



## Data Structures and Algorithms.

### PROBLEM 1 (1.5 points)

In this exercise, a linear abstract data type must be applied to detect and delete a pattern. Let's suppose that a singly linked list of numbers (only with the head reference) is available. The pattern is described as follows: given a number (for example 5, the next one must be its double value (10)).

You must implement a method that searches into the list for the described pattern and, if found, removes all the elements in the pattern but the first one. This method does not have parameters and returns the number of removed nodes.

The following example shows the result after executing the method.

- o Initial list (example): 4, 1, 2, 4, 8, 9, 3, 2, 5, 10, 20, 2, 1, 7
- o list content after first execution: 4, 1, 9, 3, 2, 5, 2, 1, 7
- o Method answer: 5 (that is, 5 removed nodes)

Identify and describe the worst case and compute the complexity (Big Oh!) of the implemented method. Do the same with the best case.

Note:

- It is not allowed to use any auxiliary variable of data structures such as Python list or other linear data structures to implement your solution.
- However, you can use any method of the singly linked list class.

### PROBLEM 2 (0.5 points)

Let's suppose that a doubly linked list class is available. Implement a **recursive** method in the class that reverse the list. For example, if the initial list is: 1,3,4,5, the list content after l.reverse(), should be: 5,4,3,1. It is not allowed to use variables of data structures such as Python list or other linear data structures to implement your solution. However, you can use any method of the doubly linked list class.