



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №1 по курсу ”Функциональное и логическое программирование”

Студент Прянишников А. Н.

Группа ИУ7-65Б

Оценка _____

Преподаватели Толпинская Н.Б., Строганов Ю.В.

Практическое задание

Задание 1

Представить следующие списки в виде списочных ячеек:

1. `'(open close halph);`
2. `'((open1)(close2)(halph3));`
3. `'((TOOL)(call));`
4. `'((TOOL1)((call2))((sell)));`
5. `'((one) for all (and (me (for you)))));`
6. `'(((TOOL)(call))((sell)));`

Решение прикреплено к отчёту на отдельном листе.

Задание 2

Используя только функции CAR и CDR, написать выражения, возвращающие:

1. Второй элемент заданного списка;
2. Третий элемент заданного списка;
3. Четвёртый элемент заданного списка;

Решения

Второй элемент:

```
(car (cdr '(1 2 3 4 5)))
```

Третий элемент:

```
(car (cdr (cdr '(1 2 3 4 5))))
```

Четвёртый элемент:

```
(car (cdr (cdr (cdr '(1 2 3 4 5)))))
```

Задание 3

Что будет в результате вычисления выражений?

```
(CAADR '((blue cube)(red pyramid)))
```

Результат: red.

```
(CDAR '((abc)(def)(ghi)))
```

Результат: Nil.

```
(CADR '((abc)(def)(ghi)))
```

Результат: (def).

```
(CADDR '((abc)(def)(ghi)))
```

Результат: (ghi).

Задание 4

Напишите результат вычисления выражений и объясните, как он получен:

```
(list 'Fred 'and 'Wilma)
```

Результат: (Fred and Wilma).

```
(list 'Fred '(and Wilma))
```

Результат: (Fred (and Wilma)).

```
(cons Nil Nil)
```

Результат: (Nil).

```
(cons T Nil)
```

Результат: (T).

```
(cons Nil T)
```

Результат: (Nil . T).

```
(list Nil)
```

Результат: (Nil).

```
(cons '(T) Nil)
```

Результат: ((T)).

```
(list '(one two) '(free temp))
```

Результат: ((one two) (free temp)).

```
(cons 'Fred '(and Wilma))
```

Результат: (Fred and Wilma).

```
(cons 'Fred '(Wilma))
```

Результат: (fred Wilma).

```
(list Nil Nil)
```

Результат: (Nil Nil).

```
(list Nil T)
```

Результат: (Nil T).

```
(cons T (list Nil))
```

Результат: (T Nil).

```
(list '(T) Nil)
```

Результат: ((T) Nil).

```
(cons '(one two) '(free temp))
```

Результат: ((one two) free temp).

Задание 5

Напишите лямбда-выражение и соответствующую функцию:

Напишите функцию (f ar1 ar2 ar3 ar4), возвращающую список:

```
((ar1 ar2) (ar3 ar4))
```

Лямбда-выражение:

```
((lambda (ar1 ar2 ar3 ar4)  
 (list(list ar1 ar2) (list ar3 ar4)))  
 'ar1 'ar2 'ar3 'ar4)
```

Соответствующая функция:

```
(defun f (ar1 ar2 ar3 ar4)
  (list
    (list ar1 ar2)
    (list ar3 ar4)))
```

Напишите функцию (f ar1 ar2), возвращающую список: ((ar1)(ar2))

Лямбда-выражение:

```
((lambda (ar1 ar2)
  (cons (cons ar1 Nil)
    (cons (cons ar2 Nil) Nil))
  'ar1 'ar2))
```

Соответствующая функция:

```
(defun f (ar1 ar2)
  (cons (cons ar1 Nil)
    (cons (cons ar2 Nil) Nil)))
```

Напишите функцию (f ar1), возвращающую список: (((ar1)))

Лямбда-выражение:

```
((lambda (ar1)
  (cons (cons (cons ar1 Nil) Nil) Nil)
  'ar1))
```

Соответствующая функция:

```
(defun f (ar1) (cons (cons (cons ar1 Nil) Nil) Nil))
```

Представление в виде списочных ячеек прикреплено к отчёту.

Ответы на теоретические вопросы

Элементы языка: определение, синтаксис, представление в памяти

Элементы языка — атомы и точечные пары (структуры, которые строятся с помощью унифицированных структур - блоков памяти - бинарных узлов). Атомы бывают:

- **символы** (идентификаторы) — синтаксически представляют собой набор литер (последовательность букв и цифр, начинающаяся с буквы; могут быть связанные и несвязанные);
- **специальные символы** — используются для обозначения «логических» констант (T, Nil);
- **самоопределимые атомы** — числа, строки - последовательность символов в кавычках ("abc").

Синтаксис элементов языка:

Точечная пара ::= (<атом> . <атом>) | (<точечная пара> . <атом>) | (<атом> . <точечная пара>) | (<точечная пара> . <точечная пара>)

Список ::= <пустой список> | <непустой список>, где

<пустой список> ::= () | Nil,

<непустой список> ::= (<S-выражение> . <список>),

Список — частный случай S-выражения.

Синтаксически любая структура (точечная пара или список) заключается в круглые скобки: (A . B) — точечная пара. (A) — список из одного элемента. Непустой список — (A . (B . (C . (D . Nil)))) или (A B C D) Пустой список — Nil или ().

Элементы списка могут быть списками, например — ((A (B C) (D (E)))). Таким образом, синтаксически наличие скобок является признаком структуры — списка или точечной пары.

Любая непустая структура Lisp в памяти представляется списковой ячейкой, хранящий два указателя: на голову (первый элемент) и хвост (все остальное).

Особенности языка Lisp. Структура программы.

Символ апостроф

Lisp - язык символьной обработки. В Lisp программа и данные представлены списками. По умолчанию список считается вычислимой формой, в которой 1 элемент - название функции, остальные элементы - аргументы функции.

Т.к. и программа и данные представлены списками, то их нужно как-то различать. Для этого была создана функция `quote`, а `'` - ее сокращенное обозначение. `quote` - функция, блокирующая вычисление.

Базис языка Lisp. Ядро языка

Базис - минимальный набор конструкций языка и структур данных, который позволяет решить любую задачу.

Базис в Lisp образуют:

- атомы;
- структуры;
- базовые функции;
- функционалы.