

Universidad de la Habana, Facultad de Matemática y Computación
Carrera Ciencia de la Computación, Asignatura Programación
1er Problema de Programación, Curso 2006 – 2007

Implemente una DLL de nombre **Problema1.DLL** que tenga la clase siguiente

```
namespace Problema1
{
    public class Prueba
    {
        public static string[] Justificar(string[] lineas, int ancho)
        {
            //... Aquí va su implementación
        }
    }
}
```

El método `Justificar` recibe un parámetro `lineas` que es un array de strings, y debe devolver un nuevo array de strings resultante de “justificar” la cadena que hay en cada elemento del array al `ancho` que se le indica como parámetro.

La justificación de una cadena consistirá en obtener a partir de ésta una cadena que corresponda a haber insertado equilibradamente suficientes espacios para separar las palabras de modo que la longitud de la línea coincida exactamente con el ancho que se haya solicitado

Ejemplo si una línea original fuese la cadena (que tiene ancho **24**)

E	n	u	n	l	u	g	a	r	d	e	L	a	M	a	n	c	h	a
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

y se pudiese justificar a un ancho **31**, entonces podrían ser justificaciones correctas las siguientes (las cuadrículas sombreadas nos indican el equivalente a los espacios insertados con respecto a la cadena original)

E	n			u	n				l	u	g	a	r			d	e			L	a				M	a	n	c	h	a
---	---	--	--	---	---	--	--	--	---	---	---	---	---	--	--	---	---	--	--	---	---	--	--	--	---	---	---	---	---	---

E	n			u	n			l	u	g	a	r				d	e				L	a			M	a	n	c	h	a
---	---	--	--	---	---	--	--	---	---	---	---	---	--	--	--	---	---	--	--	--	---	---	--	--	---	---	---	---	---	---

Note que, sin embargo, los siguientes intentos de justificación son incorrectos, bien porque no hay un balanceo de los espacios (se insertaron tres espacios en un tramo mientras en otro solo se insertó un espacio) o porque se ha colocado un espacio en un lugar inadecuado

E	n			u	n					l	u	g	a	r			d	e			L	a			M	a	n	c	h	a
---	---	--	--	---	---	--	--	--	--	---	---	---	---	---	--	--	---	---	--	--	---	---	--	--	---	---	---	---	---	---

4 espacios en total, se
insertaron 3 espacios

E	n			u	n			l	u		g	a	r			d	e					L	a			M	a	n	c	h	a
---	---	--	--	---	---	--	--	---	---	--	---	---	---	--	--	---	---	--	--	--	--	---	---	--	--	---	---	---	---	---	---

No se puede insertar un
espacio aquí

La justificación debe seguir las siguientes reglas

- Solo se considerarán los espacios como separadores de palabras. Los espacios sólo serán insertados junto a otros espacios y nunca entre dos caracteres que sean distintos del espacio (2do caso incorrecto del ejemplo de arriba).
- Al obtenerse la cadena justificada las cantidades de espacios de cada tramo podrán diferenciarse a lo sumo en 1, o sea, nunca podrá haber un tramo de espacios cuya diferencia con la longitud de otro tramo sea mayor que 1 (1er caso incorrecto del ejemplo de arriba, en el que hay un tramo de 4 espacios y otro de 2 espacios). Es decir, los espacios a insertar deben ser distribuidos lo más uniformemente posible.

Nota:

Observe que las longitudes de las líneas originales (el string en cada elemento del array) pueden ser diferentes, pero la longitud de cada línea justificada (el string en cada elemento del array resultado) será de la misma, ya que es el resultado de insertar espacios hasta alcanzar en todas el ancho indicado.

Se garantiza que cada cadena del array pasado al método:

1. No será null
2. Contendrá al menos un espacio
3. No contendrá secuencias de más de 1 espacio consecutivos
4. No comenzará ni terminará en espacio
5. No contendrá caracteres de cambio de línea ('\\n')

Por ejemplo, en el array original no aparecerán cadenas de la forma

	E	n		u	n		l	u	g	a	r		d	e		L	a		M	a	n	c	h	a
--	---	---	--	---	---	--	---	---	---	---	---	--	---	---	--	---	---	--	---	---	---	---	---	---

Comienza en espacio

E	n		u	n			l	u	g	a	r		d	e		L	a		M	a	n	c	h	a
---	---	--	---	---	--	--	---	---	---	---	---	--	---	---	--	---	---	--	---	---	---	---	---	---

Dos espacios consecutivos