

Ejercicio 1

Un grupo de inteligencia militar desea codificar los mensajes secretos de tal forma que no puedan ser interpretados con una lectura directa, para lo cual han establecido las siguientes reglas:

- a) Todo mensaje debe tener sus letras en mayúsculas.
- b) Reemplazar cada letra por la que sigue según abecedario, excepto Z que se deberá reemplazar con la letra A.
- c) Reemplazar cada dígito encontrado por el siguiente número excepto el 9 que deberá ser reemplazado por el 0.
- d) Los espacios en blanco se reemplazan por la letra Ç.
- e) No se reemplazarán el resto de caracteres.

Debe tener al menos las funciones:

```
string leerTexto();  
string textoMayuscula(String cadena)  
string reemplazarLetraSiguiente(String cadena )  
string reemplazarNumeroSiguiente(String cadena)  
string reemplazarEspaciosBlanco(String cadena)  
string reemplazarLetraAnterior(String cadena )  
string reemplazarNumeroAnterior(String cadena)  
string ponerEspacios en blaco(String cadena)
```

Crear el algoritmo que decodifique el mensaje

Crear el algoritmo que codifique el mensaje

Crea un menú que pida codificar o decodificar el mensaje.

Ejercicio 2

Una agencia de venta de autos paga a su personal de ventas un salario de 800,00€ más una comisión de 170,00€ por auto vendido más un 5% del valor de venta.

- Diseñar un algoritmo para calcular el salario de un vendedor en un determinado mes,
- conociendo el nº de automóviles vendidos y el total del monto de ventas.
 - `public static double salario(double montoVentas, int autosVendidos){ }`

Ejercicio 3

Programa que lea por teclado tres números enteros H, M, S correspondientes a hora, minutos y segundos respectivamente, y comprueba si la hora que indican es una hora válida.

- Supondremos que leemos una hora en modo 24 Horas, es decir, el valor válido para las horas será mayor o igual que cero y menor que 24.
- El valor válido para los minutos y segundos estará comprendido entre 0 y 59 ambos incluidos
- Debe tener las funciones
 - `public static boolean esHora(short valor){}`
 - `public static boolean esMinutooSegundo(short valor){}`

Ejercicio 4

Programa que lea una variable entera mes y compruebe si el valor corresponde a un mes de 30 días, de 31 o de 28. Supondremos que febrero tiene 28 días. Se mostrará además el nombre del mes. Se debe comprobar que el valor introducido esté comprendido entre 1 y 12.

- Debe tener las funciones
 - `public static String nombreMes(byte numero){}`
 - `public static byte diasMes(byte numero){}`

Ejercicio 5

Diseña una función que muestre en binario un número entre 0 y 255

Ejercicio 6 - Extra

Número narcisista en Java

Programa Java que compruebe si un número es narcisista.

Se considera que un número es narcisista si la suma de sus cifras elevadas al número de cifras que tiene el número es igual al propio número.

Por ejemplo, el número 153 es un número narcisista. El número tiene 3 cifras y si sumamos cada una de sus cifras elevada al cubo el resultado es el propio número:

$$1^3 + 5^3 + 3^3 \rightarrow 1 + 125 + 27 \rightarrow 153$$

Otro ejemplo de número narcisista es el 1634

$$1^4 + 6^4 + 3^4 + 4^4 \rightarrow 1 + 1296 + 81 + 256 \rightarrow 1634$$

El programa pedirá que se introduzca por teclado un número entero positivo y determinará si es un número narcisista o no lo es.

Solución:

El primer paso para determinar si un número es narcisista es calcular el número de cifras que tiene. Los pasos a seguir para calcular el número de cifras de un número se explican en el ejercicio calcular las cifras de un número y aquí lo haremos de la misma forma.

Una vez sepamos la cantidad de cifras que tiene el número procederemos a obtener cada una de las cifras del número, la elevaremos al número de cifras que tiene y sumaremos el resultado. Para obtener cada una de las cifras del número lo haremos de la misma forma que se explica en el ejercicio sumar las cifras de un número.