Grupo Weboo
Universidad de La Habana

Primer Examen de Programación
Curso 2017-2018

Inviertebot

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.

Inviertebot es un robot programable que es empleado en concursos de participación donde se ofrecen regalos a los concursantes ganadores de las distintas secciones. Al inicio de cada concurso se reunen los N regalos disponibles uno al lado del otro, de forma tal que el moderador los va tomando de izquierda a derecha durante cada sección. Para hacer más imparcial las premiaciones, dado que los regalos pueden ser de diferente valor, se utiliza *Inviertebot* para reorganizar la secuencia de regalos de forma tal que la disposición final depende de su algoritmo (Figura 1).

Figura 1: Inviertebot y los N regalos dispuestos uno al lado del otro (N=6)

Para poner en funcionamiento a *Inviertebot*, se enumera la secuencia inicial de regalos desde 1 a N y se disponen en orden creciente formando una circunferencia alrededor del robot. Esto se expresa en un array de longitud N donde inicialmente se han ubicado los valores de 1 a N. Interprete que el array se representa de modo circular en el orden de las manecillas del reloj (como se muestra en la Figura 2).

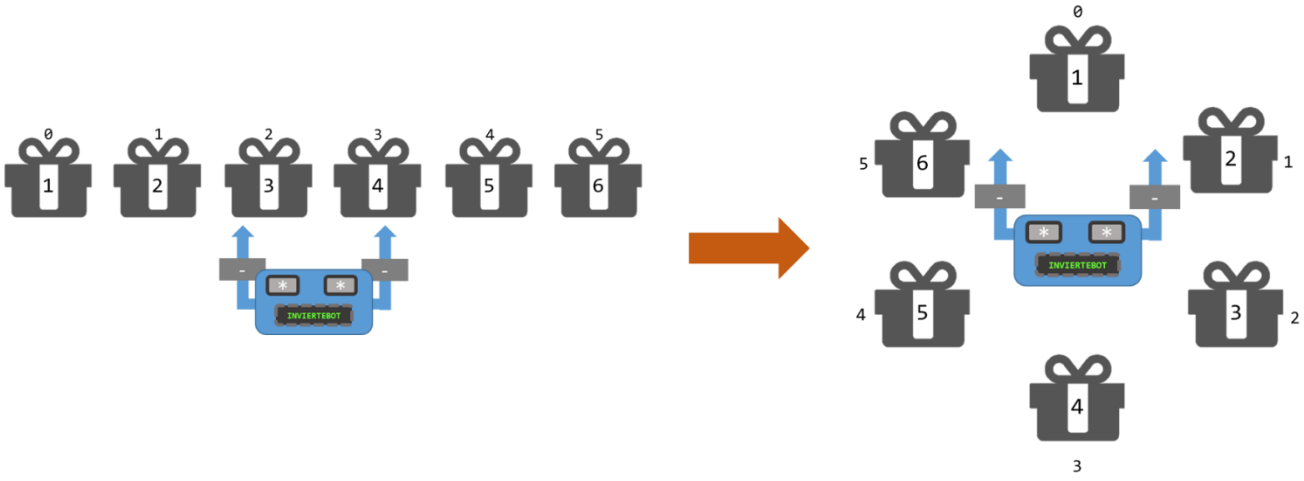
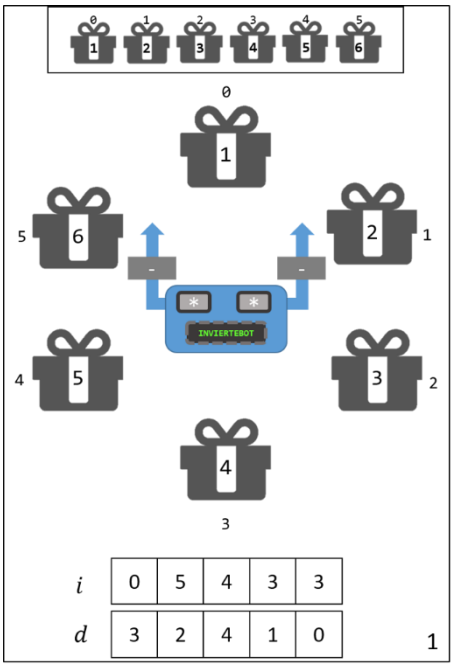


