

Examen Final de Programación Curso 2005-2006

Pregunta 2

Encontrar el ciclo ganador

Para cambiar de un tipo de moneda a otro se aplica una tasa de cambio. Por ejemplo para cambiar de euros a dólares USA se aplica una tasa de cambio de 1.25, es decir que por 100 euros obtenemos 125 dólares USA. No necesariamente la tasa de cambio en el sentido inverso es el inverso de la tasa de cambio original. Así por ejemplo la tasa de dólares USA a euros podría ser 0.9 y no 0.8 (que es el inverso de 1.25)

Si tenemos varios tipos de moneda llamamos un ciclo de cambios a una secuencia de cambios que llevan de un tipo de moneda X pasando por varios tipos de moneda (no necesariamente por todos los tipos de moneda y pasando sólo una vez por cada tipo de moneda) hasta llegar al tipo X nuevamente. Se dice que un ciclo de cambios da ganancia si al final obtenemos más dinero que el original aplicando dicho ciclo de cambios.

Si por ejemplo tenemos los tipos de monedas A, B, C, D

- Y las tasas de cambios
 - B→A = 1.25
 - A→C = 0.9
 - A→B = 0.5
 - C→B = 1.50

El ciclo B, A, B no da ganancia, ya que si cambiamos 1 B por A y luego de nuevo por B tenemos

$$1 * 1.25 * 0.5 = 0.625 \text{ que es menos que el 1 original}$$

Sin embargo el ciclo de cambio B, A, C, B sí da ganancia, ya que si tenemos 1 unidad de la moneda B, entonces:

$$1 * 1.25 * 0.9 * 1.50 = 1.6875 \text{ que es más que el 1 original.}$$

Suponga que tenemos n tipos de moneda y que una tabla de $n \times n$ nos da todas las tasas de cambio existentes entre ellas. Es decir, que el valor de la celda $[i, j]$ nos da la tasa de cambio de la moneda i a la moneda j . Las celdas de la forma $[i, i]$ tendrán siempre valor 1.

En la librería de clases `Weboo.ExamenFinal.dll` queremos definir un método `MáximaGanancia` que a partir de un array rectangular que nos representa una tabla con las tasas de cambio y un tipo de moneda i nos de como respuesta el ciclo de cambios que da mayor ganancia para cambiar la moneda i y volver a llegar a la moneda i .

```

namespace Weboo.ExamenFinal
{
    public class Pregunta2
    {
        public static System.Collections.ArrayList
            MaximaGanancia(double[,] tasas, int monedaBase)
        {.....}
    }
}

```

El método `MáximaGanancia` recibe como parámetros un array rectangular de `double` `tasas` y un `int` `monedaBase`. Si el valor de `tasas` fuera `null` o si la longitud de ambas dimensiones fuera 0, o si el array `tasas` no fuera cuadrado, el método `MáximaGanancia` debe lanzar `ArgumentException`. El parámetro `monedaBase` indica la moneda de la que se parte para encontrar el ciclo de máxima ganancia.

Si el valor de `monedaBase` no está en el rango adecuado según la tabla `tasas` entonces se debe lanzar `ArgumentOutOfRangeException`.

El `ArrayList` devuelto por el método debe contener los números de tipos de monedas que participan en el ciclo de máxima ganancia, en el orden en que se realizan los cambios (note que entonces el valor `monedaBase` debe estar tanto al principio como al final del `ArrayList` que se devuelve).

Si hubiera varios ciclos con el que se obtiene igual máxima ganancia, entonces el método `MáximaGanancia` debe devolver un `ArrayList` que contenga a uno cualquiera de estos ciclos.

Si no hubiera ningún ciclo con el que se obtenga ganancia, entonces el método `MáximaGanancia` debe devolver un `ArrayList` de longitud 0.

Ejemplo:

A continuación se muestra una tabla de tasas de cambio para las monedas 0, 1, 2 y 3.

	0	1	2	3
0	1.00	0.50	0.90	10.25
1	1.25	1.00	0.90	17.00
2	1.03	1.50	1.00	15.10
3	0.01	0.04	0.05	1.00

Si al llamar al método `MáximaGanancia` el parámetro `tasas` tiene como valor un array que representa a la tabla anterior y el parámetro `monedaBase` tiene como valor 1, entonces los ciclos con los que se obtiene ganancia son:

- a) 1, 2, 1 ya que $1 * 0.90 * 1.50 = 1.35$ es mayor que 1
- b) 1, 0, 2, 1 ya que $1 * 1.25 * 0.90 * 1.50 = 1.6875$ es mayor que 1
- c) 1, 3, 2, 1 ya que $1 * 17.00 * 0.05 * 1.50 = 1.275$ es mayor que 1

De todos los anteriores el ciclo b) es el de mayor ganancia y por lo tanto, el `ArrayList` devuelto por el método `MáximaGanancia` debe ser `{1, 0, 2, 1}`.