

## Ejercicios Unidad 2

1. Crear un programa que indique si un número introducido es positivo o negativo.
2. Crear un programa que solicite dos números e indique cual es el mayor de los dos.
3. Crear un programa que indique si un número introducido por teclado es par y/o divisible entre 5.
4. Crear un programa que solicite dos números y los reste, de forma que siempre se reste el menor al mayor, quedando un número positivo. Si los números son iguales debe mostrar un mensaje y no realizar ninguna operación.
5. Crear un programa que solicite dos números y los muestre ordenados de mayor a menor.
6. Crear un programa que solicite tres números enteros y muestre cual es el mayor utilizando if-else. Tener en cuenta que los números introducidos pueden ser iguales entre ellos.
7. Crear un programa que solicite un número entre 0 y 999, y muestre cuantas cifras tiene.
8. Crear un programa al que se le pasa una temperatura corporal en grados centígrados. Si la temperatura es menor que 34 o mayor de 41, debe mostrar el siguiente mensaje: “Vuelva a tomar la temperatura, posible error”, si la temperatura está entre 34,1 y 37,1 muestre “temperatura correcta”, si la temperatura está entre 37,2 y 38,3 muestre “Acuda al médico, puede estar enfermo” si la temperatura está entre 38,4 y 40,9 muestre “URGENTE: acuda al médico”. Utilizar la estructura if-else.
9. Crear un programa en el que solicite un número entero del 1 al 7, y nos muestre por pantalla el correspondiente día de la semana. Utilizar la estructura if-else.
10. Crear un programa que pida una hora por teclado y que muestre buenos días, buenas tardes o buenas noches según la hora. Se utilizarán los tramos de 6 a 12, de 13 a 20 y de 21 a 5 respectivamente. Sólo se tienen en cuenta las horas, los minutos no se deben introducir por teclado.
11. Crear un programa en el que solicite un número entero del 1 al 7, y nos muestre por pantalla el correspondiente día de la semana. Utilizar la estructura switch.
12. Crear un programa en el que solicite un número entero del 1 al 12, y nos muestre por pantalla a que mes corresponde. Utilizar la estructura switch.
13. Crear un programa que solicite una nota del 1 al 10 al usuario, y le diga si es aprobado si 5 o más o suspenso menos de 5. Utilizar la estructura switch.
14. Crear un menú de una calculadora donde las opciones sean: 1. Sumar, 2. Restar, 3. Multiplicar, 4. Dividir. Se pedirán dos números al usuarios y según la opción seleccionada se realizará la operación y mostrará el resultado por pantalla.
15. Crea un programa que genere números aleatorios en [0,30] y los muestre por pantalla hasta que salga uno mayor que 25.
16. Crear un programa que muestre los números pares del 1 al 100 utilizando bucle for
17. Crear un programa que muestre los números pares del 1 al 100 utilizando bucle while

18. Crear un programa que muestre los números pares del 1 al 100 utilizando bucle do-while
19. Crear un programa que muestre los números del 550 al 340, contando de 20 en 20 hacia atrás utilizando bucle for
20. Crear un programa que muestre los números del 550 al 340, contando de 20 en 20 hacia atrás utilizando bucle while
21. Crear un programa que muestre los números del 550 al 340, contando de 20 en 20 hacia atrás utilizando bucle do-while
22. Crear un programa para acertar un número aleatorio de 3 cifras, para lo que tendremos 5 oportunidades.
23. Crear un programa que muestre la tabla de multiplicar de un número que se introduce por teclado.
24. Crear un programa que calcule la media de un conjunto de números positivos introducidos por teclado. A priori, el programa no sabe cuántos números se introducirán. El usuario indicará que ha terminado de introducir los datos cuando meta un número negativo.
25. Crear un programa que muestre los n primeros términos de la serie de Fibonacci. El primer término de la serie de Fibonacci es 0, el segundo es 1 y el resto se calcula sumando los dos anteriores, por lo que tendríamos que los términos son 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144... El número n se debe introducir por teclado.
26. Crear un programa que pida una base y un exponente (entero positivo) y que calcule la potencia. (No usar la clase Math, utilizar bucles)
27. Crear un programa que diga si un número introducido por teclado es o no primo. Un número primo es aquel que sólo es divisible entre él mismo y la unidad.
28. Crear un programa que obtenga los números enteros comprendidos entre dos números introducidos por teclado y validados como distintos, el programa debe empezar por el menor de los enteros introducidos e ir incrementando de 7 en 7.
29. Crear un programa que pinte una pirámide por pantalla. La altura se debe pedir por teclado. El carácter con el que se pinta la pirámide también se debe pedir por teclado.
30. Realiza un programa que vaya pidiendo números hasta que se introduzca un número negativo y nos diga cuantos números se han introducido, la media de los impares y el mayor de los pares. El número negativo sólo se utiliza para indicar el final de la introducción de datos pero no se incluye en el cómputo.
31. Escribe un programa que permita ir introduciendo una serie indeterminada de números mientras su suma no supere el valor 10000. Cuando esto último ocurra, se debe mostrar el total acumulado, el contador de los números introducidos y la media.
32. Crear una calculadora con un menú donde las opciones sean: 1. Sumar, 2. Restar, 3. Multiplicar, 4. Dividir, 0. Salir. Se pedirán dos números al usuarios y según la opción seleccionada se realizará la operación y mostrará el resultado por pantalla. Cuando realice la operación, deberá mostrar al usuario el menú, y volver a realizar la operación indicada, hasta que el usuario pulse 0.

