

Cadena de caracteres y Colecciones.

Relación Nº: 6

1. Escribe una función, de nombre `alReves()`, que recibe una cadena de texto y devuelva esta cadena al revés
2. Escribe una función, de nombre `trimD`, que se le pase una cadena de texto y le quite los espacios que haya al final.
3. Escribe una función, de nombre `trimI`, que se le pase una cadena de texto y le quite los espacios que haya al principio.
4. Usando las funciones anteriores, escribe una función, de nombre `TrimAll()`, que elimine todos los espacios por delante y por detrás de una cadena de texto.
5. Escribe una función, de nombre `contarPalabras`, que reciba una cadena de texto y un carácter (separador de palabras) y devuelva un entero con el número de palabras de la cadena.
6. Escribe una función, de nombre `letraNIF`, que dado el DNI calcule y devuelva la letra del NIF. La fórmula que permite el cálculo es: $\text{DNI} \bmod 23$, esto nos dará la posición de la letra de la Cadena de letras "TRWAGMYFDPXBNJZSQVHLCKE".
7. Haz una función para Validar ISBN (International Standard Book Number), de nombre `validarISBN`, que devuelva un valor booleano indicando si dicho código es correcto o no.

El ISBN (International Standard Book Numbers) sirve para identificar libros y otras publicaciones a nivel mundial. Si abres un libro cualquiera, verás que en sus primeras páginas consta este ISBN.

El ISBN consta de una serie de dígitos decimales (de 0 a 9), de los cuales, el último es un carácter de control, que puede ser bien un dígito decimal (de 0 a 9) o la letra "X". A veces el ISBN lleva guiones. No se deben tener en cuenta para validarlo.

Para comprobar un ISBN:

1. Si el ISBN lleva guiones en su interior, eliminarlos.
2. Multiplicar cada dígito -excepto el último, que puede ser un dígito o una X- por la posición que ocupa e ir sumándolos (el 1er dígito por 1, sumado al 2º dígito por 2, sumado al 3er dígito por 3...)
3. Obtener el módulo de la división del resultado obtenido en el paso 2 entre 11. Dicho módulo debe coincidir con el último dígito del ISBN. Si el módulo fuera 10, el último dígito es una X.

Ejemplo: ISBN = 847-210-262-9

Para comprobarlo, obtenemos

$$S = 8*1+4*2+7*3+2*4+1*5+0*6+2*7+6*8+2*9 = 130$$

$130 \bmod 11 = 9$, Si coincide con el último dígito del ISBN ES CORRECTO.

8. Haz una función para Validar códigos EAN, de nombre `validarEAN`, que devuelva un tipo booleano y me permita comprobar si dado un código EAN es correcto.



El código EAN (European Article Number) es un sistema de código de barras para asignar un número único a cada producto. Los códigos más comunes tienen 8 o 13 dígitos, especialmente 13 (sistemas conocidos como EAN8 y EAN13). En ellos van codificados el país de origen del producto, la empresa y el propio producto. El último de los dígitos es un dígito de control para evitar errores de transcripción.

El algoritmo para comprobar que un código EAN8 o EAN13 ha sido transcrito correctamente es extremadamente sencillo.

Podemos describirlo algorítmicamente de esta manera:

- Comprobar que el código tiene 8 o 13 dígitos. De no ser así, no es correcto.
- Sumar los dígitos de lugares pares por un lado y los de los impares por otro, pero sin incluir el último dígito.
- Si el código es EAN13, multiplicar la suma de los pares por 3.
- Si el código es EAN8, es la suma de los impares la que se multiplica por 3.
- Sumar el resultado de los pares y el de los impares y hallar el resto de la división por 10.
- Realizar la operación 10 menos ese resto y ese es el dígito de control.
- Si como resultado sale 10, entenderemos que el dígito de control es 0.
- Comprobar que el dígito de control que hemos calculado y el último dígito del código EAN coinciden

Por ejemplo, para validar el código EAN8 "12345678" (Obviamente es inventado)

1. Separar el dígito de control. Nos quedamos con "1234567" y "8"
2. Sumar pares: $\text{sumapares} = 2+4+6=12$
3. Sumar impares: $\text{sumaimpares} = 1+3+5+7=16$
4. Como es EAN8, multiplicamos los impares por 3.
5. $\text{sumaimpares} = 16*3=48$
6. Sumar el resultado de pares e impares: $12+48=60$



7. Hallar el resto de la división por 10: $60 \bmod 10 = 0$
8. Hacer 10-resto: $10 - 0 = 10$
9. Como nos ha salido 10, el dígito de control es 0.
10. Comparar el dígito de control que hemos calculado con el que tenía el código: Nos sale 0 y el código tenía un 8. Es incorrecto.

9. Haz una función “ValidadCC” para validar la cuenta corriente bancaria. El Código de Cuenta Corriente es un número que está formado por 20 dígitos que a su vez está dividido en 4 grupos.

