

Ejercicios Resueltos. Secuencial, Selectivas y Repetitivas

"La sabiduría es un adorno en la prosperidad y un refugio en la adversidad".

Aristóteles

Ejemplo 1: Diseñar un programa en el a partir de dos enteros no negativos, calculemos su suma y producto.

Entrada: *¿Qué necesito?* Dos enteros no negativos.

Salida: *¿Qué me proporciona?* La suma y el producto.

```
package org.pc.tema01;

public class Ejercicio01 {

    /**
     * Suma y multiplica dos enteros no negativos
     *
     * @param args
     */
    public static void main(String[] args) {

        int dato1, dato2, suma, producto;
        // PRE: dato1 >= 0 y dato2 >= 0
        dato1 = 7;
        dato2 = 10;
        suma = dato1 + dato2;
        producto = dato1 * dato2;
        System.out.println("La suma es..." + suma);
        System.out.println("El producto es..." + producto);
        ;
    }

}

//POST: producto = dato1 * dato2 y suma = dato1 + dato2
```

Ejemplo 2. Diseñar un programa que permita calcular la suma de dos enteros si son positivos y el producto en caso contrario.

Entrada: Dos números enteros.

Salida: Suma o producto.

```
package org.pc.tema01;

public class Ejercicio02 {

    /**
     * Permite calcular la suma de dos enteros si son positivos y el
     * producto en caso contrario
     *
     * @param args
     */
    public static void main(String[] args) {
```

```

    int dato1, dato2, suma, producto;
    dato1 = 10;
    dato2 = -5;
    if (dato1 > 0 && dato2 > 0) {
        suma = dato1 + dato2;
        System.out.println("la suma es...." + suma);
    } else {
        producto = dato1 * dato2;
        System.out.println("El producto es...." + producto);
    }
}
}

```

Ejemplo 3. En un centro comercial se han detectado errores en aquellos artículos cuyos códigos están comprendidos entre:

14681 hasta 15681 ambos inclusive

70001 hasta 79999 ambos inclusive

99999 hasta 110110 ambos inclusive

Diseñar un programa que con el código de un artículo muestre por pantalla un mensaje que indique si el código es o no defectuoso.

Entrada: un nº de código

Salida: mensaje “Defectuoso” o “No defectuoso”

```
package org.pc.tema01;
```

```

public class Ejercicio03 {

    /**
     * A partir del código de un artículo, muestre por pantalla un
     * mensaje que indica si el código es o no defectuoso.
     * Códigos defectuosos:
     * 14681 hasta 15681 ambos inclusive
     * 70001 hasta 79999 ambos inclusive
     * 99999 hasta 110110 ambos inclusive
     * @param args
     */
    public static void main(String[] args) {
        int codigo;
        codigo = 50000; //Ejecutar cambiando el valor a 3000000000

        if ((codigo >= 14681) && (codigo <= 15681) || (codigo >=
            70001)&& (codigo <= 79999) || (codigo >= 99999) && (codigo
            <= 110110)) {
            System.out.println("CÓDIGO DEFECTUOSO");
        } else {
            System.out.println("CÓDIGO CORRECTO");
        }
    }
}

//Cambiar el código utilizando variables booleanas para acortar la
condición

```

Ejemplo 4. Una empresa desea contratar a aquellos trabajadores cuya edad sea de al menos 18 años y tengan más de 85 puntos en alguna de las dos pruebas que han realizado. Se debe de realizar un programa que muestre por pantalla un mensaje de “Contratado” o “No contratado” si la edad y las dos calificaciones cumplen las condiciones que se han indicado.

Entrada: Edad y dos calificaciones

Salida: Mensaje “Contratado” o “No contratado”

```
package org.pc.tema01;

public class Ejercicio04 {

    /**
     * Contratación en una empresa si los trabajadores cumplen:
     * Que la edad sea de al menos 18 años y
     * que tengan más de 85 puntos en alguna de las dos pruebas
     * que han realizado
     * @param args
     */
    public static void main(String[] args) {
        int edad, nota1, nota2;
        // PRE: edad >= 0 y nota1 >= 0 y nota2 >= 0
        edad = 18;
        nota1 = 15;
        nota2 = 90;
        if ((edad >= 18) && ((nota1 > 85) || (nota2 > 85))) {
            System.out.println("CONTRATADO");
        } else {
            System.out.println("NO CONTRATADO");
        }
    }
}
```

Ejemplo 5. Diseñar un programa que determine si un año es bisiesto. Deberá mostrar un mensaje “Año Bisiesto” o “Año No Bisiesto”.

Análisis del problema.

