



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 №2
По курсу: "Функциональное и Логическое программирование"

Студент _____ Сукочева Алис
Группа _____ ИУ7-63Б
Название предприятия _____ МГТУ им. Н. Э. Баумана, каф. ИУ7
Тема _____ Базовые функции.

Студент:	_____	Сукочева А.
	подпись, дата	Фамилия, И.О.
Преподаватель:	_____	Толпинская Н.Б.
	подпись, дата	Фамилия, И. О.
Преподаватель:	_____	Строганов Ю.В.
	подпись, дата	Фамилия, И. О.

Практические задания

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

а) второй

```
1 ;; CDR от '(a b c d e f) вернет хвост (b c d e f)
2 ;; CAR от '(b c d e f) вернет голову b (Т.е. второй элемент)
3 (CAR (CDR '(a b c d e f)))
```

б) третий

```
1 ;; CDR от '(a b c d e f) вернет хвост (b c d e f)
2 ;; CDR от '(b c d e f) вернет хвост (c d e f)
3 ;; CAR от '(c d e f) вернет голову c (Т.е. третий элемент)
4 (CAR (CDR (CDR '(a b c d e f))))
```

в) четвертый элементы заданного списка.

```
1 ;; CDR от '(a b c d e f) вернет хвост (b c d e f)
2 ;; CDR от '(b c d e f) вернет хвост (c d e f)
3 ;; CDR от '(c d e f) вернет хвост (d e f)
4 ;; CAR от '(d e f) вернет голову d (Т.е. четвертый элемент)
5 (CAR (CDR (CDR (CDR '(a b c d e f)))))
```

Что будет в результате вычисления выражений?

```
1 ;; CAADR = CAR(CAR(CDR '((blue cube) (red pyramid))))
2 ;; 1. (CDR '((blue cube) (red pyramid))) вернет хвост ((red pyramid))
3 ;; 2. CAR '((red pyramid)) вернет голову (red pyramid)
4 ;; 3. CAR '(red pyramid) вернет голову red
5 (CAADR '((blue cube) (red pyramid))) ;; red
6 ;; CAR '((abc) (def) (ghi)) вернет голову (abc)
7 ;; CDR '(abc) вернет хвост nil
8 (CDAR '((abc) (def) (ghi))) ;; nil
9 (CADR '((abc) (def) (ghi))) ;; (def)
10
11 ;; 1. CDR = ((def) (ghi))
12 ;; 2. CDR = ((ghi))
13 ;; 3. CAR = (ghi)
14 (CADDR '((abc) (def) (ghi))) ;; (ghi)
```

4. Напишите результат вычисления выражений:

```
1 (list 'Fred 'and Wilma) ;; Error (The variable WILMA is unbound).
2 (list 'Fred '(and Wilma)) ;; (Fred (and Wilma))
3 (cons Nil Nil) ;; (Nil)
```

```

4 (cons T Nil) ;; (T) ;; T - true (bool)
5 (cons Nil T) ;; (Nil . T)
6 (list Nil) ;; (Nil)
7 (cons (T) Nil) ;; Error fixme: (cons '(T) Nil)
8 (list '(one two) '(free temp)) ;; ((one two) (free temp))
9
10 (cons 'Fred '(and Willma)) ;; (Fred and Willma)
11 (cons 'Fred '(Wilma)) ;; (Fred Willma)
12 (list Nil Nil) ;; (Nil Nil)
13 (list T Nil) ;; (T Nil)
14 (list Nil T) ;; (Nil T)
15
16 (cons T (list Nil)) ;; (T Nil)
17 ;; 1. (list Nil) = (Nil)
18 ;; 2. (cons T (Nil)) = (T Nil)
19
20 (list (T) Nil) ;; Error fixme: (list '(T) Nil)
21 (cons '(one two) '(free temp)) ;; ((one two) free temp)

```

Написать функции.

