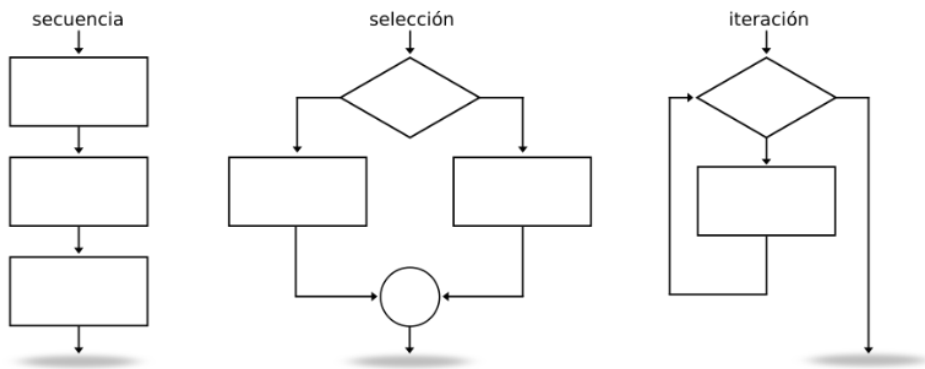


UD03

Ejercicios



Este material está bajo una [Licencia Creative Commons Atribución-Compartir-Igual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).
Derivado a partir de material de David Martínez Peña (<https://github.com/martinezpenya>).

1. Retos

2. Ejercicios

- 2.1. `if else`
- 2.2. Bucles simples
- 2.3. Bucles anidados
- 2.4. `switch`
- 2.5. en papel...
- 2.6. Trazas
- 2.7. Excepciones

3. Actividades

4. Fuentes de información

1. Retos

1. **Reto 1:** modificad el programa para que, en lugar de hacer un descuento del 8% si la compra es de 100 € o más, aplique una penalización de 2 € si el precio es inferior a 30 €.

```
import java.util.Scanner;
//Un programa que calcula descuento.

public class Descuento {
    public static final float DESCOMPTE = 8;
    public static final float COMPRA_MIN = 100;

    public static void main(String[] args) {
        Scanner lector = new Scanner(System.in);
        System.out.print("Cuál es el precio del producto, en euros? ");
        float preu = lector.nextFloat();
        lector.nextLine();
        if (preu >= COMPRA_MIN) {
            float descompteFet = preu * DESCOMPTE / 100;
            preu = preu - descompteFet;
        }
        System.out.println("El precio final para pagar es de " + preu + " euros.");
    }
}
```

2. **Reto 2:** modificad el programa para que, en lugar de un único valor secreto, haya dos. Para ganar, solo hay que acertar uno de los dos. La condición lógica que os hará falta ya no se puede resolver con una expresión compuesta por una única comparación. Será más compleja.

```
import java.util.Scanner;

public class Adivina {

    public static final int VALOR_SECRET = 4;

    public static void main(String[] args) {
        Scanner lector = new Scanner(System.in);
        System.out.println("Comencemos el juego.");
        System.out.print("Adivina el valor entero, entre 0 y 10: ");
        int valorUsuari = lector.nextInt();
        lector.nextLine();
        if (VALOR_SECRET == valorUsuari) {
            System.out.println("Exacto! Era " + VALOR_SECRET + ".");
        } else {
            System.out.println("Te has equivocado!");
        }
        System.out.println("Hemos terminado el juego.");
    }
}
```

3. **Reto 3:** modificad el ejemplo anterior (*Adivina*) para que compruebe que el valor que ha introducido el usuario se encuentra dentro del rango de valores correcto (entre 0 y 10).
4. **Repte 4:** aplicad el mismo tipo de control sobre los datos de la entrada del ejemplo siguiente al ejercicio del reto 1.

```
import java.util.Scanner;

public class AdivinaControlErrorsEntrada {

    public static final int VALOR_SECRET = 4;

    public static void main(String[] args) {
        Scanner lector = new Scanner(System.in);
        System.out.println("Comencemos el juego.");
        System.out.print("Adivina el valor entero, entre 0 y 10: ");
        boolean tipusCorrecte = lector.hasNextInt();
        if (tipusCorrecte) {
            //Se ha escrito un entero correctamente. Ya se puede leer.
            int valorUsuari = lector.nextInt();
            lector.nextLine();
            if (VALOR_SECRET == valorUsuari) {
                System.out.println("Exacto! Era " + VALOR_SECRET + ".");
            } else {
                System.out.println("Te has equivocado!");
            }
            System.out.println("Hemos terminado el juego.");
        } else {
            //No se ha escrito un entero.
            System.out.println("El valor introducido no es un entero.");
        }
    }
}
```

5. **Reto 5:** Modificad el ejemplo para que primero pregunte al usuario cuántos caracteres "-" quiere escribir por pantalla, y entonces los escriba. Cuando probéis el programa, no introducid un número muy alto!

```
//Un programa que escribe una línea con 100 caracteres ' - '.

public class Linea {

    public static void main(String[] args) {
        //Inicializamos un contador
        int i = 0;
        //Ja hemos hecho esto 100 veces
        while (i < 100) {
            System.out.print("-");
            //Lo hemos hecho una vez, sumamos 1 al contador
            i = i + 1;
        }
        //Forzamos un salto de línea
        System.out.println();
    }
}
```

6. **Reto 6:** un contador tanto puede empezar a contar desde 0 e ir subiendo, como desde el final e ir disminuyendo como una cuenta atrás. Modificad este programa para que la tabla de multiplicar empiece mostrando el valor para 10 y vaya bajando hasta el 1.

```
import java.util.Scanner;
public class TablaMultiplicar {

    public static void main(String[] args) {
        Scanner lector = new Scanner(System.in);
        System.out.print("Qué tabla de multiplicar quieres? ");
        int taula = lector.nextInt();
        lector.nextLine();
        int i = 1;
        while (i <= 10) {
            int resultat = taula * i;
            System.out.println(taula + " * " + i + " = " + resultat);
            i = i + 1;
        }
        System.out.println("Esta ha estado la tabla del " + taula);
    }
}
```

7. **Reto 7:** el uso de contadores y acumuladores no es excluyente, sino que puede ser complementario. Pensad cómo se podría modificar el programa para calcular el resultado del módulo y la división entera a la vez. Recordad que la división entera simplemente sería contar cuántas veces se ha podido restar el divisor.

```
import java.util.Scanner;

public class Modulo {

    public static void main(String[] args) {
        Scanner lector = new Scanner(System.in);
        System.out.print("Cuál es el dividendo? ");
        int dividend = lector.nextInt();
        lector.nextLine();
        System.out.print("Cual es el divisor? ");
        int divisor = lector.nextInt();
        lector.nextLine();
        while (dividend >= divisor) {
            dividend = dividend - divisor;
            System.out.println("Bucle: por ahora el dividendo vale " + dividend
+ ".");
        }
        System.out.println("El resultado final es " + dividend + ".");
    }
}
```

