20 29 20 85 84 7 76 76 57 3 36 61 35 93
```

```
Array con los pares al principio:
```

```
56 100 20 20 84 76 76 36 91 79 37 61 29 85 7 57 3 61 35 93
```

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

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

Introduzca la posición inicial (0 – 9): 3

Introduzca la posición final (0 – 9): 7

Array original:

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

Array resultante:

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

Ejercicio 13

Escribe un programa que rellene un *array* de 100 elementos con números enteros aleatorios comprendidos entre 0 y 500 (ambos incluidos). A continuación el programa mostrará el *array* y preguntará si el usuario quiere destacar el máximo o el mínimo. Seguidamente se volverá a mostrar el *array* escribiendo el número destacado entre dobles asteriscos.

Ejemplo:

```
459 204 20 250 178 90 353 32 229 357 224 454 260 310 140 249 332 426 423 413 96
447 465 298 459 411 118 480 302 417 42 82 126 82 474 362 76 190 104 21 257 88 21
251 6 383 47 78 392 394 244 494 87 253 376 379 98 364 237 13 299 228 409 402 225
426 267 330 243 209 426 435 309 356 173 130 416 15 477 34 28 377 193 481 368 466
262 422 275 384 399 397 87 218 84 312 480 207 68 108
```

¿Qué quiere destacar? (1 – mínimo, 2 – máximo): 1

```
459 204 20 250 178 90 353 32 229 357 224 454 260 310 140 249 332 426 423 413 96
447 465 298 459 411 118 480 302 417 42 82 126 82 474 362 76 190 104 21 257 88 21
251 **6** 383 47 78 392 394 244 494 87 253 376 379 98 364 237 13 299 228 409 402
225 426 267 330 243 209 426 435 309 356 173 130 416 15 477 34 28 377 193 481 368
466 262 422 275 384 399 397 87 218 84 312 480 207 68 108
```


Ejercicio 14

Escribe un programa que pida 8 palabras y las almacene en un *array*. A continuación, las palabras correspondientes a colores se deben almacenar al comienzo y las que no son colores a continuación. Puedes utilizar tantos *arrays* auxiliares como quieras. Los colores que conoce el programa deben estar en otro *array* y son los siguientes: verde, rojo, azul, amarillo, naranja, rosa, negro, blanco y morado.

Ejemplo:

Introduzca 8 palabras (vaya pulsando [INTRO] entre una y otra.

```

casa
azul
verde
orden
morado
bombilla
bici
rosa

```

Array original:

0	1	2	3	4	5	6	7
casa	azul	verde	orden	morado	bombilla	bici	rosa

Array resultado:

0	1	2	3	4	5	6	7
azul	verde	morado	rosa	casa	orden	bombilla	bici

Ejercicio 15

Un restaurante nos ha encargado una aplicación para colocar a los clientes en sus mesas. En una mesa se pueden sentar de 0 (mesa vacía) a 4 comensales (mesa llena). Cuando llega un cliente se le pregunta cuántos son. De momento el programa no está preparado para colocar a grupos mayores de 4, por tanto, si un cliente dice por ejemplo que son un grupo de 6, el programa dará el mensaje **“Lo siento, no admitimos grupos de 6, haga grupos de 4 personas como máximo e intente de nuevo”**. Para el grupo que llega, se busca siempre la primera mesa libre (con 0 personas). Si no quedan mesas libres, se busca donde haya un hueco para todo el grupo, por ejemplo si el grupo es de dos personas, se podrá colocar donde haya una o dos personas. Inicialmente, las mesas se cargan con valores aleatorios entre 0 y 4. Cada vez que se

sientan nuevos clientes se debe mostrar el estado de las mesas. Los grupos no se pueden romper aunque haya huecos sueltos suficientes.

Ejemplo:

Mesa nº	1	2	3	4	5	6	7	8	9	10
Ocupación	3	2	0	2	4	1	0	2	1	1

¿Cuántos son? (Introduzca -1 para salir del programa): 2

Por favor, siéntense en la mesa número 3.

Mesa nº	1	2	3	4	5	6	7	8	9	10
Ocupación	3	2	2	2	4	1	0	2	1	1

¿Cuántos son? (Introduzca -1 para salir del programa): 4

Por favor, siéntense en la mesa número 7.

Mesa nº	1	2	3	4	5	6	7	8	9	10
Ocupación	3	2	2	2	4	1	4	2	1	1

¿Cuántos son? (Introduzca -1 para salir del programa): 3

Tendrán que compartir mesa. Por favor, siéntense en la mesa número 6.

Mesa nº	1	2	3	4	5	6	7	8	9	10
Ocupación	3	2	2	2	4	4	4	2	1	1

¿Cuántos son? (Introduzca -1 para salir del programa): 4

Lo siento, en estos momentos no queda sitio.

Mesa nº	1	2	3	4	5	6	7	8	9	10
Ocupación	3	2	2	2	4	4	4	2	1	1

¿Cuántos son? (Introduzca -1 para salir del programa): -1

Gracias. Hasta pronto.

Ejercicio 16

Escribe un programa que rellene un *array* de 20 elementos con números enteros aleatorios comprendidos entre 0 y 400 (ambos incluidos). A continuación, el programa mostrará el *array* y preguntará al usuario si quiere resaltar los múltiplos de 5 o los múltiplos de 7. Seguidamente se volverá a mostrar el *array* escribiendo los números que se quieren resaltar entre corchetes.

Ejemplo 1:

