

## Aprendiendo 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**.

Debes escribir una función que, dados los valores de  $N$  y  $D$ , calcule el resto que queda al dividir  $N$  entre  $D$ .

**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 carácter %.

Por ejemplo, si  $N = 14$  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 `aprendiendo(n, d)` que reciba un string  $n$  (el número  $N$  que hay que dividir) y un entero  $d$  (el divisor  $D$ ), y retorne en un entero el resto de dividir a  $N$  por  $D$ .

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  $D$ .

Escribe a la salida estándar el valor retornado por la función `aprendiendo`.

### Cotas

- $1 \leq N < 10^{1000}$  (es decir,  $N$  no tiene más de mil cifras)
- $1 \leq D \leq 1.000.000$

**Ejemplos**

Si el evaluador local recibe la siguiente entrada:

```
1232324233234345653432789 5
```

Para un programa correcto, la salida debe ser:

```
4
```

Si en cambio recibe la siguiente entrada:

```
8000 1007
```

Para un programa correcto, la salida debe ser:

```
951
```

**Subtareas**

- $D = 1$  (3 puntos)
- $D = 2$  (4 puntos)
- $D = 5$  (5 puntos)
- $D = 10$  (6 puntos)
- $D = 4$  (6 puntos)
- $D = 8$  (10 puntos)
- $D = 3$  (10 puntos)
- $D = 9$  (11 puntos)
- $N \leq 1.000.000$  (13 puntos)
- Sin más restricciones (32 puntos)