2. Ejercicios

2.1. if else

1. **Ejercicio 1** (*MenorDeDos*) Escribir un programa que muestre el menor de dos números enteros introducidos por teclado.
2. **Ejercicio 2** (*MenorDeTres*) Escribir un programa que muestre el menor de tres números enteros introducidos por teclado. Haz dos versiones: una utilizando los operadores lógicos necesarios (&&, |, ...) y otra sin utilizar ninguno (habrá que usar sentencias if else anidadas)
3. **Ejercicio 3** (*IntermedioDeTres*) Escribir un programa que muestre el intermedio de tres números introducidos por teclado.
4. **Ejercicio 4** (*NotasTexto*) Escribir un programa que acepte del usuario la nota de un examen (valor numérico entre 1 y 10) y muestre el literal correspondiente a dicha nota según (insuficiente, suficiente, bien, notable, sobresaliente).
5. **Ejercicio 5** (*LetraNif*) Escribir un programa que lea de teclado un nif (sin guiones ni puntos). Si el nif introducido lleva la letra, se comprobará si ésta es correcta y se le indicará al usuario si lo es o no. Si el nif no lleva letra, se calculará la que le corresponde y se mostrará al usuario. Te aconsejo que dividas el problema en partes: tieneLetra(), extraerUltimoChar(), extraeNumero(), validarLetra(), calcularLetra()...
6. **Ejercicio 6** (*División*) Escribir un programa que pida al usuario dos números enteros y le muestre el resultado de la división. Tener en cuenta que si dividimos un número por cero se producirá un error de ejecución y debemos evitarlo.
7. **Ejercicio 7** (*Raiz*) Se desea calcular la raíz cuadrada real de un número real cualquiera pedido inicialmente al usuario. Como dicha operación no está definida para los números negativos es necesario tratar, de algún modo, dicho posible error sin que el programa detenga su ejecución.
8. **Ejercicio 8** (*Hora12*) Escribir un programa que lea la hora de un día en notación de 24 horas y la exprese en notación de 12 horas. Por ejemplo, si la entrada es 13 horas 45 minutos, la salida será 1:45 PM. La hora y los minutos se leerán de teclado de forma separada, primero la hora y luego los minutos.
9. **Ejercicio 9** (*Bisiesto*) Escribir un programa que determine si un año introducido por teclado es o no bisiesto. Un año es bisiesto si es múltiplo de 4 (por ejemplo 1984). Sin embargo, los años múltiplos de 100 no son bisiestos, salvo que sean múltiplos de 400, en cuyo caso si lo son (por ejemplo 1800 no es bisiesto y 2000 si lo es). Para hacer el programa, implementa un método dentro de la clase que reciba un año y devuelva true si el año es bisiesto y false en caso de que no lo sea.
10. **Ejercicio 10** (*Fechas*) Escribir un programa que pida al usuario dos fechas (día, mes y año), que se suponen correctas, y le muestre la menor de ellas. La fecha se mostrará en formato dd/mm/año. Utiliza un método mostrarFecha, para mostrar la fecha por pantalla. La fecha se mostrará siempre con dos dígitos para el día, dos para el mes y cuatro para el año.
11. **Ejercicio 11** (*DiasDelMes*) Escribir un programa que lea de teclado el número de un mes (1 a 12) y visualice el número de días que tiene el mes. Hacerlo utilizando sentencias `if else`. Para hacer el programa, implementa un método en la clase que reciba un número de mes y devuelva el número de días que tiene el mes.

12. **Ejercicio 12** (*NombreDelMes*) Escribir un programa que lea de teclado el número de un mes (1 a 12) y visualice el nombre del mes (enero, febrero, etc). Hacerlo utilizando sentencias `if` `else`. Para hacer un programa, implementa un método en la clase que reciba un número de mes y devuelva el nombre del mes
13. **Ejercicio 13** (*Salario*) Escribir un programa que lea de teclado las horas trabajadas por un empleado en una semana y calcule su salario neto semanal, sabiendo que:
- Las horas ordinarias se pagan a 6 €.
 - Las horas extraordinarias se pagan a 10 €.
 - Los impuestos a deducir son:
 - Un 2 % si el salario bruto semanal es menor o igual a 350 €
 - Un 10 % si el salario bruto semanal es superior a 350 €
 - La jornada semanal ordinaria son 40 horas. El resto de horas trabajadas se considerarán horas extra.
14. **Ejercicio 14** (*Signo*) Dados dos números enteros, num1 y num2, realizar un programa que escriba uno de los dos mensajes:
- "el producto de los dos números es positivo o nulo" o bien
 - "el producto de los dos números es negativo".
- Resolverlo sin calcular el producto, sino teniendo en cuenta únicamente el signo de los números a multiplicar.
15. **Ejercicio 15** (*Calculadora*) Escribir un programa para simular una calculadora. Considera que los cálculos posibles son del tipo num1 operado num2, donde num1 y num2 son dos números reales cualesquiera y operador es una de entre: +, -, * y /. El programa pedirá al usuario en primer lugar el valor num1, a continuación el operador y finalmente el valor num2. Resolver utilizando instrucciones `if` `else`
16. **Ejercicio 16** (*Comercio*) Un comercio aplica un descuento del 8% por compras superiores a 40 euros. El descuento máximo será de 12 euros. Escribir un programa que solicite al usuario el importe de la compra y muestre un mensaje similar al siguiente:
- Importe de la compra 100 €
 - Porcentaje de descuento aplicado: 8%
 - Descuento aplicado: 8 €
 - Cantidad a pagar: 92 €
17. **Ejercicio 17** (*Editorial*) Una compañía editorial dispone de 2 tipos de publicaciones: libros y revistas. El precio de cada pedido depende del número de elementos solicitados al cual se le aplica un determinado descuento, que es diferente para libros y para revistas. La siguiente tabla muestra los descuentos a aplicar en función del número de unidades y del tipo de producto:

Cantidad pedida	Libros	Revistas
Hasta 5 unidades	0 % de descuento	0 % de descuento
De 6 a 10 unidades	10 % de descuento	15 % de descuento
De 11 a 20 unidades	15 % de descuento	20 % de descuento

Escribe un método `calcularCoste` que, recibiendo el tipo de publicación (`String`), que puede ser "libro" o "revista", el precio individual (`double`) y el número de unidades solicitado (`int`), devuelva el coste del pedido (aplicando el descuento correspondiente).

Escribe un programa en el que el usuario indique cantidad y precio de revistas y cantidad y precio de libros que incluye un pedido, y muestre el coste del pedido

18. **Ejercicio 18 (Taxi)** Se desea calcular el coste del trayecto realizado en taxi en función de los kilómetros recorridos en las carreras metropolitanas de Valencia. Según las tarifas vigentes para el 2012, el coste se calcula de la siguiente manera:

- Días laborables en horario diurno (de 6:00 a 22:00h): 0.73 €/km.
- Días laborables en horario nocturno: 0.84 €/km.
- Sábados y domingos: 0.93 €/km.
- Además, la tarifa mínima diurna es de 2.95€ y la mínima nocturna de 4€.

Escribir un programa que solicite al usuario:

- La hora (hora y minutos) en que se realizó el trayecto.
- El día de la semana (se supone que el usuario introduce un valor entre 1 para lunes y 7 para domingo)
- Los kilómetros recorridos.

Y muestre el coste del trayecto

19. **Ejercicio 19 (Nombre)** Escribir un programa en el que el usuario pueda escribir su nombre. El programa le dirá si la primera y la última letra del nombre coinciden o no. Pruébalo con "Ana", "ana", "Angel", "Amanda" y "David"

20. **Ejercicio 20 (Validar)** Se desea implementar un programa que determine si dos datos `x` e `y` de entrada son válidos. Un par de datos es válido si es uno de los que aparecen en la siguiente tabla:

x :	a	a	a	a	a	b	b	b	b	b
y :	1	3	5	7	9	2	4	6	8	10

Se pide implementar un programa que lea de teclado el valor de `x` y el valor de `y`, e indique por pantalla "VALIDOS" o "NO VALIDOS". Se pide hacerlo de forma que no se utilice ninguna estructura condicional (`if`, `switch`,...), es decir, se calculará una expresión booleana que determine si `x` e `y` son válidos. Se procurará que la expresión booleana propuesta sea breve y concisa.

