

## Cambios de monedas

Se dispone de la tabla de tasas de cambio (un array de tipo `double[,]`) entre un conjunto de monedas y se quiere determinar si existe alguna forma de partir de una cantidad inicial en una determinada moneda y mediante cambios sucesivos llegar a obtener una mayor cantidad en la moneda original.

En general, la casilla de la fila I y columna J indica la tasa de cambio de la moneda I a la J, o sea, la cantidad de moneda J que el banco entrega al recibir una unidad de la moneda I.

	CUC	CUP	USD	EUR
CUC	1	24	0.1818	0.8182
CUP	0.04	1	0.0556	0.03984
USD	0.7692	17	1	0.7143
EUR	1.1	25	1.3	1

Veamos el resultado de cambiar EUR por USD y luego los USD obtenidos por EUR nuevamente.

1 EUR -> 1.3 USD -> 0.9286 EUR

0.9286 EUR < 1 EUR

O sea, que en este caso (que es lo común en la realidad) se terminó obteniendo menos que lo que se tenía inicialmente. Sin embargo, observe esta otra secuencia de cambios, la cual sí dejaría ganancias.

1 CUC -> 24 CUP -> 1.3333 USD -> 0.9524 EUR -> 1.0476 CUC

1.0476 CUC > 1 CUC

Dentro del proyecto `CambiosMonedas` que se le entrega junto a este enunciado, implemente el método `int[] CambioConGanancias(double[,] tasasCambio)`. El array `tasasCambio` contendrá las tasas de cambio de la siguiente manera: `tasasCambio[i, j]` = cantidad de la moneda `j` que devuelve el banco al recibir una unidad de la moneda `i`. De acuerdo con el ejemplo anterior, al ser la primera moneda el CUC y la segunda el CUP, `tasasCambio[0, 1] = 24` y `tasasCambio[1, 0] = 0.04`. De haber alguna secuencia de cambios que deje ganancias, su método deberá devolver un array con los índices de las monedas de la secuencia.

Para la tabla que hemos usado en los ejemplos de arriba un posible resultado es el array `{0, 1, 2, 3, 0}`, que expresa que una secuencia de cambios de la monedas `0` (CUC) a la `1` (CUP), a la `2` (USD), a la `3` (EUR), y luego a la `0` (CUC) de nuevo nos da un resultado ganador.

De no existir secuencias de cambios con ganancias, el método debe devolver el valor `null`.

## Aclaraciones

- Se garantiza (su implementación no lo tiene que verificar) que el parámetro pasado a su método no será `null`, y que corresponderá a una tabla que tiene la misma cantidad de “filas” que de “columnas” y todos sus valores serán estrictamente positivos. La diagonal contendrá valores 1.
- La solución de Visual Studio que se le suministra contiene dos proyectos: `CambiosMonedas` y `Pruebas`. El único que se evaluará es `CambiosMonedas`. El otro puede ser utilizado por Ud. para ayudarle a probar su algoritmo.
- Note que puede haber más de una solución válida. En tal caso Ud. puede devolver cualquiera. De acuerdo con la tabla del primer ejemplo, la secuencia `{2, 0, 1, 2}` también es válida.

**Le recomendamos tratar de compilar su proyecto lo más frecuentemente posible desde que empiece a trabajar. Entre otros beneficios, esto disminuirá la probabilidad de que pierda su trabajo si ocurre algún fallo eléctrico.**