**Minion Task Scheduling**

**Description**

Commander Lambda's minions are upset! They're given the worst jobs on the whole space station, and some of them are starting to complain that even those worst jobs are being allocated unfairly. If you can fix this problem, it'll prove your chops to Commander Lambda so you can get promoted!

Minion's tasks are assigned by putting their ID numbers into a list, one time for each day they'll work on that task. As shifts are planned well in advance, the lists for each task will contain up to 99 integers. When a minion is scheduled for the same task too many times, they'll complain about it until they're taken off the task completely. Some tasks are worse than others, so the number of scheduled assignments before a minion will refuse to do a task varies depending on the task. You figure you can speed things up by automating the removal of the minions who have been assigned a task too many times before they even get a chance to start complaining.

Write a function called answer(data, n) that takes in a list of less than 100 integers and a number n, and returns that same list but with all of the numbers that occur more than n times removed entirely. The returned list should retain the same ordering as the original list - you don't want to mix up those carefully planned shift rotations! For instance, if data was [5, 10, 15, 10, 7] and n was 1, answer(data, n ) would return the list [5, 15, 7] because 10 occurs twice, and was thus removed from the list entirely.

**Languages**

To provide a Python solution, edit solution.py

To provide a Java solution, edit solution.java

**Test Cases**

**Test Case 1**

Inputs:

(int list) data = [1, 2, 3]

(int) n = 0

Output:

(int list) []

**Test Case 2**

Inputs:

(int list) data = [1, 2, 2, 3, 3, 3, 4, 5, 5]

(int) n = 1

Output:

(int list) [1, 4]

**Test Case 3**

Inputs:

(int list) data = [1, 2, 3]

(int) n = 6

Output:

(int list) [1, 2, 3]