2.2. Bucles simples

21. **Ejercicio 21.** Crear una clase llamada `SencillosWhile` y crear en él métodos que realicen las siguientes tareas.

1. (`imparesHastaN`) Dado un nº entero `n` introducido por el usuario, mostrar los números impares que hay entre 1 y `n`. Por ejemplo, si `n` es 8 mostrará 1 3 5 7
2. (`nImpares`) Dado un nº entero `n` introducido por el usuario, mostrar los `n` primeros números impares. Por ejemplo, si `n` es 3 mostrará 1 3 5 (3 primeros impares)
3. (`cuentaAtras`) Dado un entero `n` introducido por el usuario, mostrar una cuenta atrás partiendo de `n`: `n`, `n-1`, 5, 4, 3, 2, 1, 0
4. (`sumaNPrimeros`) Dado un entero `n` introducido por el usuario, mostrar la suma de los

5. (mostrarDivisoresN) Dado un entero `n` introducido por el usuario, mostrar todos sus divisores, incluidos el 1 y el mismo `n`. Por ejemplo, si `n` es 12 mostraría 1, 2, 3, 4, 6 y 12
6. (sumaDivisoresN) Dado un entero `n` introducido por el usuario, mostrar la suma de todos sus divisores, sin incluir al propio `n`. Por ejemplo, si `n` es 12 sumará 1, 2, 3, 4 y 6 = 16
22. **Ejercicio 22.** Crear una clase llamada "SencillosFor" y crear en él los mismos métodos que en el ejercicio anterior, pero utilizando la sentencia `for` en lugar de `while`
23. **Ejercicio 23.** (PotenciasDe2) Dado un entero `n` introducido por el usuario, mostrar las `n` primeras potencias de 2. Es decir, 2^0 , 2^1 , 2^2 , 2^3 , ..., 2^n . Soluciona el ejercicio sin utilizar `Math.pow`. Ten en cuenta que, por ejemplo, $2^3 = 1 * 2 * 2 * 2$ o que $2^4 = 1 * 2 * 2 * 2 * 2$
24. **Ejercicio 24.** (Etapas) El ser humano pasa por una serie de etapas en su vida que, con carácter general se asocian a las edades que aparecen en la tabla siguiente.

Infancia	Hasta los 10 años
Pubertad	De 11 a 14 años
Adolescencia	De 15 a 21 años
Adultez	De 22 a 55 años
Vejez	De 55 a 70 años
Ancianidad	A partir de los 71 años

Escribe un programa en el que el usuario introduzca las edades de una serie de personas y calcule y muestre que porcentaje de personas que se encuentran en cada etapa. En primer lugar el programa pedirá el número de personas que participan en la muestra y a continuación solicitará la edad de cada una de ellas. El resultado será similar al siguiente:

```

Infancia: 5.3 %
Pubertad: 10.7 %
Adolescencia: 21.2 %
...
```

25. **Ejercicio 25.** (Primo) Escribir un programa en el que el usuario escriba un número entero y se le diga si se trata o no de un número primo. Recuerda que un n° primo es aquel que solo es divisible por 1 y por sí mismo.
26. **Ejercicio 26.** (Primos) Escribir un programa en el que el usuario escriba un número entero y se le diga todos los números primos entre 1 y el número introducido.
27. **Ejercicio 27.** (esPrimoMejorada) Haz una nueva versión del programa del ejercicio anterior teniendo en cuenta lo siguiente:
- El único número par que es primo es el 2.
 - Un número n no puede tener divisores mayores que $n/2$
28. **Ejercicio 28.** (Divisores) Escribir un programa que muestre los tres primeros divisores de un número n introducido por el usuario. Por ejemplo, si el usuario introduce el número 45, el programa mostrará los divisores 1, 3 y 5. Ten en cuenta que la posibilidad de que el número

29. **Ejercicio 29.** (*SumaSerie*) Dado un número `n`, introducido por el usuario, calcula y muestra por pantalla la siguiente suma $1/1 + 1/2 + 1/3 + \dots + 1/n$
30. **Ejercicio 30.** Escribir un programa en el que el usuario introduzca un número entero cualquiera (positivo, negativo o cero) y se le diga cuantas cifras tiene. Pistas: ¿Cuántas cifras tiene el nº 25688? ¿Cuántas veces podemos dividir el nº 25688 por 10 hasta que se hace cero? Cuidado, el nº 0 tiene una cifra.
31. **Ejercicio 31.** (*Transportes*) Una empresa de transportes cobra 30€ por cada bulto que transporta. Además, si el peso total de todos los bultos supera los 300 kilos, cobra 0.9€ por cada kg extra. Por último si el transporte debe realizarse en sábado, cobra un plus de 60€. La empresa no realiza el pedido si hay que transportar más de 30 bultos, si el peso total supera los 1000 kg o si se solicita hacerlo en domingo. Realizar un programa que solicite el número de bultos, el día de la semana (valor entre 1 y 7) y el peso de cada uno de los bultos y muestre el coste del transporte en caso de que pueda realizarse o un mensaje adecuado en caso contrario
32. **Ejercicio 32.** (*Containers*) La capacidad de un buque que transporta containers está limitada tanto por la cantidad de containers como por el peso, pudiendo transportar un máximo de 100 containers y un máximo de 700 toneladas. Hacer un programa en el que se vaya introduciendo el peso de los containers (en toneladas) a medida que se cargan en el barco, hasta que se llegue al máximo de capacidad. Mostrar al final la cantidad de containers cargados y el peso total. En el momento en que se desee cargar un container que haga que la carga total supere las 700 toneladas, se dará por finalizada la carga, aunque pudieran existir containers menos pesados con posibilidad de ser cargados.
33. **Ejercicio 33.** (*Notas*) Realizar un programa que permita introducir las notas de un examen de los alumnos de un curso. El usuario irá introduciendo las notas una tras otra. Se considerará finalizado el proceso de introducción de notas cuando el usuario introduzca una nota negativa. Al final, el programa le dirá:
- El número de notas introducidas.
 - El número de aprobados (mayor o igual a 5 puntos)
 - La nota media
34. **Ejercicio 34.** (*NotasExtremas*) Modificar el ejercicio anterior para que además calcule la nota máxima y la nota mínima.

2.3. Bucles anidados

35. **Ejercicio 35.** (*Edades*) Programa que pida al usuario la edad de cinco personas. Si la suma de las edades es inferior a 200, el programa volverá a solicitar las 5 edades.
36. **Ejercicio 36.** (*NotasPorAlumno*) Programa que pida al usuario las notas de `A` alumnos en `S` asignaturas, alumno por alumno. `A` y `S` se definirán en el programa como `CONSTANTES`.

```
Alumno 1
Introduce nota de asignatura 1: 8
Introduce nota de asignatura 2:
...
Alumno 2
Introduce nota de asignatura 1:
...
```

37. **Ejercicio 37.** (*NotasPorAsignatura*) Programa que pida al usuario las notas de **A** alumnos en **S** asignaturas, asignatura por asignatura. **A** y **S** se definirán en el programa como **CONSTANTES**.

```
Asignatura 1
Introduce nota del alumno 1:
Introduce nota del alumno 2:
...
Asignatura 2
Introduce nota del alumno 1:
...
```

38. **Ejercicio 38.** (*MediasPorAsignatura*) Repite el ejercicio anterior haciendo que se muestre la media de cada asignatura

```
Asignatura 1
Introduce nota del alumno 1:
Introduce nota del alumno 2:
...
Media asignatura 1: 8.5 puntos

Asignatura 2
Introduce nota del alumno 1:
...
Media asignatura 2: 6.5 puntos
...
```

