Exercise 06: Recursion

The following applies to all subtasks: do not use `for` or `while`. The tasks are only to be completed using recursion.

Method:

- Identify the base case (terminating case); when is the solution trivial?
- Identify the recursive case; when is the solution non-trivial, but based on another solution?
- What are the parameters for the method, and how must they be adapted in the recursive case?

Please note: since the signatures are already predefined, a helper method is often necessary to implement the actual recursion.

Task 1: Recursion for arrays

Implement the generic static methods `toString` and `contains` of the predefined class `Arrays`. Which parameters are useful for the helper function?

Please note: try to avoid using copies of the array.

Task 2: Recursion for lists

Implement the methods addRec and containsRec of the generic class `List`. The two iterative implementations (add and contains) are provided as help.

Please note: you can either create the helper methods in the List class, or you can extend the List.Element class.

Task 3: Recursion for binary trees

Implement the methods `addRec` and `containsRec` of the generic class `Tree`. Here too, the two iterative implementations (`add` and `contains`) are provided as help.

Additional task

Implement the toStringRec methods of the List and Tree classes recursively. The respective iterative toString methods are given.