

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?.

0	1	2	3	4	5	6	7	8	9	10	11
3	-			0		1		5	12		
9	2					4			0		

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?.

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

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 I:

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

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	n2	n3
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

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

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 |

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 I:

```
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 II:

```
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
```

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:

Introduzca la temperatura media de enero: 6

Introduzca la temperatura media de febrero: 10

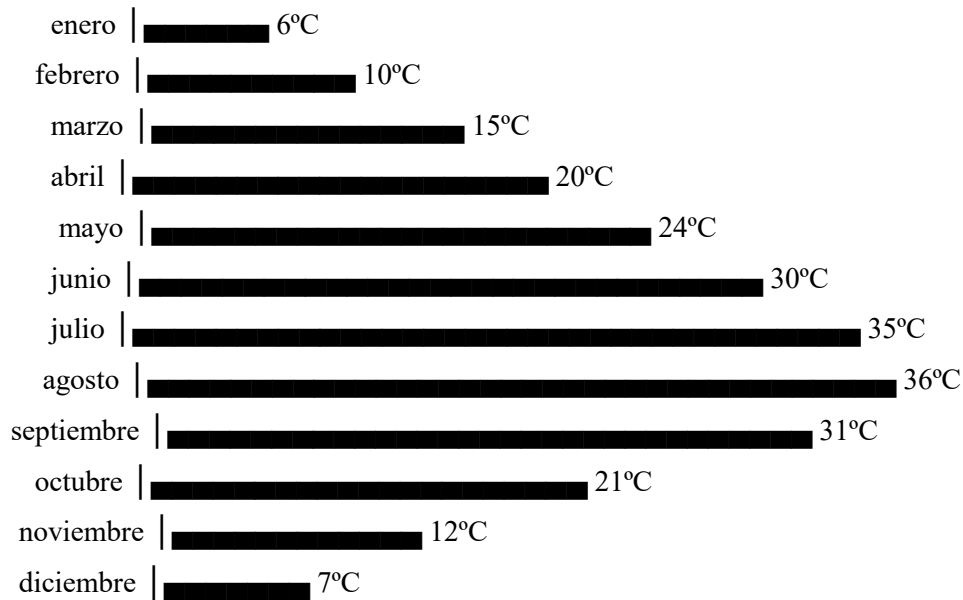
Introduzca la temperatura media de marzo: 15

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



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

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 I:

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 II:

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

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.

Ejemplo:

Array original:

Índice	0	1	2	3	4	5	6	7	8	9
Valor	20	30	19	67	11	13	100	99	111	321

Array con los primos al principio:

Índice	0	1	2	3	4	5	6	7	8	9
Valor	19	67	11	13	20	30	100	99	111	321

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.

Ejemplo:

Introduzca 10 números separados por INTRO:

20

5

7

4 32 9 2 14 11 6

Array original:

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

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

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 resultante:

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

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	3	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	3	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	3	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	0	2	4	1	0	2	1	1

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

Gracias. Hasta pronto.

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 I:

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 II:

3 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

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:

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: 500

Ese número no se encuentra en el array.

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

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 I:

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 II:

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

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 I:

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 II:

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 III:

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
-------	-----	---	----	----	-----	-----	-----	----	-----	----	-----	-----

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 I:

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 II:

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

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

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 I:

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 II:

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

23. Define un array de números enteros de 3 filas por 6 columnas con nombre num y asigna los valores según la siguiente tabla. Muestra el contenido de todos los elementos del array dispuestos en forma de tabla.

	columna 0	columna 1	columna 2	columna 3	columna 4	columna 5
fila 0	0	30	2			5
fila 1	75				0	
fila 2			-2	9		11

Ejemplo:

	Columna 0	Columna 1	Columna 2	Columna 3	Columna 4	Columna 5
Fila 0	0	30	2	0	0	7
Fila 1	75	0	0	0	0	0
Fila 2	0	0	-2	9	0	11

24. Escribe un programa que pida 20 números enteros. Estos números se deben introducir en un *array* de 4 filas por 5 columnas. El programa mostrará las sumas parciales de filas y columnas igual que si de una hoja de cálculo se tratara. La suma total debe aparecer en la esquina inferior derecha.

Ejemplo:

Por favor, introduzca los números (enteros) en el array

Fila 0, columna 0: 11

Fila 0, columna 1: 22

Fila 0, columna 2: 33

Fila 0, columna 3: 44

Fila 0, columna 4: 55

Fila 1, columna 0: 66

Fila 1, columna 1: 77

Fila 1, columna 2: 88

Fila 1, columna 3: 99

Fila 1, columna 4: 10

Fila 2, columna 0: 20

Fila 2, columna 1: 30

Fila 2, columna 2: 40

Fila 2, columna 3: 50
Fila 2, columna 4: 60
Fila 3, columna 0: 70
Fila 3, columna 1: 80
Fila 3, columna 2: 90
Fila 3, columna 3: 100
Fila 3, columna 4: 200

11	22	33	44	55		165
66	77	88	99	10		340
20	30	40	50	60		200
70	80	90	100	200		540

167	209	251	293	325		1245
-----	-----	-----	-----	-----	--	------

25. Modifica el programa anterior de tal forma que los números que se introducen en el array se generen de forma aleatoria (valores entre 100 y 999).

Ejemplo:

951	616	433	534	554		3088
788	565	133	729	622		2837
951	924	622	165	411		3073
366	464	802	485	950		3067

3056	2569	1990	1913	2537		12065
------	------	------	------	------	--	-------

26. Modifica el programa anterior de tal forma que las sumas parciales y la suma total aparezcan en la pantalla con un pequeño retardo, dando la impresión de que el ordenador se queda “pensando” antes de mostrar los números.

Ejemplo:

809	480	830	768	623		3510
804	740	981	441	846		3812
102	769	311	635	658		2475
526	364	199	505	395		1989

2241	2353	2321	2349	2522		11786
------	------	------	------	------	--	-------

NOTA: puedes añadir un retardo utilizando la función **sleep** de la clase **Thread**. Busca información sobre dicha función en la documentación oficial de Java para saber cómo usarla.

27. Realiza un programa que rellene un array de 6 filas por 10 columnas con números enteros positivos comprendidos entre 0 y 1000 (ambos incluidos). A continuación, el programa deberá dar la posición tanto del máximo como del mínimo.

Ejemplo:

	0	1	2	3	4	5	6	7	8	9
0	333	905	177	297	25	796	799	204	846	825
1	763	37	718	767	841	396	722	840	879	941
2	757	823	823	864	826	607	597	37	660	663
3	821	313	955	630	24	556	527	881	715	534
4	298	199	384	639	127	537	286	622	153	656
5	635	965	505	569	650	233	947	913	782	806

El máximo es 965 y está en la fila 5, columna 1

El mínimo es 24 y está en la fila 3, columna 4

28. Modifica el programa anterior de tal forma que no se repita ningún número en el *array*.

Ejemplo:

	0	1	2	3	4	5	6	7	8	9
0	454	293	17	193	643	691	89	472	660	715
1	565	609	84	809	975	266	615	981	754	403
2	178	32	739	869	299	1000	937	288	485	726
3	402	566	215	199	631	586	238	97	714	709
4	342	867	733	312	964	938	758	532	150	379
5	287	274	90	810	120	386	47	662	953	284

El máximo es 1000 y está en la fila 2, columna 5

El mínimo es 17 y está en la fila 0, columna 2

29. El juego «**Busca el tesoro**» coloca inicialmente, y de forma aleatoria, una mina y un tesoro sobre un tablero de 4 filas por 5 columnas. Hecho esto, el jugador debe averiguar dónde está el tesoro indicando las coordenadas (x, y). Si se encuentra el tesoro el programa debe terminar con el mensaje «**¡Enhorabuena! ¡Has encontrado el tesoro!**». Sin embargo, si se encuentra la mina el jugador pierde la partida y el programa acaba. Al finalizar, se muestra el tablero completo, mostrando los disparos realizados por el jugador, la mina y el tesoro.

Ejemplo I:

¡BUSCA EL TESORO!

3|

2|

1|

0|

0 1 2 3 4

Coordenada x: 3

Coordenada y: 2

3|

2| X

1|

0|

0 1 2 3 4

Coordenada x: 1

Coordenada y: 1

3|

2| X

1| X

0|

0 1 2 3 4

Coordenada x: 3

Coordenada y: 0

¡BOOOM! Lo siento, has perdido.

3|

2| X

1| X €

0| *

0 1 2 3 4

Ejemplo II:

¡BUSCA EL TESORO!

3|

2|

1|

0|

0 1 2 3 4

Coordenada x: 3

Coordenada y: 2

3|

2| X

1|

0|

0 1 2 3 4

Coordenada x: 4

Coordenada y: 1

¡ENHORABUENA! ¡Has encontrado el tesoro!

3|

2| X

1| €

0|

0 1 2 3 4

30. Mejora el juego «**Busca el tesoro**» de tal forma que si hay una mina a una casilla de distancia, el programa avise diciendo «**¡Cuidado! ¡Hay una mina cerca!**»

Ejemplo:

¡BUSCA EL TESORO!

3|

2|

1|

0|

0 1 2 3 4

Coordenada x: 3

Coordenada y: 2

3|

2| X

1|

0|

0 1 2 3 4

Coordenada x: 1

Coordenada y: 1

Cuidado, hay una mina cerca.

3|

2| X

1| X

0|

0 1 2 3 4

Coordenada x: 3

Coordenada y: 0

3|

2| X

1| X

0| X

0 1 2 3 4

Coordenada x: 0

Coordenada y: 2

Cuidado, hay una mina cerca.

3|