39. **Ejercicio 39.** (*TablaMult*) Escribir un programa que permita al usuario introducir un número **N** e imprima la tabla de multiplicar (del 0 al 10) de todos los números entre 1 y **N**. Ejemplo: Si el usuario introduce en número 5, el programa imprimiría

```
Tabla del 1:
1 por 0, 0
1 por 1, 1
1 por 2, 2
...
1 por 10, 10

Tabla del 2:
2 por 0, 0
2 por 1, 2
...
2 por 10, 20

Tabla del 3:
...

Tabla del 5:
...
5 por 10, 50
```

41. **Ejercicio 41.** (*CombinarLetras2*) Escribir un programa que muestre todas las palabras de dos letras que se pueden formar con los cuatro primeros caracteres del alfabeto en minúsculas ('a', 'b', 'c', 'd'):

```
aa
ab
ac
ad
ba
bb
bc
bd
...
da
db
dc
dd
```

42. **Ejercicio 42.** (*CombinarLetras3*) Repite el ejercicio anterior mostrando palabras de tres letras

```
aaa
aab
...
ddc
ddd
```

43. **Ejercicio 43.** (*LetraALetra*) Escribe un programa en el que se solicite al usuario un texto de forma repetida hasta que el usuario introduzca la cadena vacía. Con cada texto que introduzca el usuario se le mostrará carácter a carácter, cada carácter en una línea

```
Introduce texto: Hola
H
o
l
a
Introduce texto: Casa
C
a
s
a
Introduce texto:
Fin del programa
```

44. **Ejercicio 44.** (*DibujarFiguras1*) Escribe una clase que contenga los métodos que se indican a continuación. En el método main solicita al usuario las dimensiones de las figuras necesarias en cada caso y llama al método correspondiente para que se muestre por pantalla

1. (`void dibRecAsteriscos (int ancho, int alto)`) dibuja un rectángulo utilizando asteriscos, como el siguiente. En el ejemplo ancho es 7 y alto es 3

```
* * * * *
* * * * *
* * * * *
```

2. `(void dibRecNumeros1 (int ancho, int alto))` dibuja un rectángulo utilizando números, como el siguiente. En el ejemplo ancho es 7 y alto es 3

```
1 2 3 4 5 6 7
1 2 3 4 5 6 7
1 2 3 4 5 6 7
```

3. `(void dibRecNumeros2 (int ancho, int alto))` dibuja un rectángulo utilizando números, como el siguiente. En el ejemplo ancho es 7 y alto es 3

```
7 6 5 4 3 2 1
7 6 5 4 3 2 1
7 6 5 4 3 2 1
```

4. `(void dibRecNumeros3 (int ancho, int alto))` dibuja un rectángulo utilizando números, como el siguiente. En el ejemplo ancho es 7 y alto es 3

```
01 02 03 04 05 06 07
08 09 10 11 12 13 14
15 16 17 18 19 20 21
```

5. `(void dibDiagonal (int ancho, int alto))` dibuja un rectángulo con ceros y unos. Los 1 están en las posiciones en las que fila y columna coinciden. En el ejemplo ancho es 7 y alto es 3

```
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
```

6. `(void dibRecLetras (int ancho, int alto))` dibuja un rectángulo letras sucesivas comenzando por la "a". En el ejemplo ancho es 7 y alto es 3

```
a a a a a a a
b b b b b b b
c c c c c c c
```

7. `(void dibRecLetras2 (int ancho, int alto))` dibuja un rectángulo letras sucesivas terminando por la "a". En el ejemplo ancho es 7 y alto es 3

```
c c c c c c c
b b b b b b b
a a a a a a a
```

8. `(void dibRecLetras3 (int ancho, int alto))` dibuja un rectángulo letras sucesivas comenzando por la "a". En el ejemplo ancho es 7 y alto es 3

```
a b c d e f g
h i j k l m n
o p q r s t u
```

45. **Ejercicio 45.** (*dibujarFiguras2*) Escribe una clase que contenga los métodos que se indican a continuación. En el método main solicita al usuario las dimensiones de las figuras necesarias en cada caso y llama al método correspondiente para que se muestre por pantalla

1. `void dibRectNumeros3 (int ancho, int alto)` dibuja un rectángulo utilizando números, como el siguiente. En el ejemplo ancho es 7 y alto es 3

```
1 2 3 4 5 6 7 7 6 5 4 3 2 1
1 2 3 4 5 6 7 7 6 5 4 3 2 1
1 2 3 4 5 6 7 7 6 5 4 3 2 1
```

2. `void dibRectAsteriscos1 (int ancho, int alto)` dibuja un rectángulo utilizando asteriscos (*) y espacios en blanco, como el siguiente. En el ejemplo ancho es 7 y alto es 3

```
* * * * *
* * * * *
* * * * *
```

3. `void dibRectAsteriscos2 (int ancho, int alto)` dibuja un rectángulo utilizando asteriscos (*), espacios en blanco y el carácter '+', como el siguiente. En el ejemplo ancho es 7 y alto es 3

```
* + * + * + *
* + * + * + *
* + * + * + *
```

4. `void dibRectAsteriscos3 (int ancho, int alto)` dibuja un rectángulo utilizando asteriscos (*) y espacios en blanco, como el siguiente. En el ejemplo ancho es 7 y alto es 3

```
* * * * *
*           *
* * * * *
```

5. `void dibTriangulo1 (int base)` dibuja un triángulo utilizando asteriscos (*) y espacios en blanco, como el siguiente. En el ejemplo base es 5

```
*
* *
* * *
* * * *
* * * * *
```

6. `void dibTriangulo2 (int altura)` dibuja un triángulo utilizando asteriscos (*) y espacios en blanco, como el siguiente. En el ejemplo altura es 5

```

      *
    * *
  * * *
* * * *
* * * * *

```

7. `void dibTriangulo3 (int altura)` dibuja un triángulo utilizando asteriscos (*) y espacios en blanco, como el siguiente. En el ejemplo altura es 5

```

      *
    * * *
  * * * * *
* * * * * *
* * * * * * *

```

2.4. switch

46. **Ejercicio 46.** (*NotasTexto2*) Escribir un programa que acepte del usuario la nota de un examen (valor numérico entre 1 y 10) y muestre el literal correspondiente a dicha nota según (insuficiente, suficiente, bien, notable, sobresaliente). Hacerlo utilizando la sentencias switch. La nota que introduce el usuario tendrá que ser un valor entero.
47. **Ejercicio 47.** (*DiasDelMes2*) Escribir un programa que lea de teclado el número de un mes (1 a 12) y visualice el número de días que tiene el mes. Resolver utilizando la sentencias switch.
48. **Ejercicio 48.** (*NombreDelMes2*) Escribir un programa que lea de teclado el número de un mes (1 a 12) y visualice el nombre del mes (enero, febrero, etc). Resolver utilizando la sentencias switch.
49. **Ejercicio 49.** (*Calculadora2*) Escribir un programa para simular una calculadora. Considera que los cálculos posibles son del tipo num1 operado num2, donde num1 y num2 son dos números reales cualesquiera y operador es una de entre: +, -, * y /. El programa pedirá al usuario en primer lugar el valor num1, a continuación el operador y finalmente el valor num2. Resolver utilizando la sentencias switch.

2.5. en papel...