33. Crear un programa que obtenga independientemente (no en el mismo bucle) la suma de los números pares y de los impares dentro de los valores del 1 al 40
34. Escribir un programa en Java que lea un número entero por el teclado e imprima todos los números impares menores que él.
35. Crear un programa que muestre por pantalla todos los números del 1 al 10 excepto el 5. Utilizar la sentencia `continue`.
36. Crear un programa que muestre los números pares del 1 al 10. Utilizar la sentencia `continue`.
37. Crear un programa para acertar un número aleatorio del 1 al 10, para lo que tendremos 5 oportunidades. Utilizar la sentencia `break`.
38. Escriba un programa que solicite del usuario un número N y luego muestre por pantalla la siguiente ejecución:
- ```
1
1 2
1 2 3
12 3 4
.....
1 2 3 4 5 6 .....N
```
39. Escriba un programa que lea un mes en número (1 para enero, 2 para febrero, etc.) y un año e indique el número de días de ese mes. Recuerde que un año es bisiesto si es divisible por cuatro, excepto cuando es divisible por 100, a no ser que sea divisible por 400. Así, 1900 no fue bisiesto, pero el año 2000 sí lo fue.
40. Realizar un programa que muestre por pantalla los primeros 5 números pares a partir de un número dado.
41. Transforma el siguiente bucle `for` en un bucle `while`.
- ```
for (int i=5; i<15;i++)
{
    System.out.println(i);
}
```
42. Ir pidiendo por teclado una serie de números enteros e irlos sumando. Se deja de pedir números al usuario cuando la cantidad supera el valor 50. Escribir por pantalla la suma final de todos los números introducidos.
43. Algoritmo que pida caracteres e imprima 'VOCAL' si son vocales y 'NO VOCAL' en caso contrario, el programa termina cuando se introduce un espacio.
44. Escribe un programa que pida el límite inferior y superior de un intervalo. Si el límite inferior es mayor que el superior lo tiene que volver a pedir. A continuación se van introduciendo números hasta que introduzcamos el 0. Cuando termine el programa dará las siguientes informaciones
- La suma de los números que están dentro del intervalo (intervalo abierto).
  - Cuantos números están fuera del intervalo.
  - Informa si hemos introducido algún número igual a los límites del intervalo.

45. Leer por teclado un número entero N no negativo y mostrar el factorial de todos los números desde 0 hasta N.
46. El factorial de un número entero se expresa mediante el símbolo '!' y se define de la siguiente forma:

- Si  $n = 0$  entonces  $0! = 1$
- Si  $n > 0$  entonces
- $n! = n * (n - 1) * (n - 2) * \dots * 3 * 2 * 1$
- Por ejemplo,  $n = 5$  entonces
- $5! = 5 * 4 * 3 * 2 * 1 = 120$

```
Introduce un número > 0: 10
0! = 1
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
```

