

Contando alternados

Dado un número natural n y un dígito d en n , d se llama *dígito alternado* cuando, bien es el primer dígito de n , bien está precedido por un dígito par si d es impar, o por un dígito impar si d es par. El número natural n se llama *alternado* cuando todos sus dígitos son dígitos alternados. Por ejemplo, 125834 es alternado, pero 25861 no lo es.

Debemos diseñar un algoritmo recursivo eficiente que cuente la cantidad de números naturales *alternados* menores que un k dado ($k \geq 0$).

Trabajo a realizar

Debe diseñarse el algoritmo recursivo pedido, completando los apartados indicados entre comentarios en el archivo `plantilla.cpp` que se proporciona como apoyo. Debe implementarse, además, el algoritmo. El punto de entrada al mismo será la función `num_alternados`. Si se considera necesario, deberá definirse e implementarse una generalización adecuada, y definir el algoritmo pedido como una inmersión de dicha generalización. Aparte de llevar a cabo el diseño del algoritmo, e implementar el mismo, deberás determinar justificadamente su complejidad, definiendo y resolviendo las recurrencias necesarias (la resolución se llevará a cabo utilizando los patrones de recurrencias genéricas discutidos en clase).

Programa de prueba

Se proporcionan un programa que lee por la entrada estándar enteros representables como valores del tipo `long long` (enteros que utilizan, como mínimo, 64 bits). Todos los enteros leídos serán no negativos, excepto el último, que será -1 y marcará el final de los casos de prueba. Cada vez que lee un caso de prueba, el programa invoca a `num_alternados` e imprime el resultado. A continuación, se muestra algunos ejemplos de entrada / salida:

Entrada	Salida
7	7
25	17
167	73
5678	874
-1	

El archivo completo debe entregarse a través del juez en línea de la asignatura.