50. **Ejercicio 50.** ¿Qué valor se asignará a consumo en la sentencia `if` siguiente si velocidad es 120?

```

if (velocidad > 80)
    consumo = 10;
else if (velocidad > 100)
    consumo = 12;
else if (velocidad > 120)
    consumo = 15;

```

51. **Ejercicio 51.** Encuentra y corrige los errores de los siguientes fragmentos de programa .
52. fragmento a

```

if x > 25
    y = x
else

```

53. fragmento b

```

if (x<0)
    System.out.println("El valor de x es" +x);
    System.out.println ("x es negativo");
else
    System.out.println ("El valor de x es"+x);
    System.out.println ("x es positivo");

```

54. fragmento c

```

if (x = 0) System.out.println ("x igual a cero");
else System.out.println ("x distinto de cero");

```

55. **Ejercicio 52.**Cuál es la salida exacta por pantalla del siguiente fragmento de programa

```

int x = 20;
System.out.println("Comenzamos");
if (x>= 20)
    if (x>50) System.out.println("Muy grande");
    else {
        if (x%2 != 0) System.out.println("Impar");
    }
else if (x<=20) System.out.println("Pequeño");
System.out.println("Terminamos");

```

53. **Ejercicio 53.** En una tienda, por liquidación, se aplican distintos descuentos en función del total de las compras realizadas:

- Si total < 500 €, no se aplica descuento.
- Si 500 € ≤ total ≤ 2000 €, se aplica un descuento del 30 %.
- Si total > 2000€, entonces se aplica un descuento del 50 %

¿Cual de los siguientes fragmentos de programa asigna a la variable desc el descuento correcto?
Indica "Si" o "NO" al lado de cada fragmento

1. fragmento a

```

double desc = 0.0;
if (total <= 500)
    if (total >= 2000) desc = 30.0;
    else desc = 50.0;
total = total * desc / 100.0;

```

2. fragmento b

```

double desc = 0.0;
if (total >= 500)
    if (total <= 2000) desc = 30.0;
    else desc = 50.0;
total = total * desc / 100.0;

```



```
double desc = 0.0;
if (total <= 2000){
    if (total >= 500) desc = 30.0;
    } else desc = 50.0;
total = total * desc / 100.0;
```

4. fragmento d

```
double desc = 0.0;
if (total > 500)
    if (total < 2000) desc = 30.0;
    else desc = 50.0;
total = total * desc / 100.0;
```

5. **Ejercicio 54.** ¿Qué salida producirá el siguiente fragmento de programa si la variable entera platos vale 1? ¿Y si vale 3? ¿Y si vale 0?

```
switch (platos ) {
    case 1: System.out.println("\nPrimer plato");
    case 2: System.out.println ("\nSegundo plato");
    case 3: System.out.println ("\nBebida");
            System.out.println ("\nPostre");
            break;
    default: System.out.println("\nCafé");
}
```

55. **Ejercicio 55.** Dados tres enteros a, b y c, y un booleano p, el siguiente análisis por casos establece el valor de p en función de los valores de a, b y c:

```
si a > b entonces p = cierto;
si a < b entonces p = falso;
si a = b entonces
    si a > c entonces p = cierto;
    si a < c entonces p = falso;
    si a = c entonces p = falso;
```

Se pide la traducción de dicho análisis por casos a Java mediante:

- Una única instrucción if sin anidamientos.
- Una única instrucción, de la forma p = ... , sin sentencias if ni utilizar el operador ternario.
- Una única instrucción, de la forma p = ... , que utilice el operador ternario.

2.6. Trazas

56. **Ejercicio 56.** Indica cual será la salida producida por los siguientes programas, teniendo en cuenta los datos de entrada:

56.1) **Datos de entrada: 2, 5**

```
1. public static void main (String[] args){
    Scanner tec = new Scanner(System.in);
    int x,y,a;
    x = tec.nextInt();
    y = tec.nextInt();
    a = x+y;
    System.out.println(a);
}
```

```
2. public static void main (String[] args){
    Scanner tec = new Scanner(System.in);
    int x,a;
    x = tec.nextInt();
    x = tec.nextInt();
    a= x+x;
    System.out.println(a);
}
```

```
3. public static void main (String[] args){
    Scanner tec = new Scanner(System.in);
    int x,y,a;
    x = tec.nextInt();
    y = tec.nextInt();
    a = x+y;
    a = x*y;
    System.out.println(a);
}
```

```
4. public static void main (String[] args){
    Scanner tec = new Scanner(System.in);
    int x,y,a;
    x = tec.nextInt();
    y = tec.nextInt();
    a = x+y;
    System.out.println(a);
    a = x*y;
    System.out.println(a);
}
```

```
5. public static void main (String[] args){
    Scanner tec = new Scanner(System.in);
    int x,y,a;
    x = tec.nextInt();
    y = tec.nextInt();
    a = x+y;
    a = a+x+y;
    a = a+a;
    System.out.println(a);
}
```

```

6. public static void main (String[] args){
    Scanner tec = new Scanner(System.in);
    int x,y,a;
    x = tec.nextInt();
    y = tec.nextInt();
    a = x;
    a = doble(x);
    System.out.format ("%d\n%d\n%d", x,y,a);
}
public static int doble(int num){
    return 2*num;
}

```

```

7. public static void main (String[] args) {
    Scanner tec = new Scanner(System.in);
    int x,y,a;
    x = tec.nextInt();
    y = tec.nextInt();
    a = x;
    doble(a);
    System.out.format ("%d\n%d\n%d\n", x,y,a);
}
public static void doble(int x){
    x = 2*x;
}

```

```

8. public static void main (String[] args){
    Scanner tec = new Scanner(System.in);
    int x,y,a;
    x = tec.nextInt();
    y = tec.nextInt();
    a = calcular(y,x);
    System.out.format ("%d\n%d\n%d\n", x,y,a);
}
public static int calcular (int x, int y){
    return x-y;
}

```

```

9. public static void main (String[] args){
    Scanner tec = new Scanner(System.in);
    int x,y,a;
    x = tec.nextInt();
    y = tec.nextInt();
    y = calcular(x);
    a = calcular(y);
    System.out.format ("%d\n%d\n%d\n", x,y,a);
}
public static int calcular (int x){
    return x*x;
}

```

```

public static void main (String[] args){
    int k,l,m,x,y,z;
    k = tec.nextInt();
    l = tec.nextInt();
    m = tec.nextInt();
    x = k+l;
    if (x != m) {
        y = k*l;
        z = 0;
    } else {
        y = 0;
        z = k-l;
    }
    if (z < 0) z = -z;
    System.out.format("%d\n%d\n%d\n",x,y,z);
}

```

56.3) **Datos de entrada: 2, 5, 7, 9, -9, -7, -5, -2**

1.

```

public static void main (String[] args){
    int x,y;
    x = 0;
    y = tec.nextInt();
    while(!(y<0)) {
        x+=-y;
        y = tec.nextInt();
        System.out.format("%d, %d",x,y);
    }
}

```

2.

```

public static void main (String[] args){
    int x,y,z,a;
    x = y = z = a = 0;
    x = tec.nextInt();
    while(x>0) {
        if (y < z) y = tec.nextInt();
        else z= tec.nextInt();
        a = a-x+y*z;
        x = tec.nextInt();
        System.out.format("%d, %d, %d, %d",a,x,y,z);
    }
}

