



Primer Examen Parcial. Marzo 2022.

Problema: En la clase MyList (implementación de una lista simple con head y tail, y con los métodos str, len y append o add-last), completa el método *removeMultiplesOf(e)*, que recibe un número entero e, busca la primera ocurrencia del elemento y elimina todos los múltiplos de e que ocurren después de la primera ocurrencia. El método no devuelve nada. El método modifica la lista. Si el elemento no existe en la lista entonces, el método muestra un mensaje informando que el elemento no existe.

A continuación, tienes algunos ejemplos:

lista antes de la operación	operación	lista después de la operación
6->3->2->6->6->4->1->3->9->2	<i>l.removeMultiplesOf(3)</i>	6->3->2->4->1->2
6->3->2->6->6->4->1->3->9->2	<i>l.removeMultiplesOf(1)</i>	6->3->2->6->6->4->1
6->3->2->6->6->4->1->3->9->2	<i>l.removeMultiplesOf(2)</i>	6->3->2->6->1->3->9
6->3->2->6->6->4->1->3->9->2	<i>l.removeMultiplesOf(6)</i>	6->3->2->4->1->3->9->2
6->3->2->6->6->4->1->3->9->2	<i>l.removeMultiplesOf(9)</i> (no tiene múltiplos después de su primera ocurrencia)	6->3->2->6->6->4->1->3->9->2
6->3->2->6->6->4->1->3->9->2	<i>l.removeMultiplesOf(5)</i> (no existe)	6->3->2->6->6->4->1->3->9->2
(lista vacía)	<i>l.removeMultiplesOf(x)</i>	(lista vacía)

No puedes añadir nuevos atributos o métodos a las clase SNode y MyList.

Para que la solución se considere correcta, la función propuesta debe ser **correcta** (resuelva el problema), **robusta** (no tenga errores y funcione para cualquier entrada) y **eficiente** en términos de complejidad temporal y espacial (evitar el uso de estructuras auxiliares). Además, el código debe ser fácil de entender y mantener. No está permitido el uso estructuras de Python como los diccionarios o las listas. No está permitido añadir otros métodos auxiliares a la clase MyList.