

Formar números con sus 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 que el dígito 4 de mayor peso. El número resultante es 7541.

~~4~~ 7 5 ~~2~~ 4 1

El número resultante siempre tendrá sus dígitos ordenados de forma decreciente según su valor. Observad que dado un número, no queremos cualquier posible transformación que deje sus dígitos en orden decreciente, sino la que se obtiene eliminando los dígitos que no superen a los dígitos que quedan a su derecha. Por ejemplo, dado el número 4231, el resultado de la función debe ser el número 431, ya que el dígito 2 es el que no supera a los dígitos de su derecha.

El problema debe resolverse dos veces. La primera con una función recursiva no final y la segunda con una función recursiva final.

Requisitos de implementación.

Se deben escribir las ecuaciones recursivas que resuelven el problema en el fichero que se entrega.

Las funciones recursivas deben recibir un número entero, y devolver como valor de retorno el número obtenido al eliminar todos los dígitos que no superan a los que quedan a su derecha. Se pueden utilizar más parámetros por valor si la solución lo requiere.

No se deben utilizar parámetros por referencia en ninguna de las funciones implementadas. Se pueden devolver más valores como retorno de la función si la solución lo requiere.

El problema se debe resolver utilizando un número entero. No se puede utilizar el tipo `std::string`, ni el tipo `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 no final y a continuación el resultado de la función recursiva final.

Entrada de ejemplo

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

Salida de ejemplo

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

Autor: Isabel Pita