```

56.4) **Datos de entrada: 5, 5, 7, -5, -4, 2**

```
1. public static void main (String[] args){
    int x, y, a=0;
    x = 0;
    y = 99;
    while (x >= 0) {
        x = tec.nextInt();
        y = tec.nextInt();
        a = a + x*y;
    }
    System.out.println(a);
}
```

```
2. public static void main (String[] args){
    int x, y, a=0;
    x = 0;
    y = 99;
    while (x >= 0 && y >= 0) {
        x = tec.nextInt();
        y = tec.nextInt();
        a = a + x*y;
    }
    System.out.println(a);
}
```

```
3. public static void main (String[] args){
    int x, y, a=0;
    x = 0;
    y = 99;
    while (x >= 0 && y <= 0) {
        x = tec.nextInt();
        y = tec.nextInt();
        a = a + x*y;
    }
    System.out.println(a);
}
```

```
4. public static void main (String[] args){
    int x, y, a=0;
    x = 0;
    y = 99;
    while (x >= 0 || y >= 0) {
        x = tec.nextInt();
        y = tec.nextInt();
        a = a + x*y;
    }
    System.out.println(a);
}
```

56.5) Datos de entrada: 5, 5, 7, -5, -4, 2

```

public static void main(String[] args) {
    int x, y;

    x = 2;
    y = 3;
    while (x + y > 0) {
        x = tec.nextInt();
        y = tec.nextInt();
        x += y;
        y = x - y;
        System.out.format("%d, %d", x, y);
    }
}

```

56.7) Datos de entrada: 2, 4, 7, 5, -6, -3, 6, 6

1.

```

public static void main (String[] args){
    int a,b;
    do{
        a = tec.nextInt();
        b = tec.nextInt();
        for (int i=a ; i<=b ; i++)
            System.out.println(i);
    } while (a!=b)
}

```

2.

```

public static void main (String[] args){
    int a,b;
    a=5;
    b=5;
    do {
        for (int i=a ; i<=b ; i++)
            System.out.println(i);
        a = tec.nextInt();
        b = tec.nextInt();
    } while (a!=b);
}

```

56.8) Datos de entrada: 3, 3, 5, 5, -3, -7, 2, 2

```

public static void main (String[] args){
    int x,y;
    do {
        x = tec.nextInt();
        b = tec.nextInt();
    } while (x==y);
    if (x>y) {
        x=y;
        y=x;
    }
    System.out.format("%d %d %n", x, y);
}

```

56.9) Datos de entrada: 3, 2, 1, 4

```
1. public static void main (String[] args){
    int a=0,b;
    b = tec.nextInt();
    for(int i=1;i<=b,i++) a=(a+i)*i;
    System.out.println(a);
}
```

2. Datos de entrada:

```
public static void main (String[] args){
    int x,y;
    for (x=3;x>=1;x--){
        for(y=1;y<=x;y++) System.out.println(x);
        System.out.println();
    }
}
```

3. Datos de entrada:

```
public static void main (String[] args){
    int x,y;
    x=0;
    y=0;
    for (int i=1;i<=2;i++) {
        for (int j=1;j<=3;j++) x=(x+i)*j;
        y+=x;
    }
    System.out.println("%d %d %n",x,y);
}
```

4. Datos de entrada: 4, 5, 6, 7, 8, 9

```
public static void main (String[] args){
    int x,y;
    do x = tec.nextInt();
    while (x<=5);
    y=0;
    for (int i=12;i>=x;i-=2) y+=(x*i);
    System.out.println(y);
}
```

2.7. Excepciones

57. **Ejercicio 57. (01Edades)** Escribe un programa que solicite al usuario la edad de cinco personas y calcule la media. La edad de una persona debe ser un valor entero comprendido en el rango [0,110]. Realiza tres versiones:

1. Si se introduce mal la edad de una persona se vuelve a pedir la edad de esa persona.
2. Si se introduce mal la edad de una persona, el programa muestra un mensaje de error, no calcula la media y termina.

3. Si se introduce mal la edad de una persona, el programa vuelve a solicitar la edad de las cinco personas (comienza el proceso).

58. **Ejercicio 58.** Escribe programas que se indican a continuación. Ejecuta cada programa haciendo que la entrada del usuario provoque una excepción. Anota el nombre de la excepción que se produce y cuál es la jerarquía de objetos de la que descende:

1. Programa que solicita dos números enteros (a y b) y muestra el resultado de su división (a/b).

1. El usuario introduce 0 como valor de b.
2. El usuario introduce letras cuando el programa espera números enteros.
3. El usuario introduce un número real cuando el programa espera un entero.

2. Programa que solicita al usuario su nombre y una posición dentro del nombre. Se muestra al usuario la letra del nombre cuya posición se ha indicado. Por ejemplo:

```
Introduce nombre: Javi
Introduce posición: 2
En la posición 2 de Javi está la letra a
```

3. El usuario introduce una posición inválida.

59. **Ejercicio 59.** Repite el ejercicio anterior utilizando métodos y llamándolos desde el método `main`:

1. Un método `dividir` que devuelva el cociente de dos números que recibe como parámetro
2. Un método `letraNombre` que, dados un String `nombre` y un entero `pos`, devuelva el carácter del nombre que ocupa la posición indicada.

Ejecuta los programas provocando errores (como en el ejercicio anterior) y observa los mensajes que se generan.

60. **Ejercicio 60.** Escribir un programa que divida dos números que se reciben en `main` en `args[0]` y `args[1]`.

Ejemplo:

```
$ java dividir 10 5
10/5 es igual a 2
```

Donde 10 y 5 son `args[0]` y `args[1]` respectivamente, es decir los parámetros con que llamamos al programa `dividir`.

61. **Ejercicio 61.** Justifica por qué se produce error en el siguiente fragmento de código

```
try {
    System.out.println("Introduce edad: ");
    int edad = tec.nextInt();
    if (edad >= 18) {
        System.out.println("Mayor edad");
    } else {
        System.out.println("Menor edad");
    }
    System.out.println("Introduce nif");
}
```



```

    char letra = nif.charAt(nif.length() - 1);
    System.out.println("Numero: " + numero);
    System.out.println("Letra: " + letra);
} catch (Exception e){
    System.out.println("Debías introducir un número");
} catch (NumberFormatException e) {
    System.out.println("El nif es incorrecto");
}

```

62. **Ejercicio 62.** Indica qué se mostrará por pantalla cuando se ejecute esta clase y por qué:

```

public class Uno {
    private static int metodo() {
        int valor=0;
        try {
            valor = valor + 1;
            valor = valor + Integer.parseInt("42") ;
            valor = valor + 1;
            System.out.println("Valor al final del try: " + valor);
        } catch (NumberFormatException e) {
            valor = valor + Integer.parseInt ("42");
            System.out.println("Valor al final del catch: " + valor) ;
        }
        finally {
            valor = valor + 1;
            System.out.println("Valor al final de finally: " + valor) ;
        }
        valor = valor + 1;
        System.out.println ("Valor antes del return: " + valor) ;
        return valor;
    }

    public static void main(String[] args) {
        try {
            System.out.println (metodo());
        } catch (Exception e) {
            System.err.println("Excepcion en metodo()") ;
            e.printStackTrace();
        }
    }
}

```

63. **Ejercicio 63.** Indica qué se mostrará por pantalla cuando se ejecute esta clase y por qué:

```

public class Dos {
    private static int metodo() {
        int valor=0;
        try {
            valor = valor+1;
            valor = valor + Integer.parseInt("W");
            valor = valor + 1;
            System.out.println("Valor al final del try: " + valor);
        } catch (NumberFormatException e) {

```

```

        System.out.println("Valor al final del catch: " + valor) ;
    } finally {
        valor = valor + 1;
        System.out.println("Valor al final de finally: " + valor) ;
    }
    valor = valor + 1;
    System.out.println ("Valor antes del return: " + valor) ;
    return valor ;
}

public static void main (String[] args) {
    try {
        System .out.println(metodo());
    } catch (Exception e) {
        System.err.println("Excepcion en metodo() ");
        e.printStackTrace();
    }
}
}

