Facultad de Matemática y Computación, Universidad de la Habana
Segundo Problema de Programación, curso 2009-2010

Array reducidos

Llamaremos **array reducido** a aquel array de valores enteros que no contenga más de dos elementos repetidos de forma consecutiva. La siguiente figura muestra un array reducido:

1	1	2	5	8	4	4	5	5	1

El siguiente ejemplo es un **array no reducido** que es aquel en el que un valor aparece 3 o más veces consecutivas:

2	3	3	3	8	5	0	0	0	0

Cualquier array puede *reducirse* si seguimos las siguientes reglas:

1. Solo aquellos bloques de 3 o más repeticiones consecutivas de un elemento pueden ser eliminados para formar un nuevo array, pero al hacerlo debe eliminarse el bloque completo y no se tienen en cuenta repeticiones del mismo elemento fuera de este bloque.
2. Si producto de la eliminación de un bloque, en el nuevo array aparece un bloque de 3 o más repeticiones consecutivas de un elemento entonces este bloque debe eliminarse y debe continuarse hasta que no se forme ningún bloque con las características mencionadas tras la eliminación.
3. Solo se puede eliminar un bloque a la vez
4. El proceso de eliminación termina cuando se haya llegado a un array reducido.

Por ejemplo, si tomamos el siguiente array:

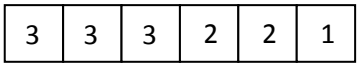
2	2	2	3	4	4	4	3	3	2	2	1

En un primer paso puede eliminarse el bloque de tres 2 consecutivos o el bloque de tres 4 consecutivos.

Si eliminamos el bloque de los tres 2 nos quedaría:

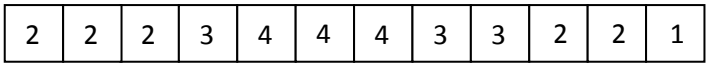
3	4	4	4	3	3	2	2	1

Si ahora eliminamos el bloque de los tres 4 obtenemos:



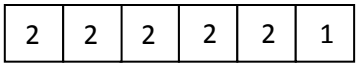
Note que ahora nos queda un nuevo bloque de 3 consecutivos que podemos eliminar obteniendo finalmente:

Si inicialmente hubiésemos comenzado eliminando el bloque de los 4 consecutivos entonces podrían haberse seguido los siguientes pasos:



Primero el de 4:

Luego el de 3:



Finalmente el de 2:

Lo que termina en un array de un elemento en lugar de un array de 3 elementos como en la variante anterior.

El problema a programar debe encontrar, dado un array de enteros cualquiera, la longitud del menor array reducido que se puede obtener haciendo reducciones a partir del original. Note que si el array original ya está reducido entonces la respuesta es la longitud de dicho array.

Para ello cree un ensamblado (proyecto *Class Library*) **`Reduccion.dll`** en la que se implemente una clase y un método como se muestra a continuación:

```
namespace Reduccion
{
  public class Reductor
    {
        public static int LongitudMenorReducido(int[] original)
        {
            ...
        }
    }
}
```

Nota: Considere que el parámetro que se pasa a **`original`** no es **`null`**