Partición balanceada

Se define *Partición de un grupo de elementos* E como la división de dicho grupo en subgrupos:

E_1, E_2,..., E_n, con 1<=n<=| E |, (cantidad de elementos de E) que cumplen:

- Los subgrupos son disjuntos, es decir, no tienen elementos comunes y cada elemento del grupo original pertenecerá a uno y solo uno de los subgrupos E_i.
- Los subgrupos no contendrán elementos que no sean del grupo original.

Ejemplos de particiones para un grupo de enteros como `{1, 1, 2, 3, 4, 5}` son:

- `{ {1}, {1, 2}, {3, 5}, {4}}` donde los subgrupos son E_1=`{1}`, E_2=`{1,2}`, E_3=`{3,5}`, E_4=`{4}`,
- `{ {1, 1, 3}, {4, 5, 2} }` donde los subgrupos son E_1=`{1, 1, 3}`, E_2=`{4, 5, 2}`
- el propio grupo original `{ {1, 1, 2, 3, 4, 5} }` es una partición.

Nótese que el orden de los elementos en un subgrupo no importa, por ejemplo la segunda partición de arriba es equivalente a la partición `{ {1, 3, 1}, {4, 2, 5} }` .

Una partición se dice "*balanceada*" cuando todos sus subgrupos suman lo mismo.

Por ejemplo, para el grupo de enteros positivos E = `{1, 7, 2, 4, 3, 6, 1}` *particiones "balanceadas"* válidas serían:

- `{ {1, 3, 4}, {2, 6}, {1, 7} }`, ya que: `1+3+4 = 8, 2+6 = 8, 1+7 = 8,`
- `{ {1 ,3, 2, 6}, {1, 4, 7} }`, `1+3+2+6 = 12, 1+4+7 = 12,`
- `{ {1, 2, 3, 1, 4, 6, 7} }`, `1+2+3+1+4+6+7 = 24`, al ser un solo grupo cumple con el enunciado de que tiene la misma suma que los demás grupos de la partición.

Sin embargo, la partición `{ {1, 2, 3}, {1}, {4, 7, 6} }` no es "*balanceada*" ya que todos sus grupos no suman lo mismo, pues `1+2+3=6, 1=1, 4+7+6=17`,

Finalmente se define la *cardinalidad* de una partición como la cantidad de subgrupos en los que quedó particionado el grupo original, por ejemplo para las particiones balanceadas válidas vistas en el ejemplo anterior las cardinalidades serían 3, 2 y 1 respectivamente.

El problema a resolver consiste en dado un array de enteros positivos y un entero n estrictamente positivo, encontrar, si existe, una partición "balanceada" del mismo cuya cardinalidad sea n.

Dentro del proyecto `Particiona` que se le entrega junto a este enunciado, implemente el método:

```
public static int[] ParticionBalanceada ( int[] original, int n) {…}
```

El array original que se pasa como parámetro representará el grupo de enteros positivos que debe particionar (recuerde que puede tener valores repetidos) y el entero n representa la cantidad de subgrupos que debe tener la partición balanceada obtenida. El array de salida tendrá en cada posición un entero indicando el número del subgrupo donde se ubicó el elemento de dicha posición en el array original, de no existir una partición balanceada con la cardinalidad indicada devolverá null.

Veamos: Para el array de enteros visto anteriormente,

original	**1**	**7**	**2**	**4**	**3**	**6**	**1**

una partición "balanceada" de cardinalidad 3 será **`{ {1, 3, 4}, {2, 6}, {1, 7} },`**

donde **`{1, 3, 4}`** será el subgrupo **`0`**, **`{2, 6}`** el subgrupo **`1`** y **`{1, 7}`** el subgrupo **`2`**.

Entonces para este caso la salida del método `ParticionBalanceada` será el array de enteros:

salida	**0**	**2**	**1**	**0**	**0**	**1**	**2**

Aclaraciones

Se garantiza (su programa no lo tiene que verificar) que los parámetros que se pasarán a su método no tendrán el valor null y que el primero es un array con todos sus valores estrictamente positivos y el segundo es un entero positivo menor que la cantidad de elementos que tenga el array original.

Puede haber más de una solución válida, para el ejemplo visto otra posible solución es:

salida	**2**	**2**	**1**	**0**	**0**	**1**	**0**

en tal caso su método puede devolver cualquiera de ellas.

Le recomendamos tratar de compilar con cierta frecuencia cada vez que haga cambios. Entre otros beneficios, esto disminuirá la probabilidad de que pierda su trabajo si ocurre algún fallo eléctrico.