

## Números exclusivos

### El problema

Se dice que un número natural es *exclusivo* cuando no contiene dígitos repetidos.

Debe diseñarse un algoritmo recursivo que, dado un número  $n \geq 0$ , encuentre la cantidad de números *exclusivos* menores que  $n$ .

**Recomendación:** Es conveniente abordar este problema considerando que, cuando  $n > 0$ , la solución puede obtenerse sumando:

- (i) La cantidad de números positivos exclusivos menores que  $n$  que tienen exactamente el mismo número de cifras que  $n$ .
- (ii) La cantidad de números positivos exclusivos que tienen menos cifras que  $n$ .
- (iii) 1 (para contabilizar el 0)

En caso de que  $n=0$ , la solución es directamente 0.

Por tanto, se recomienda abordar separadamente el diseño de algoritmos para resolver los problemas (i) y (2), y después unificar ambos algoritmos en una única generalización. Además, el diseño de ambos algoritmos conducirá a varias funciones recursivas auxiliares, que también pueden integrarse en dicha generalización.

### 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_sinrepetidos_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 des-comentada a la hora de subirla al juez.

Cada caso de prueba consiste en el número  $n$ . Los casos terminan con una línea que contiene -1.

Ejemplo de entrada / salida:

Entrada	Salida
8	8
96	88
5698	3090
193456785320	8877691
-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_sinrepetidos_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 des-comentada cuando se realice la entrega en el juez.