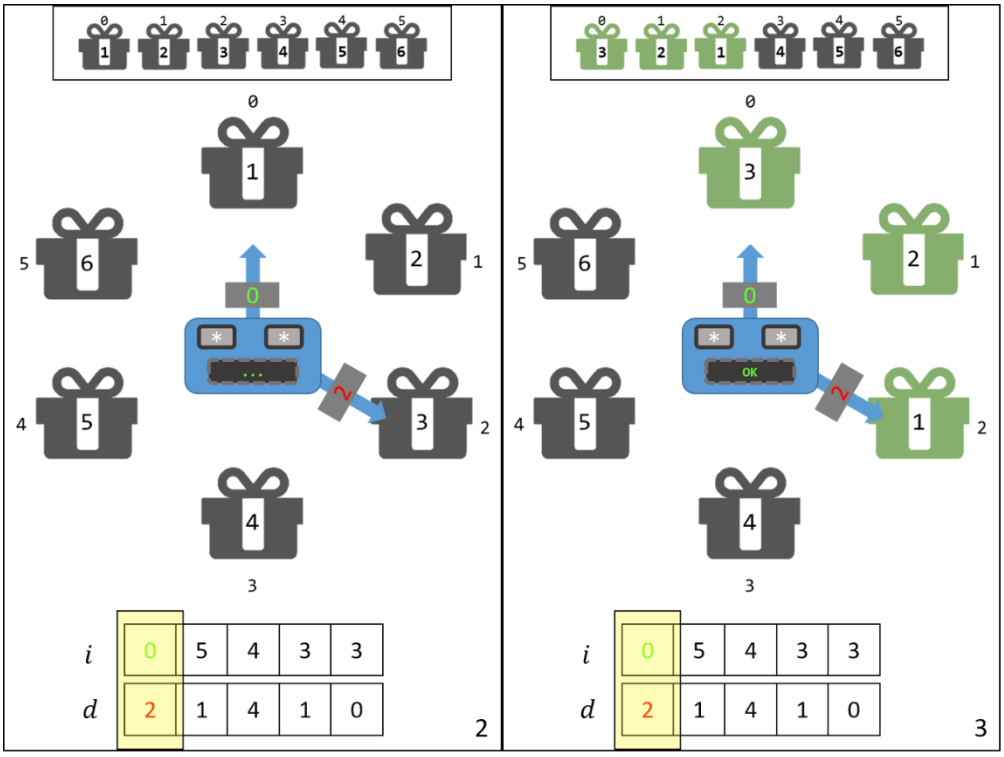
Figura 2: Preparando a Inviertebot para reorganizar los regalos