Veamos algunos ejemplos:

$1600 \Rightarrow 1600 / 4 = 400 \Rightarrow$ Bisiesto

$1800 \Rightarrow 1800 / 4 = 450 \Rightarrow$ No bisiesto

$1992 \Rightarrow 1992 / 4 = 498 \Rightarrow$ Bisiesto

$1993 \Rightarrow 1993 / 4 = 498.25 \Rightarrow$ No bisiesto

Entrada: año

Salida: mensaje “Bisiesto” o “No bisiesto”

```
package org.pc.tema01;

public class Ejercicio05 {

    /**
     * Muestra un mensaje en pantalla indicando si un año es
     * bisiesto.
     * @param args
     */
    public static void main(String[] args) {

        int año = 1800;
        // PRE: año > 0
        if ((año % 4 == 0) && (año % 100 != 0) || (año % 400 == 0)) {
            System.out.println("AÑO BISIESTO");
        } else {
            System.out.println("AÑO NO BISIESTO");
        }
    }
}
```

Ejemplo 6. Diseña un programa que a partir de un n° comprendido entre 1 y 7 muestre por pantalla el día de la semana con el que se corresponde.

Entrada: entero comprendido entre 1 y 7

Salida: mensaje “LUNES” o “MARTES” o.....”DOMINGO”

```
package org.pc.tema01;

public class Ejercicio06 {

    /**
     * A partir de un n° comprendido entre 1 y 7 muestra por
     * pantalla el día de la semana con el que se corresponde
     *
     * @param args
     */
    public static void main(String[] args) {

        int dia = 9;

        switch (dia) {
            case 1:
                System.out.println("LUNES");

                break;
            case 2:
                System.out.println("MARTES");

                break;
            case 3:
                System.out.println("MIÉRCOLES");

                break;
            case 4:
                System.out.println("JUEVES");

                break;
        }
    }
}
```

```

        case 5:
            System.out.println("VIERNES");

            break;
        case 6:
            System.out.println("SÁBADO");

            break;
        case 7:
            System.out.println("DOMINGO");

            break;

        default:
            System.out.println(" NO CORRESPONDE A NINGÚN DÍA DE
LA SEMANA");
            break;
    }
}
}

```

Ejemplo 7. Dados dos enteros positivos, diseñar un programa que calcule el cociente y el resto haciendo uso exclusivamente de los operadores + y −.

$$\begin{array}{rcl}
 39 \overline{)11} & \Rightarrow & \begin{array}{l} 39 - 11 = 28 \text{ Cociente} = 1 \\ 28 - 11 = 17 \text{ Cociente} = 2 \\ 17 - 11 = 6 \text{ Cociente} = 3 \end{array}
 \end{array}
 \left. \vphantom{\begin{array}{rcl} 39 \overline{)11} & \Rightarrow & \begin{array}{l} 39 - 11 = 28 \text{ Cociente} = 1 \\ 28 - 11 = 17 \text{ Cociente} = 2 \\ 17 - 11 = 6 \text{ Cociente} = 3 \end{array}} \right\} \Rightarrow \begin{cases} \text{Resto} - \text{Divisor} = \text{Resto} \\ y \\ \text{Cociente} = \text{Cociente} + 1 \end{cases}$$

Entrada: Dividendo y divisor, enteros positivos

Salida: Cociente y resto.

Nos debemos de plantear las 5 preguntas: ¿hay proceso que se repite? Etc.

```

package org.pc.tema01;

public class Ejercicio07 {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int dividendo = 39;
        int divisor = 11;
        int cociente, resto;
        cociente = 0;
        resto = dividendo;
        while (resto >= divisor) {
            resto = resto - divisor;
            cociente = cociente + 1;
        }
        System.out.println("El cociente es " + cociente);
        System.out.println("El resto es " + resto);
    }

}

```

Seguimiento Manual

Iteración	Dividendo	Divisor	Resto	Cociente
0	39	11	39	0
1	39	11	28	1
2	39	11	17	2
3	39	11	6	3

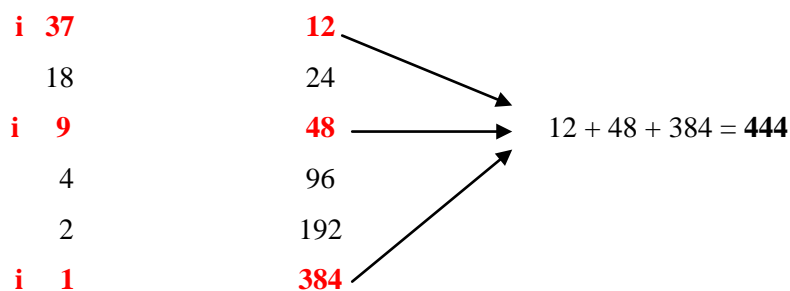
No habría sido necesario poner las columnas del dividendo y del divisor puesto que no cambian en las distintas iteraciones.

Es buena costumbre poner las variables en la tabla en el orden en el que aparecen en el cuerpo del bucle, esto permite hacer un seguimiento más fácil.

Ejemplo 8. Diseñar un programa que permita realizar el producto de dos enteros positivos utilizando la “*Multiplicación Rusa*”. Para ello se parte de los dos enteros positivos y se multiplica por 2 el multiplicando y se divide entre 2 el multiplicador. Continuamos realizando esas operaciones hasta que el multiplicador vale 1. Se suman todos los multiplicandos correspondientes a multiplicadores impares y este resultado es el producto.

