

Ejercicios métodos recursivos (2)

Ejercicio nº 1.

Deseamos codificar en Java, utilizando recursividad, un programa que nos permita obtener el mayor de los elementos de un array de una dimensión, por ejemplo:

25	14	3	12	35	9	20	1	0	-63	15	-48
----	----	---	----	----	---	----	---	---	-----	----	-----

Con el vector anterior, el programa deberá devolver «35»

Ejercicio nº 2.

Deseamos codificar en Java, utilizando recursividad, un programa que nos permita obtener la suma de los elementos de un vector, con el ejemplo anterior deberá devolver «23»

Ejercicio nº 3.

Deseamos codificar en Java, utilizando recursividad, un programa que nos permita dar la vuelta a una cadena de caracteres (Por ejemplo, a partir de “Hola” devolverá “aloH”)

Ejercicio nº 4.

Deseamos codificar en Java, utilizando recursividad, un programa que nos permita conocer si una cadena de caracteres es simétrica (un palíndromo). Por ejemplo “DABALEARROZALAZORRAELABAD” es un palíndromo

Ejercicio nº 5.

Deseamos codificar en Java, utilizando recursividad, un programa que nos permita obtener el número de secuencias de dos 1 seguidos que hay en una cadena de caracteres que represente un número en binario, por ejemplo:

Número introducido: 1 0 1 1 1 0 1 1 0 1 0 1 0

Número de secuencias: 3 (En la secuencia 1 1 1 se considera 2 secuencias de 1, ver según los colores siguientes: 11 1 y 1 11)

Ejercicio nº 6.

Dada la siguiente serie:

$$a_n = a_{n-1} + 2^n$$

deseamos codificar en Java, utilizando recursividad, un programa que nos permita obtener el término «n» de la serie anterior.

Ejercicio nº 7.

Deseamos codificar en Java, utilizando recursividad, un programa que nos permita calcular el cuadrado de un número entero n siendo: $n \geq 1$, mediante el siguiente método: el cuadrado de «n» es igual a la suma de los «n» primeros números impares. Por ejemplo; el cuadrado de 4 es $1 + 3 + 5 + 7 = 16$

Ejercicio nº 8.

Un número diabólico es un número que, cuando se pasa a base 2, tiene un número impar de 0's. Por ejemplo, el 0, el 2 (10), el 5 (101), el 6 (110), el 8 (1000), el 11(1011)

Se desea codificar en Java, utilizando recursividad, un programa que tras introducir un número «K», nos indique se es o no diabólico

Ejercicio nº 9.

Se dice que un número es apocalíptico si contiene entre sus dígitos la secuencia «666». Por ejemplo, el 16667234, el 6660, el 34666. El 6666666, El 1346786956, por ejemplo, NO lo es: los '6' no forman secuencia.

Deseamos codificar en Java, utilizando recursividad, un programa que nos permita introducir un número y nos comuniqué si es o no apocalíptico.

Ejercicio nº 10.

Deseamos codificar en Java, utilizando recursividad, un programa que nos permita calcular la suma de los dígitos pares de un número. Por ejemplo, dado el «16582234», debería devolver «22», ya que si obtenemos $6 + 8 + 2 + 2 + 4 = 22$.

Ejercicio nº 11.

Deseamos codificar en Java, utilizando recursividad, un programa que nos indique si todos los dígitos de un número se encuentran ordenados en orden creciente o no, por ejemplo: «156689», nos tendría que indicar que cumple y que todos sus dígitos están ordenados crecientemente, mientras que «1658923», no cumple.

Ejercicio nº 12.

Deseamos codificar en Java, utilizando recursividad, un programa que nos permita introducir una cadena de caracteres, que pueda contener cualquier carácter. Se deberá devolver una cadena idéntica a la original, pero en la que se haya eliminado todos los caracteres que no sean alfanuméricos. Es decir, se eliminan blancos, signos de puntuación, etc.

Por ejemplo si la cadena fuera: “*Vivo en el numero 13, calle Melancolia*”, nos deberá devolver “Vivoenelnúmero13calleMelancolia”