R1: Recursive programming

Write a recursive program (Python or C++ programming language) for the following requirements. You can use and extend for implementation the C++/Python model available in /Lab/R1, this model already containing recursive implementations for creating, printing and destroying of a **List.**

It is mandatory to work with a structure/class **List.**

For a **List** will be used a linked representation. Do not use containers from STL or predefined operations on lists in Python (append, len, slicing, etc.).

1. a. Transform a list in a set.
b. Determine the union of two sets. The sets are represented as lists.

2. a. Substitute the *i-th* element from a list, with a value *v*.
b. Determine difference of two sets represented as lists.

3. a. Check if a list is a set.
b. Determine the number of distinct elements from a list.

4. a. Determine if a list has even number of elements, without computing the length of the list.
b. Delete all occurrences of an element *e* from a list.

5. a. Determine the greatest common divisors of elements from a list.
b. Insert an element on the n-position in a list.

6. a. Add an element at the end of a list.
b. Concatenate two lists.

7. a. Test the equality of two lists.
b. Determine the intersection of two sets represented as lists.

8. a. Determine the lowest common multiple of the elements from a list.
b. Substitute in a list, all occurrence of a value *e* with a value *e1.*

9. a. Invert a list
b. Determine the maximum element of a numerical list.

10. a. Determine the number formed by adding all even elements and subtracting all odd numbers of the list.
b. Determine difference of two sets represented as lists.

11. a. Determine if a certain element is member in a list.
b. Determine the length of a list.

12. a. Test the inclusion of two lists
b. Insert in a list, after value *e*, a new value *e1.*

13. a. Test the inclusion of two sets, represented as lists.
b. Eliminate all occurrences of an element from a list.

14. a. Determine the last element of a list.
b. Delete elements from a list, from position n to n.

15. a. Substitute all occurrences of an element from a list with another list.
b. Determine the element from the *n-th* position in a list.