

## Compressor

Estás trabando en un programa para comprimir documentos de texto. Escribe una función que encuentre la subcadena repetida más larga dentro de un fragmento de texto, tal que las dos ocurrencias de la subcadena no se superpongan. Tu programa distingue entre mayúsculas y minúsculas. Si hay más de una opción para la elección de subcadena, escoge aquella que ocurre antes en el texto.

Por ejemplo, dada la cadena “ABCDABCFG”, la aparición repetida más larga es “ABC”. En la cadena “ABABA”, tanto “AB” como “BA” son subcadenas repetidas, por lo que escogemos “AB” por aparecer en primer lugar. (Aunque “ABA” aparece dos veces como subcadena, las dos ocurrencias tienen una letra en común y no pueden usarse.)

### Entrada

La entrada contiene diversas líneas. Cada línea contiene un texto a considerar. Cada texto contiene entre 1 y 50 caracteres entre ‘A’-‘Z’, ‘a’-‘z’, ‘0’-‘9’ y ‘ ’. (Pista: `getline(cin,s)` lee una línea entera de texto y la guarda en la variable de tipo string `s`.)

### Salida

Por cada texto de la entrada, una línea conteniendo la subcadena repetida más larga. Si ninguna subcadena se repite, escribe una línea vacía.

#### Ejemplo de entrada 1

This is a test.

#### Ejemplo de salida 1

is

#### Ejemplo de entrada 2

Testing testing 1 2 3.

#### Ejemplo de salida 2

esting

#### Ejemplo de entrada 3

The quick brown fox jumps over the lazy dog.

#### Ejemplo de salida 3

he