Entidad(4) – Sucursal(4) – Control(2) – N° Cuenta corriente (10)

- **Código de la entidad bancaria:** Está formado por los 4 primeros dígitos. Cada entidad financiera tiene su propio código.
- **Código de sucursal:** Está formado por los siguientes 4 dígitos. Cada sucursal de una entidad tiene su propio código.
- **Código de control:** Está formado por 2 dígitos. El primer dígito de control es de la entidad y sucursal, mientras que el segundo dígito es de control de la cuenta corriente. Estos dígitos no son al azar si no que se utiliza un algoritmo para calcularlo.
- **Código de cuenta corriente:** Está formada por los 10 últimos dígitos del CC

Para hacer este ejercicio mira el algoritmo en la Wikipedia o en Internet.

10. Haz una función para validar el código IBAN (International Bank Account Number) de una cuenta bancaria.

11. Haz un programa que pida un número por pantalla entre 0 y 999.999 y lo escriba en letra.

Por ejemplo:

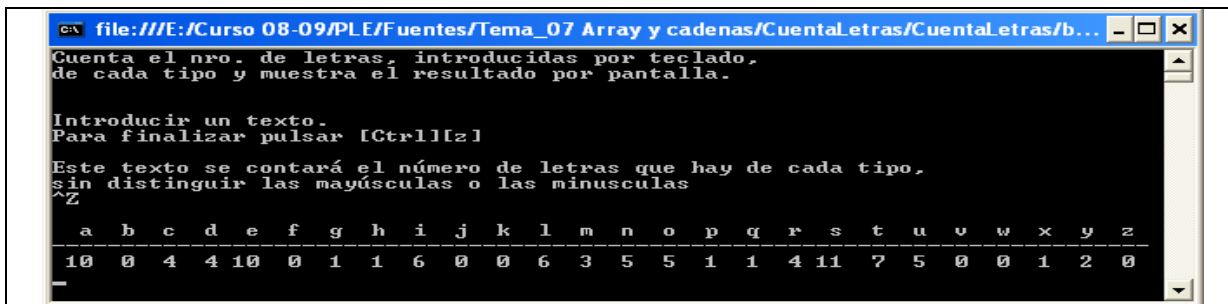
234674 -> DOSCIENTOS TREINTA Y CUATRO MIL SEISCIENTOS SETENTA Y CUATRO

Indicación:

- Construye una función Unidades (N) que imprima números entre 1 y 15.
- Construye una función Decenas (N) que imprima números entre 1 y 99.
- Construye una función Centenas (N) que imprima números entre 1 y 999.
- Construye una función Miles (N) que imprima números entre 1 y 999.999.

Cada una de estas funciones hace uso de las anteriormente construidas para realizar su cometido.

12. Haz una función, de nombre AMayuscula, que devuelva una cadena de texto siempre en mayúsculas.
13. Escribe un programa que pida desde teclado la introducción de una frase y muestre por pantalla la lista de todas las palabras y comenzando por mayúscula cada palabra. El separador de palabras puede ser algunos de estos caracteres: espacio, guión(-) o el símbolo del O-lógico (|) que es el Unicode 124.
14. Escribir un programa que cuente el número de veces que aparece cada una de las letras de un texto introducido por el teclado y a continuación imprima el resultado. Para hacer el ejemplo sencillo, vamos a suponer que el texto sólo contiene letras minúsculas del alfabeto inglés (no hay ni letras acentuadas, ni la //, ni la ñ). La solución podría ser de la forma siguiente:



| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
|----|---|---|---|----|---|---|---|---|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|---|---|
| 10 | 0 | 4 | 4 | 10 | 0 | 1 | 1 | 6 | 0 | 0 | 6 | 3 | 5 | 5 | 1 | 1 | 4 | 11 | 7 | 5 | 0 | 0 | 1 | 2 | 0 |

Antes de empezar el problema, vamos a analizar algunas de las operaciones que después utilizaremos en el programa. Por ejemplo, la expresión:

'z' - 'a' + 1 // Resultado 26, nro. De letras entre ambas incluidas

Da como resultado 26. Recuerde que cada carácter tiene asociado un valor entero (código Unicode) que es el que utiliza la máquina internamente para manipularlo. Así por ejemplo la 'z' tiene asociado el entero 122, la 'a' el 97, etc. Según esto, la evaluación de la expresión anterior es: $122 - 97 + 1 = 26$.

PROG – Programación

15. Haz un programa que use la función *bool EsPalindromo(string s)*, que devuelva un *true* o *false* si la cadena de texto pasada es palíndromo(*true*) o no que devolverá *false*.
16. Haz un programa, usando un *ArrayList*, para gestionar una lista de datos dinámicos enteros.
17. Haz un programa para gestionar una PILA de registros con los campos: *nombre, código, fecha*
18. Haz un programa para gestionar una COLA de caracteres.
19. Haz un programa para gestionar un diccionario de registros de datos personales (apellidos, nombre, teléfono) la clave es entera.
20. Haz un programa que simule un juego de cartas, ejemplo el de las siete y media u otro, que use una baraja española de 40 cartas (para las 7 y ½) y cuatro palos o de 52 cartas para otros juegos y haz un menú que permita:
 - Iniciar Juego Termina el juego en curso y comienza otro con un mazo nuevo de cartas sin barajar
 - Barajar las cartas Baraja el mazo de cartas
 - Extraer cartas Extrae una carta si la baraja no está vacía, la carta se mostrará
 - Etc.

Para este juego puedes usar la colección que creas más apropiada.

NOTA: Para probar las funciones tendrás que hacer un menú principal