```

64. **Ejercicio 64.** Indica qué se mostrará por pantalla cuando se ejecute esta clase y por qué:

```

public class Tres {
    private static int metodo() {
        int valor = 0;
        try {
            valor = valor +1;
            valor = valor + Integer.parseInt("W");
            valor = valor + 1;
            System.out.println("Valor al final del try : " + valor);
        } catch (NumberFormatException e) {
            valor = valor + Integer.parseInt("W");
            System.out.println("Valor al final del catch : " + valor);
        } finally {
            valor = valor + 1;
            System.out.println("Valor al final de finally: " + valor);
        }
        valor = valor + 1;
        System.out.println ("Valor antes del return: " + valor);
        return valor ;
    }

    public static void main (String[ ] args)
    {
        try {
            System.out.println(metodo ());
        } catch (Exception e) {
            System.err.println("Excepcion en metodo()") ;
            e.printStackTrace();
        }
    }
}

```

```

import java.io.*;

public class Cuatro
{
    private static int metodo() {
        int valor = 0;
        try {
            valor = valor+1;
            valor = valor + Integer.parseInt("W");
            valor = valor + 1;
            System.out.println("Valor al final del try : " + valor) ;
            throw new IOException();
        } catch (IOException e) {
            valor = valor + Integer.parseInt("42");
            System.out.println("Valor al final del catch : " + valor);
        } finally {
            valor = valor + 1;
            System.out.println("Valor al final de finally: " + valor);
        }
        valor = valor + 1;
        System.out.println ("Valor antes del return: " + valor) ;
        return valor ;
    }

    public static void main(String[] args) {
        try {
            System.out.println(metodo());
        } catch (Exception e) {
            System.err.println("Excepcion en metodo()");
            e.printStackTrace();
        }
    }
}

```

66. **Ejercicio 66.** Indica qué se mostrará por pantalla cuando se ejecute esta clase:

1. Si se ejecuta con java Cinco casa
2. Si se ejecuta con java Cinco 0
3. Si se ejecuta con java Cinco 7

```

public class Cinco {
    public static void main(String args[]) {
        try {
            int a = Integer.parseInt(args[0]);
            System.out.println("a = " + a);
            int b=42/a;
            String c = "hola";
            char d = c.charAt(50);
        } catch (ArithmeticException e) {
            System.out.println("div por 0: " + e);
        } catch (IndexOutOfBoundsException e) {
            System.out.println("Índice del String fuera de límites: " + e);
        } finally {

```

```

    }
}

```

67. **Ejercicio 67.** Indica cuál será la salida del siguiente programa y por qué

```

public class Seis {
    public static void procA() {
        try {
            System.out.println("dentro del procA"); 2
            throw new RuntimeException("demo"); 3
        } finally {
            System.out.println("Finally del procA"); 4
        }
    }

    public static void procB() {
        try {
            System.out.println("dentro del procB"); 6
            return; 7
        } finally {
            System.out.println("finally del procB"); 8
        }
    }

    public static void main(String args[]) {
        try {
            procA(); 1
        } catch (Exception e) {
            procB(); 5
        }
    }
}

```

68. **Ejercicio 68.** Indica cuál será la salida del siguiente programa y por qué

```

public class Siete {
    public static void metodo() {
        try {
            throw new NullPointerException("demo"); 2
        } catch (NullPointerException e) {
            System.out.println("capturada en método"); 3
            throw e; 4
        }
    }

    public static void main (String args[]) {
        try {
            metodo(); 1
        } catch (NullPointerException e) {
            System.out.println("capturada en main " + e); 5
        }
    }
}

```

3. Actividades

1. **Actividad 1.** Transforma el siguiente bucle for en un bucle while:

```
for (i=5; i<15; i++) {
    System.out.println(i);
}
```

2. **Actividad 2.** Programa que muestre por pantalla los 5 primeros números pares.
 3. **Actividad 3.** Programa que muestre por pantalla del número 200 al 300.
 4. **Actividad 4.** Programa que muestre en pantalla la tabla de multiplicar del 1 al 10 con el formato:

```
...
Tabla del 2
*****
2 x 1 = 2
2 x 2 = 4
...
2 x 10 = 20
...
Tabla del 3
*****
...
```

5. **Actividad 5.** Programa que muestre los números del 1 al 100 sin mostrar los múltiplos de 5.
 6. **Actividad 6.** Leer un número y mostrar su cuadrado, repetir el proceso hasta que se introduzca un número negativo.
 7. **Actividad 7.** Leer un número e indicar si es positivo o negativo. El proceso se repetirá hasta que se introduzca un 0.
 8. **Actividad 8.** Leer números hasta que se introduzca un 0. Para cada uno indicar si es par o impar.
 9. **Actividad 9.** Pedir números hasta que se teclee uno negativo, y mostrar cuántos números se han introducido.
 10. **Actividad 10.** Realizar un juego para adivinar un número **X**. Para ello pedir un número **N**, y luego ir pidiendo números indicando "mayor" o "menor" según sea mayor o menor con respecto a **X**. El proceso termina cuando el usuario acierta.
 11. **Actividad 11.** Pedir números hasta que se teclee un 0, mostrar la suma de todos los números introducidos.
 12. **Actividad 12.** Pedir números hasta que se introduzca uno negativo, y calcular la media.
 13. **Actividad 13.** Pedir un número **N**, y mostrar todos los números del 1 al **N**.
 14. **Actividad 14.** Escribir todos los números del 100 al 0 de 7 en 7.
 15. **Actividad 15.** Pedir 15 números y escribir la suma total.
 16. **Actividad 16.** Diseñar un programa que muestre el producto de los 10 primeros números impares.
 17. **Actividad 17.** Pedir un número y calcular su factorial (el factorial se representa con el símbolo **!**).

```

1! = 1
2! = 2 * 1 = 2
3! = 3 * 2 * 1 = 6
4! = 4 * 3 * 2 * 1 = 24
5! = 5 * 4 * 3 * 2 * 1 = 120

```

18. **Actividad 18.** Pedir 10 números. Mostrar la media de los números positivos, la media de los números negativos y la cantidad de ceros.
19. **Actividad 19.** Pedir 10 sueldos. Mostrar su suma y cuantos hay mayores de 1000€.
20. **Actividad 20.** Dadas las edades y alturas de 5 alumnos, mostrar la edad y la estatura media, la cantidad de alumnos mayores de 18 años, y la cantidad de alumnos que miden más de 1.75.
21. **Actividad 21.** Pide un número (que debe estar entre 0 y 10) y mostrar la tabla de multiplicar de dicho número.
22. **Actividad 22.** Dadas 6 notas, escribir la cantidad de alumnos aprobados y suspensos.
23. **Actividad 23.** Pedir un número `N`, introducir `N` sueldos, y mostrar el sueldo máximo.
24. **Actividad 24.** Pedir 10 números, y mostrar al final si se ha introducido alguno negativo.
25. **Actividad 25.** Pedir 5 calificaciones de alumnos y decir al final si hay algún suspenso.
26. **Actividad 26.** Pedir 5 números e indicar si alguno es múltiplo de 3.
27. **Actividad 27.** Realiza un programa que pida una hora por teclado y que muestre luego 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.
28. **Actividad 28.** Escribe un programa en que dado un número del 1 a 7 escriba el correspondiente nombre del día de la semana.
29. **Actividad 29.** Escribe un programa que calcule el salario semanal de un trabajador teniendo en cuenta que las horas ordinarias (40 primeras horas de trabajo) se pagan a 12 euros la hora. A partir de la hora 41, se pagan a 16 euros la hora.
30. **Actividad 30.** Realiza un programa que calcule la media de tres notas.
31. **Actividad 31.** Amplía el programa anterior para que diga la nota del boletín (insuficiente, suficiente, bien, notable o sobresaliente).
32. **Actividad 32.** Escribe un programa que nos diga el horóscopo a partir del día y el mes de nacimiento.
33. **Actividad 33.** Realiza un minicuestionario con 4 preguntas tipo test sobre las asignaturas que se imparten en el curso. Cada pregunta acertada sumará un punto. El programa mostrará al final la calificación obtenida.
34. **Actividad 34.** Calcula la nota de un trimestre de la asignatura Programación. El programa pedirá las dos notas que ha sacado el alumno en los dos primeros controles. Si la media de los dos controles da un número mayor o igual a 5, el alumno está aprobado y se mostrará la media. En caso de que la media sea un número menor que 5, el alumno habrá tenido que hacer el examen de recuperación que se califica como apto o no apto, por tanto se debe preguntar al usuario ¿Cuál ha sido el resultado de la recuperación? (apto/no apto). Si el resultado de la recuperación es apto, la nota será un 5; en caso contrario, la nota será 1.