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

```

1 (defun f1 (arg1 arg2 arg3 arg4) (list (list arg1 arg2) (list arg3 arg4)))
2 (f1 'a 'b 1 2)

```

Написать функцию (f ar1 ar2), возвращающую ((ar1) (ar2)).

```

1 (defun f2 (arg1 arg2) (list (list arg1) (list arg2)))
2 (f2 1 2)

```

Написать функцию (f ar1), возвращающую (((ar1))).

```

1 (defun f3 (arg1) (list (list (list arg1))))
2 (f3 'a)

```

Ответы на вопросы

Классификация функций.

- а) Чистые математические функции - имеет фиксированное количество аргументов и один результат.
- б) Специальные функции - произвольное количество аргументов.
- в) Псевдофункции - создают эффект на экране.
- г) Рекурсивные функции.
- д) Функции с вариантными значениями, которые возвращают одно значение.
- е) Функции высших порядков - используются для построения синтаксически управляемых программ.

Базис лиспа.

Базис - минимальный набор средств для решения любой задачи.

Базис:

- 1) атомы и бинарные узлы;
- 2) atom, eq, cons, car, cdr, cond, quote, eval.

atom проверяет, является ли объект, переданный в качестве аргумента, атомом.

```
1 (atom 'a) ;; t
2 (atom '(a b c)) ;; nil
```

eq проверяет идентичность двух символов.

```
1 (eq 'a 'b) ;; nil
2 (eq 'a 'a) ;; t
```

cond - сокращение от англ. condition – условие. Не имеет фиксированного количества аргументов. Каждый аргумент - это список, голова которого рассматривается как условие, и если оно истинно, то результатом будет хвост рассматриваемого списка.

```
1 (cond ((eq 'A 'B) 'are_equal)
2       (T 'not_equal)) ;; NOT_EQUAL
```

eval - выполняет двойное вычисление своего аргумента.

```
1 (eval (cons (quote car) (quote ('(A B))))) => A
2 |-----(car '(A B))-----|
```

Функции car и cdr

car и cdr - базовые функции доступа к данным.

car - принимает точечную пару или список и возвращает голову (первый элемент).

cdr - принимает точечную пару или список и возвращает хвост (все элементы, кроме первого).

Отличие list и cons

cons - имеет фиксированное количество аргументов (два). В случае, когда аргументами являются атомы создает точечную пару. В случае, когда первый аргумент атом а второй список, атом становится головой, а второй аргумент (список) становится хвостом.

```
1 (cons 'a 'b)           ;; (A . B)
2 (cons 'a '(a b c))    ;; (A A B C)
3 (cons '(a c) '(b d))  ;; ((A C) B D)
4 (cons 'a 'v 'd)       ;; Error (invalid number of arguments: 3)
```

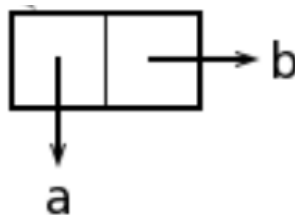


Рисунок 0.1 — Результат cons.

list - не имеет фиксированное количество аргументов. Создает список, у которого голова - это первый аргумент, хвост - все остальные аргументы.

```
1 (list 'a 'b)           ;; (A B)
2 (list 'a 'b 'v '(c d) 'd) ;; (A B V (C D) D)
```

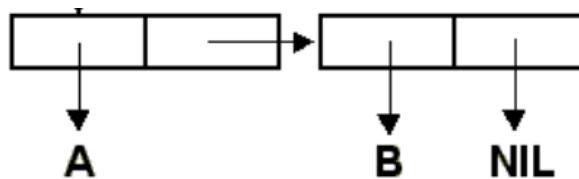


Рисунок 0.2 — Результат list.

cons - имеет фиксированное число аргументов и более экономный по памяти.

Ядро - основные действия, которые наиболее часто используются. Ядро шире, чем базис.