

Contar secuencias de múltiplos de longitud k

El problema

Dados dos números enteros, A y B , se dice que A es *múltiplo* de B cuando existe un tercer entero C tal que $A = B \times C$. Por ejemplo, 10 es múltiplo de 5 y 40 es múltiplo de 20. Por su parte, una *secuencia de múltiplos* es una sucesión de números enteros en las que todos los números, a excepción del primero, son múltiplos del anterior. Un ejemplo de este tipo de sucesiones sería 5 10 20 40 80 160.

Debe desarrollarse sistemáticamente, aplicando el método explicado en clase, un algoritmo iterativo eficiente que, dado un vector de enteros, que no contiene ceros, y un número natural $k > 0$, determine cuántas secuencias de múltiplos de longitud k tiene el vector.

Programa de prueba

Para realizar el control se proporciona un archivo `iter.cpp` que contiene un programa que lee desde la entrada estándar casos de prueba, los ejecuta, e imprime por la salida estándar el resultado. Cada caso de prueba consiste en dos líneas, la primera con el número de elementos del vector y el número k y la segunda con los elementos del vector en sí. Para cada caso de prueba el programa imprime el número de secuencias de múltiplos de longitud k que contiene el vector. La entrada finaliza con una línea con -1. A continuación se muestra un ejemplo de entrada/salida:

Entrada	Salida
8 4	4
9 5 10 20 40 80 160 320	2
4 2	4
2 4 6 12	
4 1	
2 4 6 12	
-1	

- En el primer caso, el tamaño del vector es **8** y $k=4$. Las 4 secuencias de múltiplos de longitud 4 contadas son: (i) 5 10 20 40; (ii) 10 20 40 80; (iii) 20 40 80 160; y (iv) 40 80 160 320.
- En el segundo caso, el tamaño del vector es **4** y $k=2$. Las 2 secuencias de múltiplos de longitud 2 contadas son: (i) 2 4; y (ii) 6 12
- En el tercer caso, el tamaño del vector es **4** y $k=1$. El resultado es 4, porque cualquier entero v es, a su vez, una secuencia de múltiplos de longitud 1.

Trabajo a desarrollar

Tu trabajo consiste en:

- Especificar el algoritmo, rellenando los huecos correspondientes, entre comentarios, que rodean la declaración de `num_sec_multiplos`
- Diseñar sistemáticamente un algoritmo eficiente que resuelva el problema, siguiendo los pasos indicados en la plantilla “DISEÑO DEL ALGORITMO”.
- Codificar el algoritmo como cuerpo de la función `num_sec_multiplos`
- Entregar `iter.cpp` a través del juez en línea de la asignatura.

Importante:

- **Sólo puntuarán aquellas entregas que superen los casos del juez (el resto de entregas puntuarán 0).**
- **Debes poner tu nombre y apellidos en el comentario habilitado para ello en la plantilla (aquellas entregas no identificadas con nombre y apellidos puntuarán 0)**
- No modificar el código proporcionado. Únicamente deben responderse a los distintos apartados, en el interior de los comentarios, e implementar la función `num_sec_multiplos` (sí pueden incluirse funciones auxiliares, si se considera necesario).