

# Compilando texto HTML

Un texto HTML comienza con la etiqueta `<HTML>` y termina con la etiqueta `</HTML>`. El lenguaje nos permite formatear documentos utilizando etiquetas como `<p>` y `</p>` para delimitar los párrafos o `<b>` y `</b>` para delimitar el texto que se debe mostrar en “negrita”. Dentro de un bloque se puede definir otro. Por ejemplo, dentro de un párrafo pueden aparecer textos en negrita y dentro del texto en negrita podría aparecer un texto en cursiva.

En el problema se pide implementar un programa que dado un texto HTML decida si está bien formateado, es decir si cada etiqueta que comienza un bloque tiene su correspondiente etiqueta finalizando el bloque. Además debe comprobar que las etiquetas no se solapan, es decir, todos los bloques internos a otro deben estar cerrados cuando se cierra el bloque externo, no puede quedar ninguna etiqueta sin cerrar, ni cerrar etiquetas que no hayan sido abiertas.

Para simplificar el programa, se consideran sólo cuatro tipos de etiquetas: `<p>` ; `</p>`, `<b>` ; `</b>`, `<i>` ; `</i>` y `<q>` ; `</q>`

*Requisitos de implementación.*

En la implementación del problema, se pueden utilizar los tipos *stack* y *queue* de la librería STL. Se utilizarán las estructuras más eficientes teniendo en cuenta como vienen dados los datos de entrada y el uso que se hace de ellos.

El texto HTML de la entrada se leerá en un *string*. Se pueden utilizar las siguientes operaciones sobre el tipo *string*:

- *string substr(size\_t pos = 0, size\_t len = npos) const;* donde *pos* es la posición del primer carácter que se copiará al substring. El primer carácter de la cadena es el cero. *len* es el número de caracteres que se incluyen en el substring; la constante *npos* indica todos los caracteres hasta el final de la cadena y *size\_t* es un entero sin signo. Dada una cadena *s* si queremos obtener una subcadena que empiece en la posición *i* con tres caracteres ejecutaremos la instrucción *s.substr(i,3)*

- El operador `[]` que obtiene el carácter *i*-ésimo de la cadena, se utiliza igual que en el caso de los vectores. Dada una cadena *s* para acceder al carácter *i*-ésimo haremos *s[i]*.

## Entrada

La entrada consta de varios casos de prueba. Cada uno de ellos se escribe en una línea y termina con el símbolo de final de línea `'\n'`.

Cada caso comienza con la etiqueta `<HTML>` y finaliza con la etiqueta `</HTML>`. Esto no hay que comprobarlo y nos sirve para garantizar que siempre hay texto antes y después de la primera/última etiqueta que se considera en el estudio.

## Salida

Para cada caso de prueba, si está bien formateado, se escribe SI seguido del número de párrafos (bloques `<p>` ; `</p>`), del número de bloques en negrita (`<b>` ; `</b>`), el número de bloques en cursiva (`<i>` ; `</i>`) y el número de citas (bloques `<q>` ; `</q>`) separados por un carácter blanco y NO en caso de que no esté bien formateado.

## Entrada de ejemplo

```
<HTML> <p>Este es el primer <b>parrafo</b> de este <i>texto</i></p>. </HTML>
<HTML><puedo tener/>simbolos que no sean delimitadores<j>.No daran <<error>>.</HTML>
<HTML>Puedo tener texto <q>antes</q> y <q>despues</q> del delimitador</HTML>
<HTML><p>Pero <q><b>siempre</b> debo cerrar todos los bloques</p></HTML>
<HTML> y <i> no dejar nada abierto</HTML>
<HTML><p>Tampoco puedo cerrar de mas</p></b></HTML>
```



### Salida de ejemplo

```
SI 1 1 1 0
SI 0 0 0 0
SI 0 0 0 2
NO
NO
NO
```

**Autor:** Isabel Pita