**Лабораторная работа №1**

Программа должна быть написана на языке ЛИСП.

Текст функции можно набирать в любом текстовом редакторе (например, в Блокноте). Файл следует сохранить с расширением lsp, лучше в той же папке, где находится ЛИСП. Запуск интерпретатора ЛИСПа и загрузки текста написанной функции можно осуществить 3 способами:

1. Наложите с помощью мышки файл с текстом функции на файл mulisp.com.
2. Запустите файл mulisp.com и после появления приглашения в виде $ введите в командную строку:

load 'имя файла

1. Скопируйте в буфер обмена текст функции, запустите файл mulisp.com и после появления приглашения в виде $, зайдите в системное меню (черный квадратик в левом верхнем углу окна работы) и выполните команду Изменить-Вставить. После вставки текста нажмите клавишу Enter.

При отсутствии ошибок появится приглашение $, после которого можно обращаться к функции и проверять ее работу. Выход из ЛИСПа осуществляется с помощью команды (system).

Обращение к функции можно также включить в файл после описания функции, но для того, чтобы результаты появились на экране, следует использовать функцию PRINT с аргументом – обращением к функции. Например, для функции ATOM\_IN\_LIST из примера2 п 1.10.1 можно записать в файл определение и вызов функции:

(**defun** **ATOM\_IN\_LIST** (l)

  (COND

      ((**NULL** l) nil)  
       ((**ATOM** (**CAR** l)) t)  
       (t (**ATOM\_IN\_LIST** (**CDR** l)))   
  )  
       )

(**PRINT** (**ATOM\_IN\_LIST** '(() 1 (1 2) )))

После запуска ЛИСПа и загрузки файла, на экран будут выведен результат работы функции: t.

В теле функции использование операторов set и setq не допускается!

**Вариант 1**

Опpеделите на языке ЛИСП и проверьте работу функции, преобразующей список L в "двойной" список таким образом, чтобы каждый элемент удвоился.

Например, при L=(1 2 3) функция должна возвращать список (1 1 2 2 3 3).

**Лабораторная работа №2**

Текст функции сохраняйте в файле. Функция может вызывать вспомогательные функции. В теле функций использование операторов set и setq не допускается!

**Вариант 1**

Опpеделите на языке ЛИСП и проверьте работу функции, вычисляющей глубину списка (максимальное количество вложенных скобок).

Например, глубина списка ((((1))) 2 (3 4)) равна 4.

**Лабораторная работа №3**

Текст функции сохраняйте в файле. В теле функции использование операторов set и setq не допускается!

**Вариант 1,6**

Опpеделите на языке ЛИСП функционал, заменяющий все элементы списка, не обладающие определенным свойством, на символ \*. Проверьте работу функционала для предикатов:

-число;

-неположительное число (при вызове используйте лямбда-функцию).

**Лабораторная работа №4**

Программы должны быть написаны на языке Пролог. Для запуска Turbo Prolog запустите файл prolog.exe. Запуск на выполнение Alt+R, переход в окно редактора Alt+E.

**Вариант 1**

Напишите на языке ПРОЛОГ программу, которая преобразует список целых чисел во множество. Список вводится с клавиатуры, цель – внутренняя.

Например: Список [1,2,1,4,1,2,3] преобразуется в [1,2,4,3].

**Лабораторная работа №5**

Программы должны быть написаны на языке Пролог. При преобразовании файла должны быть сохранены переводы строк. К отчету следует приложить тексты файлов с входными и выходными данными. Для просмотра файлов с данными можно открыть вспомогательное окно редактора с помощью клавиши F8 (при этом должно быть активным окно редактора).

**Вариант 1**

В текстовом файле, состоящем из нескольких строк, во всех словах, оканчивающихся на "ing", замените это окончание на "ed".

**Контрольная работа**

Программы должны быть написаны на языке двух языках: Лисп и Пролог. В программе на Прологе исходные данные должны вводиться с клавиатуры, цель - внутренняя.

**Вариант 1**

Вставьте в список новые элементы X перед всеми вхождениями Y.

Например: На Лиспе при x=5, y=1 и списке (1 2 1 5 8 1 0) результатом будет список  
(5 1 2 5 1 5 8 5 1 0).