



Número con cifras decrecientes

Dado un número queremos eliminar los dígitos que no superen a todos los dígitos que quedan a su derecha. Por ejemplo, dado el número 4321 todos sus dígitos igualan o superan a los que quedan a su derecha porque $4 \geq 3 \geq 2 \geq 1$ por lo que no se elimina ninguno de ellos. Sin embargo, dado el número 475241, el dígito 2 no supera al 4 y por lo tanto se debe eliminar, lo mismo ocurre con el dígito 4 de mayor peso. El número resultante es 7541.

El número resultante siempre tendrá sus dígitos ordenados de forma decreciente según su valor. Observa también que dado un número, no queremos cualquier posible transformación que deje sus dígitos en orden decreciente, sino la que lo hace eliminando los dígitos que no superen a los dígitos que quedan a su derecha.

Requisitos de implementación.

- El problema debe resolverse con una función que sea **recursiva final**.
- Debes incluir también la recursión del coste y decir qué coste tiene tu función.
- La función recursiva debe recibir un número entero, y devolver como valor de retorno el número pedido.
- No se pueden utilizar los tipos `std::string`, ni `std::vector` para resolver el ejercicio.

Entrada

La entrada comienza con el número de casos de prueba. Cada caso se escribe en una línea y consiste en un número, $0 \leq N \leq 100.000.000$.

Salida

Para cada caso de prueba se muestra en una línea el resultado obtenido con la función recursiva final.

Entrada de ejemplo

```
6
4321
1234
1000
888
64839
9576135
```

Salida de ejemplo

```
4321
4
1000
888
9
9765
```

Autor: Isabel Pita (modificado por Ignacio Fábregas)