

Aplicando operaciones

Contribución de David Lescano

Descripción del problema

Benjamín se encuentra estudiando una carrera de matemática en una importante universidad. Esta se caracteriza por ser muy exigente y el estudio le está costando bastante ya que, por falta de tiempo, no practica mucho. Sabe que debe hacerlo para poder aprobar.

Investigando por internet logró dar con la OIA (Organización Internacional de Aritmética) y allí conoció a Matías, un programador reconocido por sus grandes logros en olimpiadas de todo tipo.

Matías le recomendó que aprendiera a programar para **desarrollar un programa** que lo ayudara con la práctica de **calcular el resto en la división de un número N por otro D** .

Benjamín ha hecho su programa, pero parece que este no funciona bien, pues **el número N es demasiado grande**.

Más aún, Benjamín no se conforma con calcular el resto de N en la división por D , sino que quiere ser capaz de calcular el resto de **cualquier subnúmero** de N .

Un subnúmero de N se forma tomando todos los dígitos de N desde el a -ésimo hasta el b -ésimo, para ciertos valores de a y b . Por ejemplo, si $a = 2$ y $b = 5$, el correspondiente subnúmero de 1834323 es 8343.

Debes escribir una función que, dados los valores de N y D , y múltiples valores de a y b , calcule en cada caso el resto del correspondiente subnúmero en la división por D .

Nota: El “resto” de la división o “módulo” es lo que queda después de repartir, y en los lenguajes de programación habitualmente se escribe con el caracter %.

Por ejemplo, si el subnúmero es 14 y $D = 3$, el resto es 2, pues se reparten 4 unidades entre 3, y sobran así $14 - 4 \cdot 3 = 2$ unidades. Es decir, $14 \% 3 == 2$.

Detalles de implementación

Debes implementar la función `aplicando(n, d, a, b)` que reciba un string n (el número N que hay que dividir), un entero d (el divisor D) y los arreglos a y b , de Q enteros cada uno.

La función debe retornar un arreglo de Q enteros: Para cada $0 \leq i \leq Q - 1$, el entero en la posición i de la respuesta deberá ser el resto en la división por D del subnúmero de N que comienza en $a[i]$ y termina en $b[i]$.

Notar que como N puede ser muy grande, a diferencia de D , **se recibe n como cadena de texto** y no como número directamente.

Se garantiza que el número N no tiene ceros a la izquierda.

Evaluador local

El evaluador local lee de la entrada estándar:

- Una línea con los números N y D .
- Una línea con el entero Q .
- Q líneas, cada una con 2 enteros. La i -ésima de estas líneas (contando desde 0) contiene los valores $a[i]$ y $b[i]$.

Escribe a la salida estándar el arreglo de valores retornado por la función `aplicando`.

Cotas

- $1 \leq N < 10^{300.000}$ (es decir, N no tiene más de 300.000 cifras)
- $1 \leq D \leq 1.000.001.000$
- $1 \leq Q \leq 300.000$
- $1 \leq a[i] \leq b[i] \leq \text{cifras}(N)$

Ejemplos

Si el evaluador local recibe la siguiente entrada:

```
123456789 5
3
1 3
2 6
2 9
```

Para un programa correcto, la salida debe ser:

```
3
1
4
```

Si el evaluador local recibe la siguiente entrada:

```
123456789101112 1000000007
5
1 15
6 15
1 9
2 5
7 7
```

Para un programa correcto, la salida debe ser:

```
788236920
789101070
123456789
2345
7
```

Subtareas

- $D = 1$ (3 puntos)
- $D = 2$ (4 puntos)
- $D = 8$ (10 puntos)
- $D = 3$ o $D = 9$ (18 puntos)
- $N \leq 1.000.000$, $Q \leq 1000$ (11 puntos)
- $N < 10^{1.000}$, $Q \leq 1000$ (20 puntos)
- $a[i] = 1$ para todo i (22 puntos)
- Sin más restricciones (12 puntos)