Ejemplo 1:

```

Nota del primer control: 7 Nota del segundo control: 10
Tu nota de Programación es 8.5

```

Nota del primer control: 6 Nota del segundo control: 3
 ¿Cuál ha sido el resultado de la recuperación? (apto/no apto): apto
 Tu nota de Programación es 5

Ejemplo 3:

Nota del primer control: 6 Nota del segundo control: 3
 ¿Cuál ha sido el resultado de la recuperación? (apto/no apto): no apto
 Tu nota de Programación es 1

35. **Actividad 35**. Muestra los números múltiplos de 5 entre el 0 y el 100 utilizando un bucle ``for``.

35. **Actividad 36**. Muestra los números múltiplos de 5 entre el 0 y el 100 utilizando un bucle ``while``.

35. **Actividad 37**. Muestra los números múltiplos de 5 entre el 0 y el 100 utilizando un bucle ``do while``.

35. **Actividad 38**. Muestra los números del 320 al 160, contando de 20 en 20 hacia atrás utilizando un bucle ``for``.

35. **Actividad 39**. Muestra los números del 320 al 160, contando de 20 en 20 hacia atrás utilizando un bucle ``while``.

35. **Actividad 40**. Muestra los números del 320 al 160, contando de 20 en 20 utilizando un bucle ``do-while``.

35. **Actividad 41**. Realiza el control de acceso a una caja fuerte. La combinación será un número de 4 cifras. El programa nos pedirá la combinación para abrirla. Si no acertamos, se nos mostrará el mensaje " Lo siento, esa no es la combinación" y si acertamos se nos dirá "La caja fuerte se ha abierto satisfactoriamente" . Tendremos cuatro oportunidades para abrir la caja fuerte.

35. **Actividad 42**. Escribe un programa que muestre en tres columnas, el cuadrado y el cubo de los 5 primeros números enteros a partir de uno que se introduce por teclado.

35. **Actividad 43**. Escribe un programa que pida una base y un exponente (entero positivo) y que calcule la potencia. (sin usar ``Math``)

35. **Actividad 44**. Realiza un programa que sume los 100 números siguientes a un número entero y positivo introducido por teclado. Se debe comprobar que el dato introducido es correcto (que es un número positivo).

35. **Actividad 45**. Escribe 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.

35. **Actividad 46**. Realiza un programa que vaya pidiendo números hasta que se introduzca un numero 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.

35. **Actividad 47**. 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.

35. **Actividad 48**. Escribe un programa que muestre, cuente y sume los múltiplos de 3 que hay entre 1 y un número leído por teclado.

35. ****Actividad 49****. Escribe un programa que calcule el precio final de un producto según su base imponible (precio antes de impuestos), el tipo de IVA aplicado (general, reducido o superreducido) y el código promocional. Los tipos de IVA general, reducido y superreducido son del 21%, 10% y 4% respectivamente. Los códigos promocionales pueden ser nopro, mitad, meno5 o 5porc que significan respectivamente que no se aplica promoción, el precio se reduce a la mitad, se descuentan 5 euros o se descuenta el 5%.

Ejemplo:

```
Introduzca la base imponible: 25
Introduzca el tipo de IVA (general, reducido o superreducido): reducido
Introduzca el código promocional (nopro, mitad, meno5 o 5porc): mitad
Base imponible 25.00
Cód. promo. (mitad): -12.50
IVA (10%) 1.25
Precio con IVA 13.75
TOTAL 13.75
```

50. **Actividad 50**. Pedir un año e indicar si es bisiesto, teniendo en cuenta que son bisiestos todos los años divisibles por 4, excluyendo los que sean divisibles por 100, pero no los que sean divisibles por 400.

En pseudocódigo se calcularía así:

```
SI (año divisible por 4) Y (año no divisible por 100) O (año divisible por 400))
    ENTONCES
        es bisiesto
SINO
    no es bisiesto
FIN_SI
```

51. **Actividad 51**. Pedir un número de 20 a 99 y mostrarlo escrito. Por ejemplo, para 56 mostrar: cincuenta y seis.

52. **Actividad 52**. Introducir datos de un vehículo (marca, modelo y precio). Devolver el precio con IVA del vehículo. Controlar con Excepciones que el precio del vehículo introducido son números y que el cálculo de Precio Final con IVA no devuelva error.

53. **Actividad 53**. Introducir códigos de alumnos, nombre y nota hasta que se introduzca un código de alumno negativo. Devolver la nota media de los alumnos la clase. Controlar con Excepciones que las notas introducidas son números y que si no se introducen alumnos el cálculo de la media no devuelva error.

54. **Actividad 54**. Crear una función o método llamado `impFinal`, que calcule el importe final de una compra. Los parámetros que se le pasarán a la función son el `precio` del producto, las `cantidad de unidades` compradas, el `porcentaje de iva` y el `porcentaje de descuento`. El método principal debe pedir por teclado el precio del producto, las unidades adquiridas, el porcentaje de IVA y el porcentaje de descuento y devolver el `Importe final` de la Factura.

55. **Actividad 55**. Crear una función que calcule la capacidad de un disco. La capacidad se calcula multiplicando los Cabezales o pistas del disco por los Cilindros por los Sectores por Tamaño de Sector. El método principal debe pedir por teclado los Cabezales o Pistas del

Por ejemplo: Calcular la capacidad de un disco teniendo en cuenta que dispone de 10 Cabezales o Pistas, 65535 Cilindros, 1024 Sectores/pista y un Tamaño de 512 bytes/sector:

Capacidad del disco = $10 * 65535 * 1024 * 512 = 343597383680$ bytes

$343597383680 \text{ bytes} / 1024 / 1024 / 1024 = 320 \text{ Gbytes}$

56. **Actividad 56.** Función que devuelva el mayor de tres números. El método principal debe pedir por teclado los tres números introducidos por el teclado. La función debe recibir como parámetros los tres números y devolver el mayor.

4. Fuentes de información

- [Wikipedia](#)
- [Programación \(Grado Superior\) - Juan Carlos Moreno Pérez \(Ed. Ra-ma\)](#)
- Apuntes IES Henri Matisse (Javi García Jimenez?)
- Apuntes AulaCampus
- [Apuntes José Luis Comesaña](#)
- [Apuntes IOC Programació bàsica \(Joan Arnedo Moreno\)](#)
- [Apuntes IOC Programació Orientada a Objectes \(Joan Arnedo Moreno\)](#)