```
159 204 20 250 178 90 353 32 229 357 224 54 260 310 140 249 335 326 223 13
¿Qué números quiere resaltar? (1 – los múltiplos de 5, 2 – los múltiplos de 7): 1
159 204 [20] [250] 178 [90] 353 32 229 357 224 54 [260] [310] [140] 249 [335] 326 223 13
```

Ejemplo 2:

```
63 90 375 65 77 133 201 381 26 186 244 176 58 350 276 378 294 169 391 384
¿Qué números quiere resaltar? (1 – los múltiplos de 5, 2 – los múltiplos de 7): 2
[63] 90 375 65 [77] [133] 201 381 26 186 244 176 58 [350] 276 [378] [294] 169 391 384
```

Ejercicio 17

Escribe un programa que muestre por pantalla un *array* de 10 números enteros generados al azar entre 0 y 100. A continuación, el programa debe pedir un número al usuario. Se debe comprobar que el número introducido por teclado se encuentra dentro del array, en caso contrario se mostrará un mensaje por pantalla y se volverá a pedir un número; así hasta que el usuario introduzca uno correctamente. Seguidamente, el programa rotará el *array* hacia la derecha las veces que haga falta hasta que el número introducido quede situado en la posición 0 del *array*. Por último, se mostrará el *array* rotado por pantalla.

Ejemplo 1:

Array original:

Índice	0	1	2	3	4	5	6	7	8	9
Valor	235	20	70	215	276	114	135	90	243	150

Introduzca uno de los números del array: 276

Array resultante:

Índice	0	1	2	3	4	5	6	7	8	9
Valor	276	114	135	90	243	150	235	20	70	215

Ejemplo 2:

Array original:

Índice	0	1	2	3	4	5	6	7	8	9
Valor	321	170	299	229	166	353	61	336	360	269

Introduzca uno de los números del array: 20

Ese número no se encuentra en el array.

Introduzca uno de los números del array: 10

Ese número no se encuentra en el array.

Introduzca uno de los números del array: 61

Array resultante:

Índice	0	1	2	3	4	5	6	7	8	9
Valor	61	336	360	269	321	170	299	229	166	353

Ejercicio 18

Realiza un programa que genere 10 números enteros aleatorios entre 0 y 200 ambos incluidos y que los almacene en un *array*. A continuación, el programa debe mostrar el contenido de ese *array* junto al índice (0 – 9). Seguidamente el programa debe colocar de forma alterna y en orden los menores o iguales de 100 y los mayores de 100: primero menor, luego mayor, luego menor, luego mayor... Cuando se acaben los menores o los mayores, se completará con los números que queden.

Ejemplo 1:

Array original:

Índice	0	1	2	3	4	5	6	7	8	9
Valor	127	178	11	39	121	82	130	47	128	129

Array resultado:

Índice	0	1	2	3	4	5	6	7	8	9
Valor	11	127	39	178	82	121	47	130	128	129

Valor	5	82	0	131	113	200	134	44	48	134	68	151
-------	---	----	---	-----	-----	-----	-----	----	----	-----	----	-----

Introduzca el número que quiere insertar: 77

Introduzca la posición donde lo quiere insertar (0 – 11): 6

Array resultado:

Índice	0	1	2	3	4	5	6	7	8	9	10	11
Valor	5	82	0	131	113	200	77	134	44	48	134	68

Ejemplo 2:

Array original:

Índice	0	1	2	3	4	5	6	7	8	9	10	11
Valor	148	86	64	126	77	148	182	99	8	126	73	20

Introduzca el número que quiere insertar: 33

Introduzca la posición donde lo quiere insertar (0 – 11): 11

Array resultado:

Índice	0	1	2	3	4	5	6	7	8	9	10	11
Valor	148	86	64	126	77	148	182	99	8	126	73	33

Ejemplo 3:

Array original:

Índice	0	1	2	3	4	5	6	7	8	9	10	11
Valor	186	4	58	150	200	141	34	137	62	170	200	70

Introduzca el número que quiere insertar: 88

Introduzca la posición donde lo quiere insertar (0 – 11): 2

Array resultado:

Índice	0	1	2	3	4	5	6	7	8	9	10	11
Valor	186	4	88	58	150	200	141	34	137	62	170	200

Ejercicio 20

Implementa un programa que calcule la denominación ordinal de los reyes de una secuencia histórica. El programa solicitará la cantidad de reyes que se van a introducir y, a continuación, recibirá los nombres de los reyes. Presentará por pantalla dichos nombres, pero colocándoles el ordinal correspondiente. Así, por ejemplo, si hay dos Felipes en los nombres de los reyes, el primero debería aparecer como Felipe 1º y el segundo como Felipe 2º.

Ejemplo 1:

