



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



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:

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 a 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. El funcionamiento del programa se ilustra a continuación:

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



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. A continuación, 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.



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

Ejemplo 2:

Array original:

Índice	0	1	2	3	4	5	6	7	8	9
Valor	170	189	87	149	176	110	119	9	33	157

Array resultado:

Índice	0	1	2	3	4	5	6	7	8	9
Valor	87	170	9	189	33	149	176	110	119	157

Ejemplo 3:

Array original:

Índice	0	1	2	3	4	5	6	7	8	9
Valor	119	88	139	146	34	4	195	160	27	115

Array resultado:

Índice	0	1	2	3	4	5	6	7	8	9
Valor	88	119	34	139	4	146	27	195	160	115



Ejercicio 19

Realiza un programa que sea capaz de insertar un número en una posición concreta de un array. En primer lugar, el programa generará un array de 12 números enteros aleatorios entre 0 y 200 ambos incluidos. A continuación se debe mostrar el contenido de ese array junto al índice (0 – 11). Seguidamente el programa preguntará por el número que se quiere insertar y por la posición donde será insertado. Los números del array se desplazan a la derecha para dejar sitio al nuevo. El último número (el que se encuentra en la posición 11) siempre se perderá.

Ejemplo 1:

Array original:

Índice	0	1	2	3	4	5	6	7	8	9	10	11
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
```