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

**Задача №1 (Lab5-1-1.lsp):**

Текст представляется списком списков: каждое предложение – список слов, весь текст – список предложений.

В функции upper-case-first-letter используется отображающий функционал map. Он отображает аргументы-списки в новый список применением к одинаково расположенным элементам этих списков функции, представленной первым аргументом. Функция применятся в каждому подсписку исходного списка (предложению). В каждом предложение берется первый элемент (слово) и в нём делается заглавным первая буква, после чего выводятся остальные элементы (слова).

Тестовые наборы данных и результаты приведены в Таблице 1:

Таблица 1. Тестовые набора данных

|  |  |
| --- | --- |
| Тестовые данные | Результат |
| '(("дан" "текст") ("Выполнить" "представленное" "задание") ("представить" "отчет" "выполненной" "работы")) | (("Дан" "текст") ("Выполнить" "представленное" "задание")  ("Представить" "отчет" "выполненной" "работы")) |
| '(("лабораторная" "работа")) | (("Лабораторная" "работа")) |

**Задача №2 (Lab5-2.lsp):**

Была реализована программа, которая разбивает каждое слово фразы на слоги.

Правило слогораздела включает в себя следующие принципы:

1. Сочетание двух шумных или двух сонорных согласных отходит к последующему слогу.

2. Сочетание шумного и сонорного отходит к последующему слогу.

3. Сочетание сонорного и шумного имеет раздел посередине

Тестовые наборы данных и результаты приведены в Таблице 2:

Таблица 2. Тестовые набора данных

|  |  |
| --- | --- |
| Исходное выражение | Результат |
| '(("гости" "маска")) | ("го-сти" "ма-ска") |
| '(("окно" "храбрый")) | ("о-кно" "хра-брый") |
| '(("волна" "корма")) | ("во-лна" "ко-рма") |
| '(("карта" "банда")) | ("кар-та" "бан-да") |
| '(("тайфун" "тайна")) | ("тай-фун" "тай-на") |
| '(("простой" "добро" "парта" "иначе" "полотно")) | ("про-стой" "до-бро" "пар-та" "и-на-че" "по-ло-тно") |

**Задача №3 (Lab5-3.lsp):**

Каждое слово преобразуется в пару слов. Первое слово есть конкатенация первого слога ключего слова и части переводимого слова, оставшейся после отделения от него первого слога. Второе слово есть конкатенация первого слога переводимого слова и части ключевого слова, оставшейся после отделения от него первого слога.

Тестовые наборы данных и результаты приведены в Таблице 3:

Таблица 3. Тестовые набора данных

|  |  |
| --- | --- |
| Тестовые наборы данных | Результат |
| '(("сплетня")) '(("слово")) | ("сплево" "слотня") |
| '(("фляга")) '(("песня")) | ("флясня" "пега") |
| '(("тендер" "бригантина")) '(("волна")) | ("тенлна" "водер" "брилна" "вогантина") |
| '(("сбережение" "недруг" "птенец")) '(("зеркало")) | ("сбекало" "зеррежение" "некало" "зердруг" "птекало"  "зернец") |

**Задача №4 (Lab5-4.lsp):**

Текст представляется списком списков: каждое предложение – список слов, весь текст – список предложений.

Была разработана функция для шифрования по методу Юлия Цезаря для русского алфавита. Если в предложение появится слово из другого алфавита, то оно выводится без изменения.

В функции сaesar-cipher используется отображающий функционал map. Он отображает аргументы-списки в новый список применением к одинаково расположенным элементам этих списков функции, представленной первым аргументом. Функция применятся в каждому подсписку исходного списка (предложению). Каждый подсписок (предложение) передается в функцию cipher, которая в свою очередь передает каждый элемент подсписка (слово) в функцию cipher-word, в котором каждая буква русского алфавита сдвигается на одну вперед.

Тестовые наборы данных и результаты приведены в Таблице 4:

Таблица 4. Тестовые набора данных

|  |  |
| --- | --- |
| Тестовые наборы данных | Результат |
| '(("А" "Б" "В" "Э" "Ю" "Я")) | (("Б" "В" "Г" "Ю" "Я" "А")) |
| '(("а" "б" "в" "э" "ю" "я")) | (("б" "в" "г" "ю" "я" "а")) |
| '(("абракадабра")) | (("бвсблбебвсб")) |
| '(("Шифр" "Цезаря" "также" "известный" "как" "шифр" "сдвига") ("Cимвол" "в" "открытом" "тексте" "заменяется" "символом" "находящимся" "на" "некотором" "постоянном" "числе" "позиций" "левее" "или" "правее" "него" "в" "алфавите")) | (("Щйхс" "Чжибса" "ублзж" "йигжтуоьк" "лбл" "щйхс"  "тегйдб")  ("Cйнгпм" "г" "пулсьупн" "ужлтуж" "ибнжоажута" "тйнгпмпн"  "обцпеаъйнта" "об" "ожлпупспн" "рптупаоопн" "шйтмж"  "рпийчйк" "мжгжж" "ймй" "рсбгжж" "ождп" "г" "бмхбгйуж")) |
| '(("Диалект" "newLISP")) | (("Ейбмжлу" "newLISP")) |

**Задача №5 (Lab5-1-2.lsp):**

Текст представляется списком списков: каждое предложение – список слов, весь текст – список предложений.

В функции add-letters-in-text используется отображающий функционал map. Он отображает аргументы-списки в новый список применением к одинаково расположенным элементам этих списков функции, представленной первым аргументом. Функция применятся в каждому подсписку исходного списка (предложению). Каждый подсписок (предложение) передается в функцию choice-of-words, которая в свою очередь передает каждый элемент подсписка (слово) в функцию add-letters, в которой при нахождении в слове заданного 3-буквенного сочетания добавляется заданное 2-буквенное сочетание.

Тестовые наборы данных и результаты приведены в Таблице 5:

Таблица 5. Тестовые набора данных

|  |  |
| --- | --- |
| Тестовые данные | Результат |
| '(("абракадабра")) "абр" "XX" | (("абрXXакадабрXXа")) |
| '(("берег" "беречь" "оберег") ("роббер" "поберу" "бер")) "бер" "ХХ" | (("берХХег" "берХХечь" "оберХХег") ("робберХХ" "поберХХу"  "берХХ")) |