

# Búsqueda de polidivisibles

Los números *polidivisibles* son aquellos números que:

- Son mayores que cero.
- El número formado por su primer dígito es múltiplo de 1 (esto lo cumplen todos los números).
- El número formado por sus dos primeros dígitos es múltiplo de 2.
- El número formado por sus tres primeros dígitos es múltiplo de 3.
- El número formado por sus cuatro primeros dígitos es múltiplo de 4.
- ...
- En general, el número formado por sus  $K$  primeros dígitos es múltiplo de  $K$ . Se debe asumir que los números se escriben en base 10 y sin ceros a la izquierda.

Por ejemplo, el número 2.016 es polidivisible, pues es mayor que cero y 2 es divisible por 1, 20 lo es por 2, 201 por 3 y, por último, el propio 2.016 es divisible por 4. Sin embargo, el número 2.225 no es polidivisible pues a pesar de que el 2 es divisible por 1, el 22 lo es por 2 y el 222 por 3, el propio 2.225 no es divisible por 4.

Sorprendentemente la cantidad de números polidivisibles no es infinito. De hecho hay únicamente 20.456 números polidivisibles, el mayor de ellos de 25 dígitos.

El mundo de las matemáticas no nos tiene muy acostumbrados a series finitas de números. Para corroborar que efectivamente el conjunto total no es infinito queremos empezar por ser capaces de generar los números polidivisibles. Dado un número polidivisible  $N$  y una cantidad máxima de dígitos  $D$ , queremos obtener todos los números polidivisibles que comiencen por  $N$  y tengan como mucho  $D$  dígitos.

## Entrada

La entrada estará compuesta por distintos casos de prueba. Cada uno de ellos se compone de una línea que contiene dos números,  $N$  ( $0 < N < 10^{18}$ ) y  $D$  ( $0 < D \leq 18$ ), que indican el comienzo (prefijo) del número y la cantidad máxima de dígitos de los polidivisibles a generar. Se garantiza que  $N$  es un número polidivisible y que  $D$  será siempre mayor o igual que el número de dígitos del propio  $N$  (es decir, se obtendrá siempre al menos un número polidivisible, el  $N$  leído).

## Salida

Para cada caso de prueba se escribirán todos los números polidivisibles que comiencen con  $N$  y tengan como mucho  $D$  dígitos. Deberán aparecer en *orden lexicográfico* (o “alfabético”) y en líneas independientes.

Cada caso de prueba terminará con una línea con tres guiones, ---.

## Entrada de ejemplo

2016 4
2016 5
2016 6

## Salida de ejemplo

```
2016
---
2016
20160
20165
---
2016
20160
201600
201606
20165
201654
---
```

## Notas

Debes implementar una función *no recursiva*:

```
void escribePolidivisibles(long long raiz, int maxDigitos);
```

que **debe llamar a otra función recursiva** que no necesite “trocear” el número que recibe.

La única escritura debe hacerse en esa función recursiva (a excepción de los --- que irán en el **resuelve** o equivalente).

## Nota

Este ejercicio debe verse en el contexto de la asignatura de Estructura de Datos y Algoritmos (EDA), FDI-UCM 2017/2018 (prof. Marco Antonio Gómez Martín). Por tanto *no* vale cualquier solución, sino sólo aquellas que utilicen los conceptos de EDA. Es muy posible que se den aclaraciones adicionales en clase a este respecto.