

## Dominando 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.

Para mayor práctica, entre medio del proceso de ir calculando restos de los subnúmeros, Benjamín **realiza modificaciones** al número inicial  $N$ . En estas modificaciones, elige un cierto dígito de  $N$  y lo cambia por otro.

Debes escribir una función que, dados los valores de  $N$  y  $D$ , y el listado de los  $Q$  cálculos y modificaciones a realizar, calcule en cada caso el resto del correspondiente subnúmero en la división por  $D$ .

### Detalles de implementación

Debes implementar la función `dominando( $n, d, t, a, b$ )`, con parámetros:

- Un string  $n$ , el número  $N$  **inicial**.
- Un entero  $d$ , el divisor  $D$
- Arreglos  $t, a, b$ , todos de longitud  $Q$ . La operación  $i$ -ésima (contando desde 0), viene dada por  $(t[i], a[i], b[i])$ :
  - Si  $t[i] = 0$ , se debe **calcular** el resto del subnúmero que va desde el  $a[i]$ -ésimo dígito hasta el  $b[i]$ -ésimo.
  - Si  $t[i] = 1$ , se **modifica** el número  $N$ , cambiando el dígito en la posición  $a[i]$  por  $b[i]$ .

La función debe retornar un arreglo con el resultado de cada operación de *calcular*, en el mismo orden en que las operaciones se reciben en los parámetros de entrada.

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  $N$  nunca tendrá ceros a la izquierda (ni al comienzo, ni luego de ninguna operación).

### 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 3 enteros. La  $i$ -ésima de estas líneas (contando desde 0) contiene los valores  $t[i], a[i]$  y  $b[i]$ .

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

**Cotas**

- $1 \leq N < 10^{300.000}$  (es decir,  $N$  no tiene más de 300.000 cifras)
- $1 \leq D \leq 1.000.001.000$
- $1 \leq Q \leq 100.000$
- Si  $t[i] = 0$ :
  - $1 \leq a[i] \leq b[i] \leq \text{cifras}(N)$
- Si  $t[i] = 1$ :
  - $1 \leq a[i] \leq \text{cifras}(N)$
  - $0 \leq b[i] \leq 9$

**Ejemplos**

Si el evaluador local recibe la siguiente entrada:

```
123456789 10000000
5
0 1 3
1 3 9
0 2 6
1 7 4
0 1 9
```

Para un programa correcto, la salida debe ser:

```
123
29456
9456489
```

Si en cambio recibe la siguiente:

```
123456789101112 1000000007
10
1 1 7
0 1 15
0 6 15
1 9 8
0 1 9
0 2 5
1 4 4
0 7 7
1 7 6
1 6 7
```

La salida debe ser:

```
784036920
789101070
723456788
2345
7
```

**Subtareas**

- $D = 1$  (3 puntos)
- $D = 2$  (4 puntos)
- $D = 8$  (9 puntos)
- $D = 3$  o  $D = 9$  (15 puntos)
- $N \leq 1.000.000, Q \leq 1000$  (10 puntos)
- $N < 10^{1.000}, Q \leq 1000$  (17 puntos)
- $t[i] = 0$  para todo  $i$  (19 puntos)
- Sin más restricciones (23 puntos)