

## Balanceando cadenas

**NOTA:** Si usted está leyendo este documento sin haber extraído el compactado que se le entregó, ciérrelo ahora, extraiga todos los archivos en el escritorio, y siga trabajando desde ahí. Es un error común trabajar en la solución dentro del compactado, lo cual provoca que los cambios no se guarden. Si usted comete este error y entrega una solución vacía, no tendrá oportunidad de reclamar.

De forma análoga a la definición del Conjunto de Cadenas de Paréntesis  $()$  Balanceados es posible definir el Conjunto de Cadenas de Paréntesis  $()$ , Llaves  $\{\}$  y Corchetes  $[]$  Balanceados:

Los elementos del **Conjunto de Cadenas de Paréntesis, Llaves y Corchetes Balanceados** son cadenas formadas solamente con los caracteres '(', ')', '{', '}', '[', ']' y se cumple que:

- 1- Las cadenas  $()$ ,  $\{\}$  y  $[]$  pertenecen al conjunto
- 2- Si la cadena  $\alpha$  pertenece al conjunto entonces las cadenas  $(\alpha)$ ,  $\{\alpha\}$  y  $[\alpha]$  también pertenecen al conjunto
- 3- Si las cadenas  $\alpha$  y  $\beta$  pertenecen al Conjunto entonces su concatenación  $\alpha\beta$  también pertenece al conjunto

Algunos ejemplos de cadenas que pertenecen y que no pertenecen a este conjunto son:

Cadenas en el conjunto	Cadenas que no pertenecen al conjunto
$()\{\}[]$	$(())<>$
$((()))$	$(($
$\{\}\{\}\{\}\{\}$	$[[[[]]]]$
$[]$	$\}\{$
$([])\{\}$	$([])$
$[({}[]\{\})]$	$\{[]\}$
$((\{\}[[[]])\{\})]$	$\{[]\}\{\{\}\}$

Si se considera en una cadena de paréntesis, llaves y corchetes la operación de **reemplazar un carácter por otro cualquiera** entre '(', ')', '{', '}', '[', ']' entonces es posible que luego de una secuencia de operaciones se obtenga una cadena balanceada.

A continuación la columna de la izquierda muestra algunos ejemplos válidos de la aplicación de **exactamente una operación** y la columna de la derecha muestra reemplazos erróneos de la aplicación de una operación.

Operaciones Válidas	Operaciones Inválidas
$\{[]\}[\{\{\}\}] \Rightarrow \{[]\}\{\{\}\}$	$\{[]\}[\{\{\}\}] \Rightarrow \{[]\}\{\{\}\}$
$\{[]\}\{\{\}\} \Rightarrow \{[]\}\{\{\}\}$	$\{[]\}[\{\{\}\}] \Rightarrow \{[]\}\{\{\}\}$
$\{[]\}\{\{\}\} \Rightarrow \{[]\}\{\{\}\}$	$\{\}$ $\Rightarrow$ $\{\}$
$\{\}$ $\Rightarrow$ $[\ ]$	$\{(\}$ $\Rightarrow$ $\{\}$
$[\ ]$ $\Rightarrow$ $[\ ]$	
$\}\}$ $\Rightarrow$ $\}\}$	

El problema consiste en dada una cadena compuesta solamente por '(', ')', '{', '}', '[' y ']' determinar la **menor** cantidad de operaciones que hay que realizar para balancearla. También es necesario detectar el caso en que no es posible. Por ejemplo:

"{}" => no hay forma de balancearla con la operación definida

"}}{}" => "{ }{}" => "([[])" => "([[])" => 3 operaciones (no es óptimo)

"}}{}" => "{ }{}" => "{ }()" => 2 operaciones (óptimo) [solución 1]

"}}{}" => "{ }{}" => "{ }[]" => 2 operaciones (óptimo) [solución 2]

Usted debe haber recibido junto a este documento una solución de Visual Studio con dos proyectos: una biblioteca de clases (*Class Library*) y una aplicación de consola (*Console Application*). Usted debe implementar el método `MinOperacionesParaBalancear` que se encuentra en la clase `CadenasBalanceadas` en el *namespace* `Weboo.Examen`. En la biblioteca de clases encontrará la siguiente definición:

```
namespace Weboo.Examen
{
    public class CadenasBalanceadas
    {
        public static int MinOperacionesParaBalancear(string s)
        {
            //Borre la siguiente línea y escriba su código
            throw new NotImplementedException();
        }
    }
}
```

Este método tiene como parámetro un `string` que representa una cadena formada por '(', ')', '{', '}', '[' y ']' que se quiere balancear aplicando reemplazos, de uno en uno, como fue definido. Este método que Ud debe implementar tiene que computar **la menor cantidad de operaciones que son suficientes para balancear la cadena**. De no existir forma de balancear la cadena debe devolver el valor **-1**.

Puede asumir que:

- El parámetro `s` **siempre** tendrá al menos 1 carácter. La cadena solo estará formada por '(', ')', '{', '}', '[' y ']' de modo que no hay que verificar que puedan haber otro tipo de caracteres.

**ATENCIÓN:** La nota del examen puede verse afectada por el tiempo que consuma evaluar su solución. Esto está directamente relacionado a cuán eficiente sea la idea implementada. Por lo general una solución por fuerza bruta es menos eficiente que otras estrategias más inteligentes. Como referencia sepa que la solución esperada computa en un segundo la respuesta correcta para cadenas de longitud 20.

NOTA: Todo el código de la solución debe estar en este proyecto (biblioteca de clases), pues es el único código que será evaluado. Usted puede adicionar todo el código que considere necesario, pero no puede cambiar los nombres del namespace, clase o método mostrados. De lo contrario, el probador automático fallará. En particular, es imprescindible que usted no cambie los parámetros del método `MinOperacionesParaBalancear`, ni su orden. Por supuesto, usted puede (y debe) adicionar todo el código que necesite.

## Ejemplos

```
//EJEMPLO 1
string s1 = "{}{}";
int r1 = CadenasBalanceadas.MinOperacionesParaBalancear(s1);
//r1 = -1

//EJEMPLO 2
string s2 = "}}{}";
int r2 = CadenasBalanceadas.MinOperacionesParaBalancear(s2);
//r2 = 2 => {}[]

//EJEMPLO 3
string s3 = "({[])}{}";
int r3 = CadenasBalanceadas.MinOperacionesParaBalancear(s3);
//r3 = 0

//EJEMPLO 4
string s4 = "]}({){}{}";
int r4 = CadenasBalanceadas.MinOperacionesParaBalancear(s4);
//r4 = 4 => [{}]{[]}

//EJEMPLO 5
string s5 = "{}[]{}({[])}";
int r5 = CadenasBalanceadas.MinOperacionesParaBalancear(s5);
//r5 = 3 => {}[]({[]()})

//EJEMPLO 6
string s6 = ")][]({({[]}[])}";
int r6 = CadenasBalanceadas.MinOperacionesParaBalancear(s6);
//r6 = 6 => ()[]{}(){}[]()

//EJEMPLO 7
string s7 = ")]({[])}({({[]}[])}";
int r7 = CadenasBalanceadas.MinOperacionesParaBalancear(s7);
//r7 = 9 => []()[]{}({[]}[](){})
```

NOTA: Los casos de prueba que aparecen en este proyecto son solamente de ejemplo. Que usted obtenga resultados correctos con estos casos no es garantía de que su solución sea correcta y de buenos resultados con otros ejemplos. De modo que usted debe probar con todos los casos que considere convenientes para comprobar la validez de su implementación.