Для решения поставленных задач был использован диалект языка Лисп – newLISP.

*Решение:*

**Задача 1 (Lab-2-1.lsp):**

Условие (null? lst) выполняется в нескольких случаях: когда изначально на вход подается пустой список или в ходе работы рекурсии он приходит пустым. Выполнение этого условия означает, что в исходном списке нет атомарного элемента под необходимом номером.

Условие ((and (= n 1) (atom? (first lst))) выполняется если голова текущего списка является атомарным элементом и n = 1 (значение n в ходе рекурсии уменьшается поэтому если n = 1, то это требуемый для решения атомарный элемент). Выполнение этого условия означает, что необходимый по заданию атомарный элемент был найден.

Условие (atom? (first lst)) выполняется если голова текущего списка является атомарным элементом, при этом функция вызывает сама себя уменьшив значение n на 1 и вместо целого списка передает его хвост.

Если же не выполняются 3 предыдущих условия, то функция вызывает сама себя не изменяя значение n и вместо целого списка передает его хвост.

Тестовые наборы данных и результаты приведены в Таблице 1:

Таблица 1. Тестовые набора данных

|  |  |
| --- | --- |
| Тестовые данные | Результат |
| ‘ ((2) (3) 4 5 a (e r) g) 3 | A |
| ‘() 2 | nil |
| ‘ ((2) (3) (e r)) 3 | nil |
| ‘(b) 1 | B |
| ‘ ((2) (3) 4 (e r) g) 1 | 4 |

**Задача 2 (Lab-2-2.lsp):**

Для начала происходит проверка на то, что количество элементов в списке меньше ли чем n, если же это условие верно, то возвращается весь список (в условие задания не указано что требуется делать в этом случае).

Далее рекурсивно составляется список (функцией CONS): берется голова исходного списка, после чего вызывается исходная функция при этом в качестве списка передается хвост исходного списка и n уменьшается на 1.

Тестовые наборы данных и результаты приведены в Таблице 2:

Таблица 2. Тестовые набора данных

|  |  |
| --- | --- |
| Тестовые данные | Результат |
| ‘() 3 | () |
| '(2 4) 3 | (2 4) |
| '(4 3 3 2) 3 | (4 3 3) |
| '(4 3 9 3 8 3 2) 4 | (4 3 9 3) |