

Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Сибирский государственный университет телекоммуникаций и информатики»  
(СибГУТИ)

Кафедра прикладной математики и кибернетики

Лабораторная работа № 1  
по дисциплине «Функциональное и логическое программирование»

Бригада № 2

Выполнил: студент группы ИП-814

Якунин А.В.

Работу проверил: ассистент кафедры ПМиК

Пащенко А.А.

Новосибирск 2020 г.

## **Содержание:**

1. Задание.....	3
2. Листинг программы.....	4
3. Демонстрация работы программы.....	5

## 1. Задание

1. Напишите сложную функцию, используя композиции функций CAR и CDR, которая возвращает атом \* при применении к следующему списку:

**(1 (((2 3) (4 5) \*) (6)))**

2. Объясните работу функций и определите результат обращения:

**(list '(a b) 7 '(()))**

3. Из атомов 1, 2, 3, nil создайте указанные списки двумя способами:

а) с помощью композиций функций CONS;

б) с помощью композиций функций LIST.

**(1 (2) 3)**

4. С помощью DEFUN определите функцию, которая возвращает измененный список по заданию (в теле функции разрешается использовать только следующие встроенные функции: CAR, CDR, CONS, APPEND, LIST, LAST, BUTLAST с одним аргументом). Проверьте её работу, организовав обращение к функции со списками разной длины.

**Функция меняет местами первый и второй элементы списка**

## 2. Листинг программы

**1)**

```
( car (cdr ( cdr( car (car (cdr (car ( car `(((1 ((2 3) (4 5) *) (6))))))))))))))
```

**2)**

```
(list '(a b) 7 '(()))
```

**3)**

```
(list `(1 (2) 3))
```

```
(cons `1 (cons `(2) `(3)))
```

**4)**

```
(defun main()
  (defvar x `(1 2 3 4 5 6 7))
  (write x)
  (defvar first (CAR x))
  (defvar second (CAR (CDR x)))
  (defvar remnant (CDR (CDR x)))
  (CONS second (CONS first remnant))
)
(main)
```

### 3. Демонстрация работы программы

The image displays two screenshots of the LispIDE application, which is running on a Windows operating system. The window title is "LispIDE - D:\Lab\Code\FLP\lab1.1.lisp".

The top screenshot shows the editor with the following code in `lab1.lisp`:

```
1 (car (cdr (cdr (car (car (cdr (car (car `(((1 ((2 3) (4 5) *) (6))))))))))))))
2 (list '(a b) 7 '())
3 (list '(1 (2) 3))
4 (cons '1 (cons '(2) '(3)))
```

The bottom pane shows the REPL output:

```
Copyright (c) Bruno Haible, Michael Stoll 1992, 1993
Copyright (c) Bruno Haible, Marcus Daniels 1994-1997
Copyright (c) Bruno Haible, Pierpaolo Bernardi, Sam Steingold 1998
Copyright (c) Bruno Haible, Sam Steingold 1999-2000
Copyright (c) Sam Steingold, Bruno Haible 2001-2010

Напечатайте :h и нажмите Ввод для получения справки.

[1]>
*
[2]>
((A B) 7 (NIL))
[3]>
((1 (2) 3))
[4]>
(1 (2) 3)
[5]>
```

The bottom screenshot shows the same application with a different code snippet in `lab1.lisp`:

```
1 (defun main()
2   (defvar x '(1 2 3 4 5 6 7))
3   (write x)
4   (defvar first (CAR x))
5   (defvar second (CAR (CDR x)))
6   (defvar remnant (CDR (CDR x)))
7   (CONS second (CONS first remnant))
8 )
9 (main)
10
```

The bottom pane shows the REPL output, including a memory visualization:

```
-----+-----      8      0      8      8      0      8      8
      | 00000  8000000  0008000  00000  8

Добро пожаловать GNU CLISP 2.49 (2010-07-07) <http://clisp.cons.org/>

Copyright (c) Bruno Haible, Michael Stoll 1992, 1993
Copyright (c) Bruno Haible, Marcus Daniels 1994-1997
Copyright (c) Bruno Haible, Pierpaolo Bernardi, Sam Steingold 1998
Copyright (c) Bruno Haible, Sam Steingold 1999-2000
Copyright (c) Sam Steingold, Bruno Haible 2001-2010

Напечатайте :h и нажмите Ввод для получения справки.

[1]>
MAIN
[2]> (1 2 3 4 5 6 7)
(2 1 3 4 5 6 7)
[3]>
```