```

2|X  X
1| X
0|  X
-----
0 1 2 3 4

```

Coordenada x: 0

Coordenada y: 0

Cuidado, hay una mina cerca.

```

3|
2|X  X
1| X
0|X  X
-----
0 1 2 3 4

```

Coordenada x: 2

Coordenada y: 2

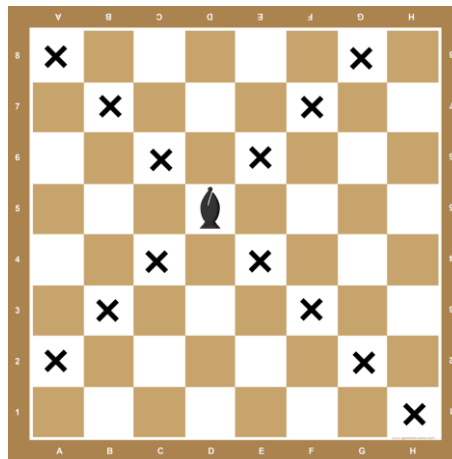
¡ENHORABUENA! ¡Has encontrado el tesoro!

```

3|
2|X  € X
1|* X
0|X  X
-----
0 1 2 3 4

```

31. Escribe un programa que, dada una posición en un tablero de ajedrez, nos diga a qué casillas podría saltar un alfil que se encuentra en esa posición. Como se indica en la figura, el alfil se mueve siempre en diagonal. El tablero cuenta con 64 casillas. Las columnas se indican con las letras de la “a” a la “h” y las filas se indican del 1 al 8. En la siguiente imagen puedes ver un alfil colocado en la posición 5D y marcadas todas aquellas casillas a las que se puede mover.



Ejemplo I:

Introduzca la posición del alfil: d5

El álfil puede moverse a las siguientes posiciones:

h1 a2 g2 b3 f3 c4 e4 c6 e6 b7 f7 a8 g8

Ejemplo II:

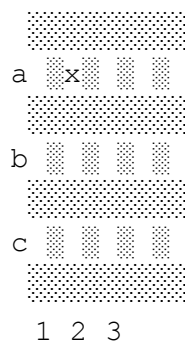
Introduzca la posición del alfil, por ejemplo d5: e2

El alfil puede moverse a las siguientes posiciones:

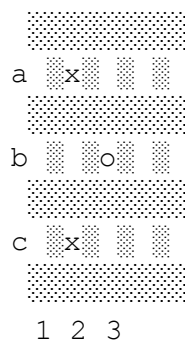
a6 b5 h5 c4 g4 d3 f3 d1 f1

32. Realiza el juego de las tres en raya para jugar contra el ordenador.

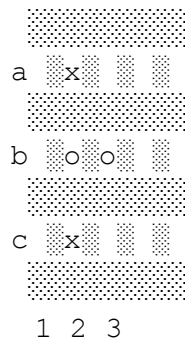
Ejemplo:



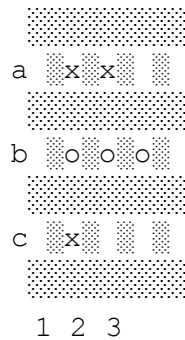
Introduzca las coordenadas, por ejemplo b2: b2



Introduzca las coordenadas, por ejemplo b2: b1



Introduzca las coordenadas, por ejemplo b2: b3



¡ENHORABUENA! ¡Has ganado!

33. Realiza un programa que muestre por pantalla un *array* de 10 filas por 10 columnas relleno con números aleatorios entre 200 y 300. A continuación, el programa debe mostrar los números de la diagonal que va desde la esquina superior izquierda a la esquina inferior derecha, así como el máximo, el mínimo y la media de los números que hay en esa diagonal.

Ejemplo:

```
287 200 203 204 275 253 267 288 205 295
292 254 254 271 241 297 250 272 227 216
292 243 247 232 239 232 200 235 261 300
233 276 220 233 290 204 275 285 295 281