47. Define un array de 10 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?

Valor	39	4			-5				32	
Índice	0	1	2	3	4	5	6	7	8	9

48. 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.
49. Crea un programa en el que se definan 2 arrays de 15 números, en el primero de ellos se almacenarán números aleatorios del 1 al 100, y el segundo se rellenará con el cuadrado de los números del primer array. Finalmente, mostrar el contenido de los dos arrays dispuesto en dos columnas: Número — Cuadrado.
50. Realiza un programa que pida 12 números enteros y que luego muestre esos números junto con la palabra “par” o “impar” según proceda.
51. Escribe un programa que genere 20 números enteros aleatorios entre 1 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.
52. Crear un programa para acertar un número aleatorio del 1 al 10, para lo que tendremos 5 oportunidades. Al final independientemente de si se ha acertado o no, se deberán mostrar todos los números que el usuario haya introducido (se pueden guardar en un array) y se mostrará también el número aleatorio generado.
53. Crear un programa en el que se genere un array con 50 números aleatorios del 1 al 1000, y muestre por pantalla el array completo cual es el número menor y el mayor.
54. Crear un programa que pida 10 números reales por teclado, los almacene en un array, y muestre la suma de todos los valores.
55. Crea un programa que pida 8 números enteros por teclado, los almacene en un array y luego muestre por separado la suma de todos los valores positivos y negativos.

56. Crea un programa que pida dos valores enteros N y M, luego cree un array de tamaño N, escriba M en todas sus posiciones y lo muestre por pantalla.
57. Crea un programa que pida dos valores enteros P y Q, luego cree un array que contenga todos los valores desde P hasta Q, y lo muestre por pantalla.
58. Crea un programa que cree un array con 100 números aleatorios entre 0 y 50, y luego le pida al usuario un valor R (entre 0 y 50). Por último, mostrará cuántos valores del array son igual o superiores a R.
59. Crea un programa que cree dos arrays de enteros de tamaño 100. Luego introducirá en el primer array todos los valores del 1 al 100. Por último, deberá copiar todos los valores del primer array al segundo array en orden inverso, y mostrar ambos por pantalla.
60. Crea un programa que cree un array de enteros e introduzca la siguiente secuencia de valores: 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, etc. hasta introducir 10 diez veces, y luego la muestre por pantalla.

**Nota:** Ninguna función mostrará nada por pantalla a no ser que se diga lo contrario.

61. Realiza una función que elimine los decimales de un número decimal. Pruébalo generando un número aleatorio con decimales y mostrando por pantalla el número sin decimales.
62. Escribe un programa que pida la edad por teclado y muestre por pantalla si eres mayor de edad o no. Implementa y utiliza la función:

```
boolean esMayorEdad(int a) // Devuelve verdadero si a>=18, falso en caso contrario
```

63. Escribe un programa que pida dos números enteros por teclado y muestre por pantalla cual es el mínimo. Implementa y utiliza la función:

```
int minimo(int a, int b) // Devuelve el menor entre a y b (no utilizar la clase Math)
```

64. Escribe un programa que pida cinco precios y muestre por pantalla el precio de venta de cada uno tras aplicarle un 21% de IVA. Implementa y utiliza la función:

```
double precioConIVA(double precio) // Devuelve el precio tras sumarle un 21% de IVA
```

65. Escribe un programa que pida el ancho y alto de un rectángulo y muestre por pantalla su área y su perímetro. Implementa y utiliza las funciones:

```
double perimetroRectangulo(double ancho, double alto) // Devuelve el perímetro
```

```
double areaRectangulo(double ancho, double alto) // Devuelve el área
```

66. Realiza un programa que lea una fecha introduciendo el día, mes y año por separado y nos diga si la fecha es correcta o no. Supondremos que todos los meses tienen 30 días. Se debe crear una función donde le pasemos los datos y devuelva si es correcta o no.
67. Realiza un programa que escriba la tabla de multiplicar de un número introducido por teclado. Para ello implementa una función que reciba como parámetro un número entero y muestre por pantalla la tabla de multiplicar de dicho número.
68. Escribe un programa que imprima las tablas de multiplicar del 1 al 10. Implementa una función que reciba un número entero como parámetro e imprima su tabla de multiplicar.

69. Escribe una función que muestre por pantalla un triángulo como el del ejemplo. Deberá recibir dos parámetros: el carácter que se desea imprimir y el número de líneas del triángulo.

A

AAA

AAAAA

70. . Escribe un programa que cree un array de tamaño 100 con los primeros 100 números naturales. Luego muestra la suma total y la media. Implementa una función que calcule la suma de un array y otra que calcule la media de un array.
71. Realiza una función que calcule el factorial de un número. En el programa, se solicitará al usuario el número sobre el que quiere calcular el factorial (el método solo realiza el factorial, no interacciona con el usuario)