

## Errores de tipeo

*Contribución de Natalia Pérez y Carlos Mendioroz*

### Descripción del problema

Un importante periódico ha decidido agregar un suplemento dedicado a la medicina del deporte en su edición dominical. Dado el vocabulario algo específico que se maneja, el jefe de redacción necesita contratar nuevos dactilógrafos.

Para juzgar la calidad de los postulantes que se han presentado, les pide copiar un texto extenso sobre el tema, sabiendo que éste mide a lo más **1.000.000** caracteres.

Luego de revisar los textos mecanografiados, el jefe ha concluido que, de haber errores, estos se reducen a un desorden de las letras. De esta manera, conociendo cuántas permutaciones entre letras vecinas hacen falta para dejarlo en orden, es capaz de establecer un orden entre los candidatos para elegir criteriosamente el mejor.

Para ayudar al jefe de redacción en la evaluación de los textos, se te pide que escribas una función:

`procesador( texto, prueba )`

cuyos parámetros proveen:

**texto** : el texto correcto y

**prueba** : el texto mecanografiado por un postulante (que es una permutación de los mismos caracteres).

Determine la menor cantidad de movimientos de letras que debe realizar el editor para poder corregirla. Dichos movimientos consisten en intercambiar dos letras consecutivas.

Cada texto ha sido redactados usando caracteres ASCII del rango 32 a 126 inclusivos. (Se diferencian mayúsculas de minúsculas.)

### Ejemplo

Si los datos fueran:

<b>Texto</b>	Estamos pintando!
<b>Prueba</b>	ostant!Em inpados

La respuesta deberá ser 40.

### Detalles de implementación

En un único archivo, llamado `procesador.cpp`, `procesador.c`, o `procesador.pas`. debes enviar una función que implemente la función descrita antes, usando los siguientes prototipos:

**En C/C++** `long long procesador( char texto [], char prueba[] )`

Debes incluir también un archivo de encabezamiento `procesador.h`

### Para programadores Pascal

**En Pascal** `function procesador( var texto, prueba: ansistring ): int64;`

### Evaluador local

El evaluador local (programa para probar ejemplos propios) lee la entrada por `stdin` en el siguiente formato:

Una línea por cada texto y entrega por `stdout` el valor retornado por la función

### Puntaje

Habrán casos de prueba por un total de **15** puntos en cuyos textos aparecerán solamente dos caracteres del repertorio indicado.

Habrán otro juego de datos por un total de **40** puntos en los cuales el tamaño de los textos no pasará de **330.000** caracteres.