285 257 238 264 254 220 222 269 264 212
228 261 251 243 251 211 221 254 280 215
207 230 251 283 296 205 248 287 240 279
275 254 256 223 251 224 218 231 246 276
275 236 210 262 261 258 266 252 223 249
202 276 249 227 259 223 261 296 262 225
```

Diagonal desde la esquina superior izquierda a la esquina inferior derecha: 287 254 247 233 254 211 248 231 223 225

Máximo: 287

Mínimo: 211

Media: 241.3

34. Realiza un programa que muestre por pantalla un *array* de 9 filas por 9 columnas relleno con números aleatorios entre 500 y 900. A continuación, el programa debe mostrar los números de la diagonal que va desde la esquina inferior izquierda a la esquina superior derecha, así como el máximo, el mínimo y la media de los números que hay en esa diagonal.

Ejemplo:

```
794 741 664 807 849 739 518 650 839
513 868 819 757 694 613 633 637 643
838 538 761 871 896 617 546 795 731
682 748 687 812 501 811 550 700 656
506 789 682 879 669 588 634 602 895
727 630 684 664 546 883 721 726 740
656 506 703 525 608 758 571 853 780
707 531 512 587 655 792 620 703 774
579 899 507 660 814 522 501 883 830
```

Diagonal desde la esquina inferior izquierda a la esquina superior derecha: 579 531 703 664 669 811 546 637 839

Máximo: 839

Mínimo: 531

Media: 597.9

35. Realiza un programa que calcule la estatura media, mínima y máxima en centímetros de personas de diferentes países. El *array* que contiene los nombres de los países es el siguiente: **pais = {"España", "Rusia", "Japón", "Australia"}**. Los datos sobre las estaturas se deben simular mediante un *array* de 4 filas por 10 columnas con números aleatorios generados al azar entre 140 y 210. Los decimales de la media se pueden despreciar. Los nombres de los países se deben mostrar utilizando el *array* de países (no se pueden escribir directamente).

Ejemplo I:

MED MIN MAX

```
España: 178 165 148 185 155 141 165 149 155 201 | 164 141 201
Rusia: 179 189 208 167 186 174 152 192 173 179 | 179 152 179
Japón: 173 182 168 170 181 197 146 168 166 177 | 172 146 177
Australia: 172 170 187 186 197 143 190 199 187 191 | 182 143 191
```