Descripción

Daniel se encuentra aburrido en el avión que lo llevará a su primera participación en la IOI. Para pasar el tiempo, y como es su costumbre, saca su cuaderno y se pone a escribir los números enteros en orden ascendente, uno por uno, empezando desde el 1 hasta el cansancio. Luego de un par de horas, llegó hasta un número N y se detuvo por un momento. Orgulloso de la longitud la lista que había escrito, ahora se pregunta emocionado:

¿Si tomo un dígito d, cuántas veces aparece dicho dígito en la lista?

Tú también estás aburrido, pero detestas contar manualmente, así que decides escribir un programa que calcule dicho número antes que Daniel lo calcule.

Entrada

- En la primera línea el valor de N.
- En la segunda línea el valor de d.

Salida

Un solo entero, representando la cantidad de veces que aparece el dígito d en la lista de números del 1 al N (ambos inclusive).

Límites

- $1 \leq n \leq 10^{18}$.
- $1 \leq d \leq 9$.

Subtarea 1 [30 puntos]

$n \leq 10^6$.

Subtarea 2 [50 puntos]

$n \leq 10^9$.

Subtarea 3 [20 puntos]

Sin restricciones.

Solución:

La solución obvia contando los dígitos d de cada número tiene una complejidad de $O(n \log_{10} n)$ ya que para cada número iteramos sobre a lo mucho $\log_{10} n$ dígitos.

Para resolver el problema eficientemente, dividimos el algoritmo en "rondas". En la primera ronda, calculamos cuántos dígitos d aparecen en la posición de las unidades. Para dicho efecto, podemos dividir los números en intervalos $[d, 10 + d - 1), [10 + d, 20 + d - 1) \ldots$ hasta cubrir todos los números hasta n. Cada intervalo tiene exactamente un dígito d en las unidades. En la segunda ronda, contamos cuántos dígitos d aparecen en las decenas. Dividimos en intervalos $[10*d, 10*d+9], [100+10*d, 100+10*d+9], [200+10*d, 200+10*d+9] \ldots$ hasta cubrir todos los números hasta n. En cada uno de estos intervalos, el dígito d aparece exactamente 10 veces. Por ejemplo, si $d = 2$ la primera ronda se consideran los intervalos $[2, 11], [12, 21], [22, 31], \ldots$. En cada uno de estos intervalos el dígito 2 aparece exactamente una vez en las unidades. Para las decenas, dividimos en intervalos $[20, 29], [120, 129], [220, 229], \ldots$. En cada uno de estos intervalos el valor de 2 aparece exactamente 10 veces en las decenas. Por lo tanto la solución se reduce a contar cuántos intervalos de cada tipo existen y multiplicar por el número de dígitos correspondiente. Continuamos con este proceso hasta cubrir todos las posiciones de dígitos en n. Cada ronda toma tiempo constante, y existen $\log_{10} n$ dígitos en n. Por lo tanto, la complejidad de este algoritmo es $O(\log_{10} n)$.