Ejemplo: $37 \times 12 = 444$

Multiplicador Multiplicando



Entrada: Multiplicador y multiplicando, enteros positivos

Salida: Producto

```
package org.pc.tema01;

public class Ejercicio08 {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
    }
}
```

```

    int multiplicador = 37;
    int multiplicando = 12;
    int producto = 0;
    System.out.println("MULTIPLICADOR      MULTIPLICANDO");

    while (multiplicador >= 1) {
        System.out.printf("%7d%18d\n", multiplicador, multiplicando);
        if (multiplicador % 2 != 0) // multiplicador es impar
            producto = producto + multiplicando;
        multiplicador = multiplicador / 2;
        multiplicando = multiplicando * 2;
    }
    System.out.println("El producto es " + producto);
}
}

```

Ejemplo 9. Diseñar un programa que permita el cálculo del factorial de un valor entero positivo.

n entero positivo $\Rightarrow n! = n \cdot (n-1) \cdot (n-2) \dots 1$

Entrada: entero positivo

Salida: factorial

Veamos varias soluciones para diseñar el bucle:

1.-

```

package org.pc.tema01;

public class Ejercicio09V1 {

    /**
     * Calcula el factorial de un entero positivo
     *  $n! = n \cdot (n-1) \cdot (n-2) \dots 1$ 
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int dato = 9;
        long fact = 1;
        int i = 1;
        while (i <= dato){
            fact = fact * i;
            i++;
        }
        System.out.println("El factorial de "+dato + " es "+fact);
    }
}

```

2.- Usando dos índices para hacer el producto, uno por la derecha y otro por la izquierda.

Ejemplo:

$$6! = 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$$

$$7! = 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$$

```
package org.pc.tema01;

public class Ejercicio09V2 {

    /**
     * Calcula el factorial de un entero positivo haciendo productos por
     * la derecha y por la izquierda n! = n.(n-1).(n-2)...1
     */
    @param args
    /**
     public static void main(String[] args) {
         // TODO Auto-generated method stub
         int dato = 9;
         long fact = 1;
         int i = 1;
         int j = dato;
         while (i < j) {
             fact = fact * i * j;
             i++;
             j--;
         }
         if (i == j) {
             fact = fact * i;
         }
         System.out.println("El factorial de " + dato + " es " +
             fact);
     }
}
```

Ejemplo 10. Diseñar un programa que muestre las potencias de 2 comprendidas entre 1 y 100000.

```
package org.pc.tema01;

public class Ejercicio10 {

    /**
     * Muestra las potencias de 2 comprendidas entre 1 y 100000
     */
    @param args
    /**
```



```

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int potencia = 1;
        int i = 0;
        int limite = 100000;
        while (potencia <= limite){
            System.out.printf("%d\t%d\n",i,potencia);
            potencia = potencia * 2;
            i++;
        }
    }
}

```

Ejemplo 11. Diseñar un programa que muestre las tablas de multiplicar del 1 al 10.

```

package org.pc.tema01;

public class Ejercicio11 {

    /**
     * Muestra las tablas de multiplicar del 1 al 10
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        for (int i = 1; i <= 10; i++) {
            System.out.println("    TABLA DEL "+i);
            System.out.println();
            for (int j = 1; j <= 10; j++) {
                System.out.println("        " + i + "X" +j +"=" + i*j);
            }
            System.out.println();
        }
    }
}

```

Ejemplo 12. Diseñar un programa que muestre si un entero positivo (>1) es primo

Definición de n° primo: aquel que es divisible entre el mismo y la unidad.

Entrada: Un n° entero positivo > 1.

Salida: Mensaje n° primo/n° no primo.

Daremos una primera solución aplicando la definición y posteriormente haremos optimizaciones.

```

package org.pc.tema01;

public class Ejercicio12V0 {

    /**
     * Muestra si un entero positivo es primo
     * @param args
     */
    public static void main(String[] args) {

```

```

        // TODO Auto-generated method stub
        int dato = 7;
        int divisor = 2;
        boolean primo = true;
        for (divisor = 2; divisor < dato; divisor++){
            if (dato % divisor == 0)
                primo = false;
        }
        if (primo)
            System.out.println("El número es primo");
        else
            System.out.println("El número no es primo");
    }
}

```

Primera optimización: si encontramos un divisor no es necesario seguir probando ya que podemos asegurar que el número ya no es primo. Esto hará que transformemos la estructura **for** en un **while** que permita escapar del bucle en la circunstancia expuesta.

```

package org.pc.tema01;

public class Ejercicio12V1 {

    /**
     * Muestra si un entero positivo es primo
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int dato = 7;
        int divisor = 2;
        boolean primo = true;
        while (primo && divisor < dato){
            if (dato % divisor == 0)
                primo = false;
            divisor++;
        }
        if (primo)
            System.out.println("El número es primo");
        else
            System.out.println("El número no es primo");
    }
}

```

Segunda optimización:

- ✓ Comprobamos si es divisible entre 2. De no serlo no comprobaremos divisores pares.
- ✓ Llegaremos probando divisores hasta \sqrt{nDato} . Si hasta ese valor no hemos encontrado ningún divisor, no es necesario seguir comprobando.