

Programar en Java los siguientes algoritmos.

1. Un texto está formado por líneas de longitud variable con 80 caracteres como máximo. Se quiere que todas tengan la misma longitud que la de la cadena más larga; para ello se deben cargar con blancos por la derecha hasta completar la longitud requerida. Escribir un programa para leer un texto de líneas de longitud variable y formatear el texto para que todas tengan la longitud de la línea mayor.
2. Escribir un programa que encuentre dos cadenas introducidas por teclado que sean anagramas. Se considera que dos cadenas son anagramas si contienen exactamente los mismos caracteres, ya sea en el mismo o en diferente orden: ignorar los blancos y considerar que mayúsculas y minúsculas son iguales.
3. Se sabe que en las líneas de un texto hay valores numéricos enteros que representan los kilogramos de patatas recogidos en una finca; dichos valores están separados de las palabras por un blanco, o el carácter fin de línea. Escribir un programa que lea el texto y obtenga la suma de los valores numéricos.
4. Escribir un programa que lea líneas de texto, obtenga las palabras de cada una y las imprima en orden alfabético; considerar que el número máximo de palabras por línea es 28.
5. Leer una frase y determinar qué caracteres se encuentran repetidos, y cuántas veces aparece repetido cada carácter. Utilizar un vector con los caracteres que van apareciendo y otro correspondiente a la misma posición con el número de repeticiones.
6. Dada una frase comprobar si el balanceo de los paréntesis es correcto. Es decir, “(ccc(ccc)cc((ccc(c))))” es correcto, pero “)ccc(ccc)cc((ccc(c)))(“ no lo es, aunque tenga el mismo número de paréntesis abiertos que cerrados.
7. Dada una cadena de entrada, comprobar si es una contraseña FUERTE o DÉBIL. Se considera que una contraseña es fuerte si contiene 8 o más caracteres, y entre ellos al menos hay una mayúscula, una minúscula, un signo de puntuación y un dígito.
8. Cifrar una frase con código Cesar. En el cifrado en código cesar a cada carácter se le suma o resta un valor, si el valor es negativo, se escribe un carácter n posiciones para atrás y si es positivo para delante. Ejemplo Cesar +2: Hola jqnc
9. International Standard Book Number (en español, ‘número estándar internacional de libro’), abreviado ISBN, es un identificador único para cualquier libro cuyo uso se prevea de carácter comercial (ya en el curso se ha utilizado este identificador).

A continuación se explica cómo se descompone y se calcula el dígito de control que figura como último dígito.

Vamos a generar un método `static char testDigitoControlISBN(String strCadena)`

La cadena que se le pasa se compone de 9 dígitos:

- Código de grupo (1 dígitos)
- Código del editor (4 dígitos)
- Código del libro (4 dígitos)

Se puede crear un método que devuelva todo el ISBN (10 dígitos).

Por ejemplo, el código ISBN para un determinado libro es 0675209935. El carácter o dígito de control (5 en el ejemplo) se obtiene en dos pasos:

1. Se multiplica cada dígito por el índice correspondiente a su posición, y se suman los números resultantes. En el ejemplo:

0 6 7 5 2 0 9 9 3 x

1 2 3 4 5 6 7 8 9

-----

$0 + 12 + 21 + 20 + 10 + 0 + 63 + 72 + 27 = 225$  2. Se divide la suma por 11, y se toma el resto como dígito de control, teniendo en cuenta que si el resto es 10, se utiliza el carácter X como carácter de control. En el ejemplo, el resto de la división entera de 225 por 11 es 5, que es el dígito de control. **IMPORTANTE:** Observe que la función `char digitoControlISBN(StringD)` devuelve un tipo `char`, por lo que necesitará, de un lado, pasar a `int` los valores `char` que forman la cadena de entrada y, de otro, una vez calculado el dígito de control, pasarlo a `char` para que se pueda devolver en el `return`. Apóyese en que: • `(int)(carácter-'0')` convierte un dato de tipo carácter, que representa un número entero, en ese número como tipo de dato entero. • `(char)(entero+'0')` convierte un entero entre 0 y 9 en su representación como carácter