Recogida y entrega de cajas de refresco

Cada mañana los clientes de una empresa distribuidora de refrescos embotellados le anuncian la cantidad que cajas con botellas llenas que necesitan y la cantidad de cajas de botellas vacías que desean entregar para reciclar.

Con la lista de los clientes y sus demandas en mano, el chofer del único camión de la empresa debe partir del almacén cargado de todas las cajas solicitadas, visitar una sola vez a cada cliente y regresar al almacén con las cajas de botellas vacías.

Pero sucede que muchas veces no se puede cumplir con la demanda de todos los clientes pues el camión tiene capacidad reducida.

Por ejemplo si el camión tuviese capacidad máxima de 6 cajas y se tuviese el siguiente listado de clientes y demandas:

Cliente	Cajas de botellas llenas a entregar	Cajas de botellas vacías a recoger
0	1	3
1	3	1
2	2	1
3	2	2

Ejemplo 1

no se podría atender en un viaje a todos los clientes relacionados porque están pidiendo más cajas llenas que con las que puede salir, sin embargo, si se deja fuera al cliente 3, entonces se puede atender al resto.

Si el camión tiene una capacidad de 3 cajas y se tiene la siguiente configuración de clientes y demandas:

Cliente	Cajas de botellas llenas a entregar	Cajas de botellas vacías a recoger
0	1	4
1	5	1
2	6	1
3	2	8

Ejemplo 2

entonces no es posible atender a ninguno. También el chofer se ha dado cuenta de que no puede visitar a los clientes en cualquier orden, pues le ha pasado que al llegar a uno y hacer la entrega, no puede a su vez recoger todo el vacío de éste porque sobrepasaría la capacidad del camión. Por ejemplo para el Ejemplo 1, con el camión de 6 cajas de capacidad, la solución nunca podría ser visitando de primero al cliente 0 pues a este se le entrega 1 caja llena pero habría que recoger 3 lo que más las 5 llenas que aún le quedan sobrepasaría la cantidad de cajas. Pero si se sigue la ruta de clientes `1→2→0` no habría problemas.

El problema consiste en obtener una ruta en la que se visite la mayor cantidad de clientes posible y a su vez no sobrepasar nunca la capacidad del camión. Para ello debe implementar el método `EncuentraRuta` que se muestra continuación:

```
public class Enrutador
    {
        public static int[] EncuentraRuta(int capacidad, Cliente[] listado)
        {
            ...
        }
    }
```

Donde:

`capacidad` representa la capacidad del camión, es decir la cantidad de cajas que puede cargar y `listado` la relación de clientes a atender, puede asumir que este parámetro nunca será `null`, así como ninguno de los valores que contiene y su longitud será siempre mayor que 0. La clase `Enrutador` debe formar parte del ensamblado Ruteo.dll.

La clase `Cliente` (dentro de Cliente.dll) tiene una propiedad `CajasBotellasLlenas` que indica la cantidad de cajas que solicita cada cliente y una propiedad `CajasBotellasVacias` que indica la cantidad de cajas de botellas vacías que cada uno de ellos va a entregar.

```
namespace Weboo.Ruteo
{

    public class Cliente
    {
        public int CajasBotellasLlenas { get;}
        public int CajasBotellasVacias { get;}
    }
}
```

El método debe retornar un array de enteros que en la primera posición tenga el índice del primer cliente a atender, en la segunda el próximo y así sucesivamente. Si no es posible atender a ningún cliente, debe devolverse un array de longitud cero. Para el ejemplo **NOTE** que un cliente no necesariamente entrega y solicita la misma cantidad de cajas y que en general el total de las cajas a entregar no tiene que ser igual al total de las cajas a recoger, **NOTE** además que el camión sale cargado con la cantidad total de cajas de botellas llenas que solicitan los clientes, por lo que no necesariamente sale con la cantidad tope de cajas que permite su capacidad.