

## Números excluyentes

### El problema

Se dice que un natural *excluye* a un dígito  $d$  cuando dicho número *no* contiene a  $d$ .

Debe diseñarse un algoritmo recursivo que, dado un número natural  $n \geq 0$  y un dígito  $d$ , encuentre la cantidad de números naturales menores que  $n$  que *excluyen* a  $d$ .

### Trabajo a realizar

Para realizar el control se proporciona un archivo `plantilla.cpp`. El código proporcionado supone que el punto de entrada al algoritmo es la función `num_sin_digito_menoresque`. Si se comenta la definición `#define DOM_JUDGE` el resultado del algoritmo se contrasta, sobre 1000 casos de prueba generados aleatoriamente, con una implementación *naïf* del mismo. La definición debe estar descomentada a la hora de subirla al juez.

Cada caso de prueba consiste en el número  $n$ , seguido del dígito  $d$ . Por ejemplo, `27 5` ( $n=27$ ,  $d=5$ ). Los casos terminan con una línea que contiene `-1`.

Ejemplo de entrada / salida:

Entrada	Salida
5 4	4
5 6	5
27 5	24
27 0	24
145 4	117
145 6	122
145 0	121
1456789543444 8	428051810655
-1	

Tu trabajo consiste en:

- Diseñar el algoritmo, rellenando los comentarios incluidos en el archivo.
- Implementar este algoritmo.
- Entregar la solución a través del juez en línea de la asignatura.

### Importante:

- No modificar el código proporcionado. Únicamente deben responderse a los distintos apartados, en el interior de los comentarios, implementar la función `num_sin_digito_menoresque` y todas las funciones auxiliares que se consideren necesarias. Si se quiere que el programa se ejecute en modo *autocorrección*, puede comentarse `#define DOM_JUDGE`. Pero debe recordarse que dicha definición debe estar descomentada cuando se realice la entrega en el juez.