

Candados

En cierta ocasión el nieto de un cerrajero se quedó solo con una caja de candados. Para cuando el viejo regresó el pequeño había enlazado los candados unos con otros. Usted deberá implementar un programa que permita determinar el número mínimo de llaves que necesita para abrir el menor número de candados y que con ello todos los candados vuelvan a quedar separados.



Usted deberá implementar una biblioteca de clases `Candados.dll` con la clase `Cerrajero` dentro del **namespace** `Candados` y el siguiente método:

```
namespace Candados {  
    public class Cerrajero {  
        public static int[] LlavesParaAbrir (int n, Enlace[] enlaces) {  
            ...  
        }  
    }  
}
```

El parámetro `n` representa la cantidad de candados de la caja los cuales están enumerados con los números del 0 al `n-1`. Cada llave se enumera igual que el candado correspondiente. El tipo **struct** `Enlace` se le provee en la biblioteca `Candados.Util` y tiene dos propiedades: `Candado1` y `Candado2`. Esto expresa que los candados con índice `Candado1` y `Candado2` están enlazados.

Su método deberá devolver un **array** con los índices de un conjunto mínimo de llaves que permite separar todos los candados¹. De haber más de una solución con igual cantidad mínima de llaves (candados a abrir) puede dar como respuesta cualquiera de ellas.

NOTE que tiene que existir una solución pues en el caso peor sería abrir todos los candados.

NOTE que al abrir un candado se rompen los enlaces de éste con los demás a los que estaba enlazado. Un candado queda libre cuando no está enlazado con ningún otro.

El ejemplo de la figura se debe poder resolver haciendo la llamada:

```
Cerrajero.LlavesParaAbrir (6, new Enlace [] {new Enlace (0,1), new Enlace (0,2), new Enlace (1,2), new Enlace (1,4), new Enlace (3,1), new Enlace (2,5) });
```

La solución mínima es {1, 2} (liberar los candados 1 y 2).

¹ Este problema pertenece a la familia de problemas conocidos como NP. Si cree tener una solución iterativa del mismo, espere a terminar la prueba y la consulta con su profesor, mientras tanto le aconsejamos desarrolle una solución recursiva.