```
Introduzca el número total de nombres de reyes: 7
Vaya introduciendo los nombres de los reyes y pulsando INTRO.
Felipe
Carlos
Carlos
Fernando
Carlos
Carlos
Felipe

Los reyes introducidos son:
Felipe 1º
Carlos 1º
Carlos 2º
Fernando 1º
Carlos 3º
Carlos 4º
Felipe 2º
```

Ejemplo 2:

```
Introduzca el número total de nombres de reyes: 9
Vaya introduciendo los nombres de los reyes y pulsando INTRO.
Luis
Fernando
Fernando
Carlos
Amadeo
Alfonso
Carlos
Alfonso
Alfonso

Los reyes introducidos son:
Luis 1º
Fernando 1º
Fernando 2º
Carlos 1º
Amadeo 1º
Alfonso 1º
Carlos 2º
Alfonso 2º
Alfonso 3º
```

Ejercicio 21

Escribe un programa que rellene un *array* de 15 elementos con números enteros comprendidos entre 0 y 500 (ambos incluidos). A continuación, se mostrará el *array* “cincuerizado”, según el siguiente criterio: si el número que hay en una posición del *array* es múltiplo de 5, se deja igual, y si no, se cambia por el siguiente múltiplo de 5 que exista a partir de él.

Ejemplo:

Array original:

459 204 20 250 178 90 353 35 229 357 224 454 260 310 140

Array cincuerizado:

460 205 20 250 180 90 355 35 230 360 225 455 260 310 140

Ejercicio 22

Realiza un programa que sea capaz de recolocar los números de un *array* de fuera hacia adentro. En primer lugar, el programa pedirá al usuario el tamaño. A continuación generará un *array* con esa longitud con números enteros aleatorios entre 0 y 200 ambos incluidos. Seguidamente, el programa irá colocando desde fuera hacia adentro los números de tal forma que el primero se coloca en la primera posición, el segundo en la última, el tercero en la segunda, el cuarto en la penúltima, el quinto en la tercera, en sexto en la antepenúltima, etc. Se debe mostrar por pantalla tanto el *array* original como el *array* resultado.

Ejemplo 1:

Introduzca el tamaño del array: 12

Array original:

Índice	0	1	2	3	4	5	6	7	8	9	10	11
Valor	120	148	40	108	188	94	60	65	152	27	121	79

Array resultado:

Índice	0	1	2	3	4	5	6	7	8	9	10	11
Valor	120	40	188	60	152	121	79	27	65	94	108	148

Ejemplo 2:

Introduzca el tamaño del array: 7

Array original:

Índice	0	1	2	3	4	5	6
Valor	130	36	93	188	20	126	36

Array resultado:

Índice	0	1	2	3	4	5	6
Valor	130	93	20	36	126	188	36