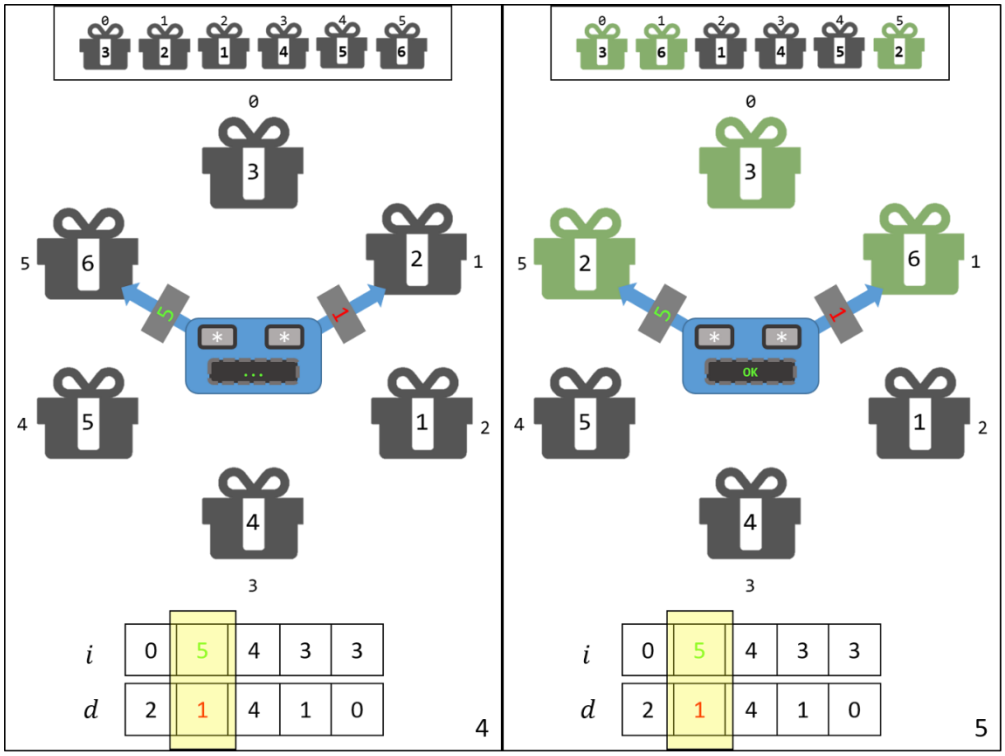
Al robot se le da una secuencia de instrucciones para que reorganice los regalos que expresaremos en dos *arrays*. Cada instrucción consiste en un par de números enteros i y d ($0 \leq i < \mathrm{N}$ y $0 \leq d < \mathrm{N}$) con los cuales *Inviertebot* ajusta sus brazos a dichas posiciones en la secuencia de regalos y procede a invertir el orden de todos los regalos en el intervalo $[i, d]$. Es decir, el robot sitúa su brazo izquierdo en la posición i y el derecho en la posición d e invierte el orden de todos los regalos que se encuentran entre el brazo izquierdo y el derecho (**en ese orden**) en la secuencia circular.

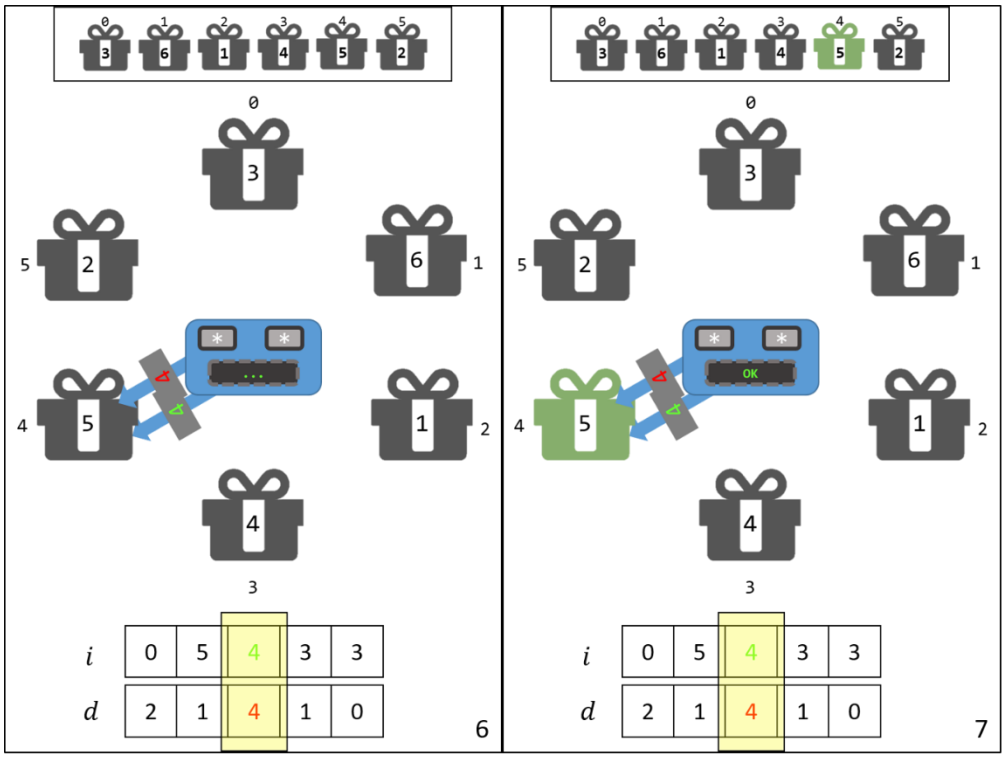
Si a la configuración anterior se le dan las siguientes intrucciones:

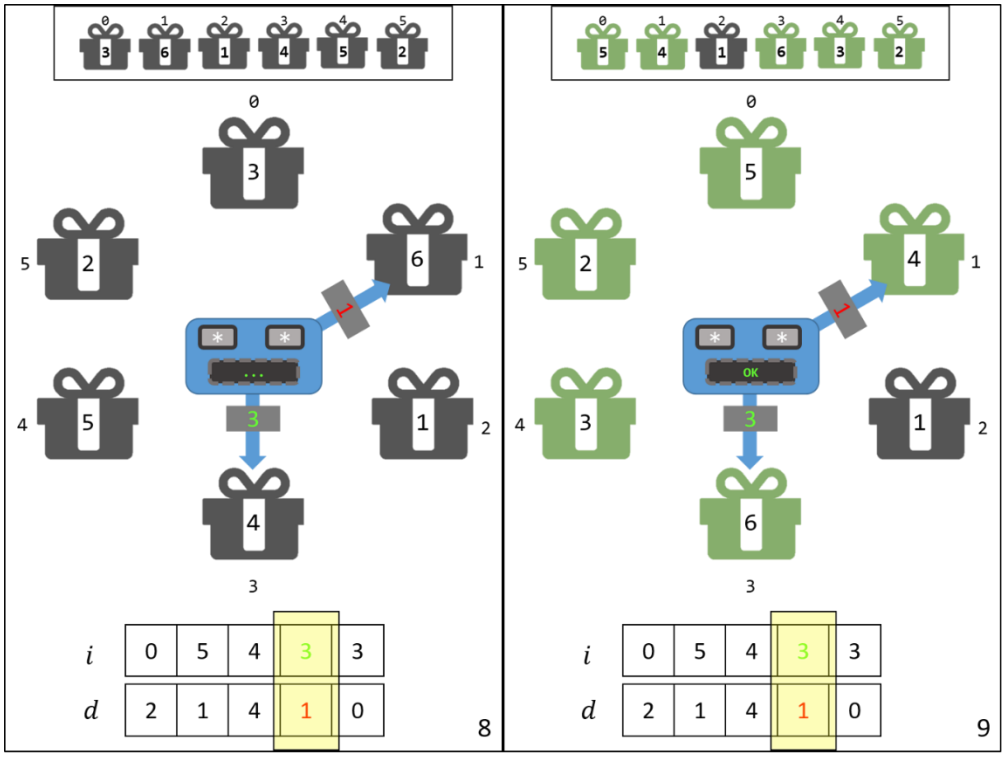


Entonces el robot procedería de la forma siguiente:



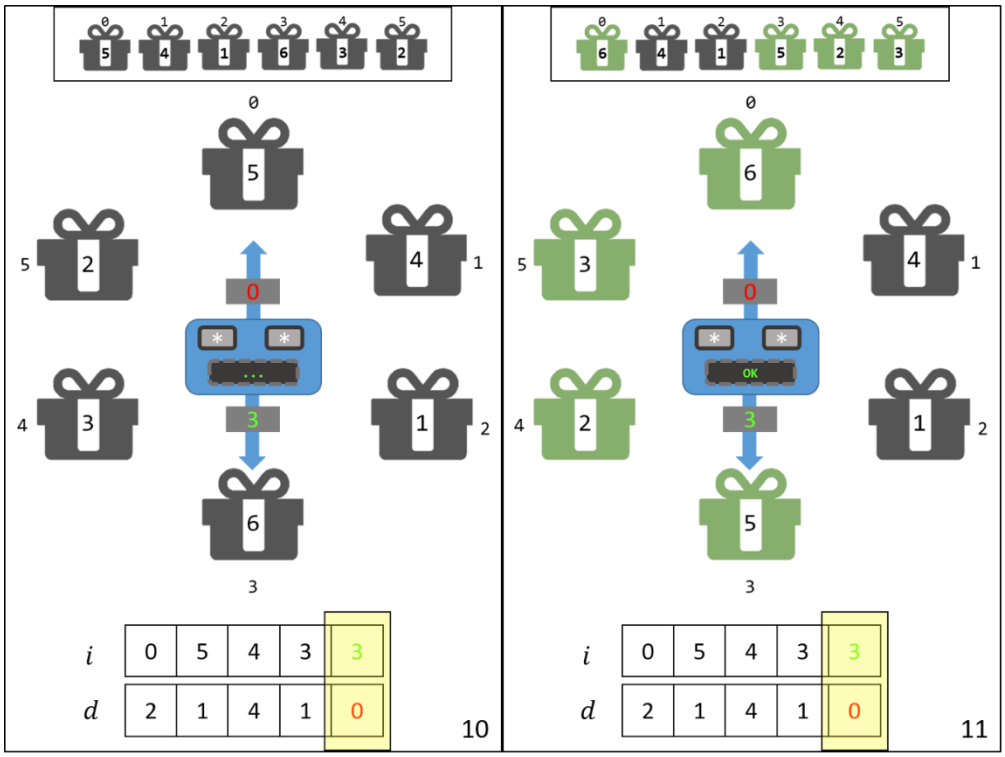

0 1 2 3 4 5
3 2 1 4 5 6
0
3
5 6
2 1
5
1
4 5
1 2
4
3
i 0 5 4 3 3
d 2 1 4 1 0
4
0 1 2 3 4 5
3 6 1 4 5 2
0
3
5 2
6 1
5
1
4 5
1 2
4
3
i 0 5 4 3 3
d 2 1 4 1 0
OK
5


0 1 2 3 4 5
3 6 1 4 5 2
0
3
5 2
6 1
4 5
1 2
4
3
i 0 5 4 3 3
d 2 1 4 1 0
6
0 1 2 3 4 5
3 6 1 4 5 2
0
3
5 2
6 1
4 5
1 2
4
3
i 0 5 4 3 3
d 2 1 4 1 0
OK
7



i	0	5	4	3	3
d	2	1	4	1	0

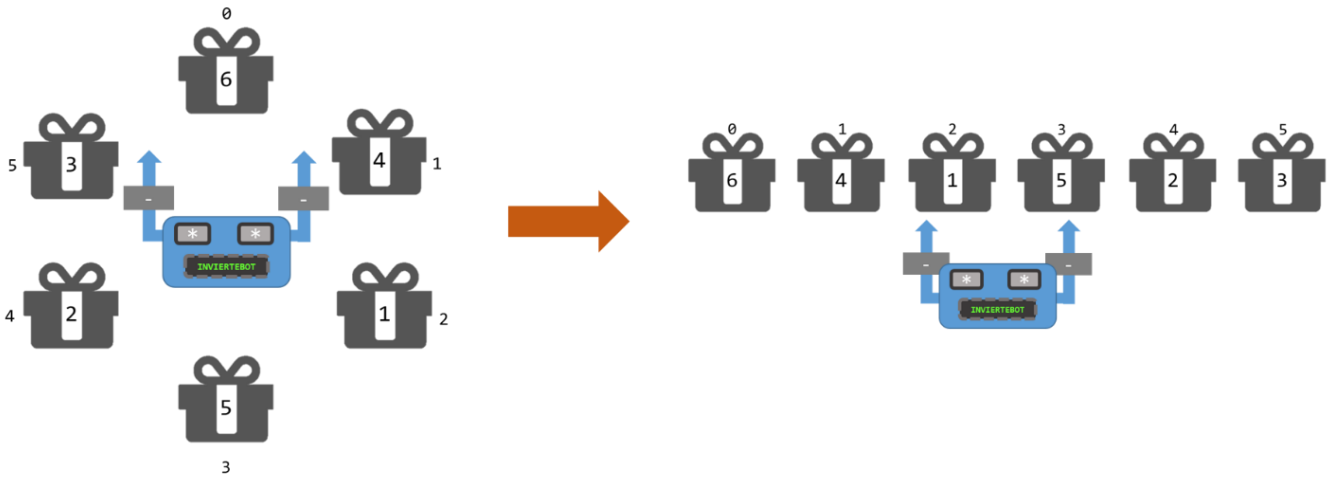
i	0	5	4	3	3
d	2	1	4	1	0



i	0	5	4	3	3
d	2	1	4	1	0

i	0	5	4	3	3
d	2	1	4	1	0

Quedando finalmente los regalos ordenados de la siguiente forma:



Como parte del equipo de *Inviertebot*, tu tarea consiste en implementar el algoritmo descrito anteriormente con el objetivo de comprobar el correcto funcionamiento del robot.

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 `EjecutaInstrucciones` que se encuentra en la clase `Inviertebot` en el *namespace* `Weboo.Examen`. En la biblioteca de clases encontrará la siguiente definición:

```
namespace Weboo.Examen
{
    public class Inviertebot
    {
        public static int[] EjecutaInstrucciones(int n, int[] i, int[] d)
        {
            //Borre la siguiente línea y escriba su código
            throw new NotImplementedException();
        }
    }
}
```

Este método tiene como primer parámetro un `int` que representa la cantidad de regalos que *Inviertebot* debe tener en cuenta. Recuerde que inicialmente los regalos están numerados desde **1 hasta n** y dispuestos **uno a continuación del otro** en orden **creciente** alrededor de *Inviertebot* siguiendo el sentido de las manecillas del reloj. Como segundo y tercer argumentos este método tiene dos `int[]` los cuales representan las instrucciones para el ajuste del brazo izquierdo y derecho de *Inviertebot* respectivamente. Note entonces que la instrucción k está representada por los valores `i[k]` y `d[k]` para el ajuste de los brazos. Usted debe implementar este método de forma tal que se devuelva finalmente un array `int[]` con la disposición final de los regalos.

Puede asumir que:

- El parámetro n **siempre** será un entero positivo.

- Los *arrays* `i` y `d` **siempre** tendrán igual longitud y tendrán al menos un elemento. **Todos** los valores en `i` y `d` indican posiciones correctas, es decir, cumplen que `0 <= i[k] < n` y `0 <= d[k] < n`

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 `EjecutaInstrucciones`, ni su orden. Por supuesto, usted puede (y debe) adicionar todo el código que necesite.

Ejemplos

```
//EJEMPLO 1
int n1 = 6;
int[] i1 = { 0, 5, 4, 3, 3 };
int[] d1 = { 2, 1, 4, 1, 0 };
int[] regalos1 = Inviertebot.EjecutaInstrucciones(n1, i1, d1);
//regalos1 = { 6, 4, 1, 5, 2, 3 }

//EJEMPLO 2
int n2 = 10;
int[] i2 = { 0, 0, 3 };
int[] d2 = { 5, 5, 3 };
int[] regalos2 = Inviertebot.EjecutaInstrucciones(n2, i2, d2);
//regalos2 = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }

//EJEMPLO 3
int n3 = 8;
int[] i3 = { 4, 1, 6, 3, 7 };
int[] d3 = { 5, 7, 1, 2, 1 };
int[] regalos3 = Inviertebot.EjecutaInstrucciones(n3, i3, d3);
//regalos3 = { 4, 8, 5, 7, 3, 2, 1, 6 }

//EJEMPLO 4
int n4 = 2;
int[] i4 = { 1, 0, 0, 1, 1 };
int[] d4 = { 0, 1, 1, 0, 0 };
int[] regalos4 = Inviertebot.EjecutaInstrucciones(n4, i4, d4);
//regalos4 = { 2, 1 }

//EJEMPLO 5
int n5 = 6;
int[] i5 = { 2, 3, 3, 4, 5 };
int[] d5 = { 3, 5, 4, 5, 1 };
int[] regalos5 = Inviertebot.EjecutaInstrucciones(n5, i5, d5);
//regalos5 = { 1, 6, 4, 5, 3, 2 }
```

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.