



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 №6  
по дисциплине «Функциональное и логическое  
программирование»**

Тема Использование функционалов

Студент Варламова Е. А.

Группа ИУ7-61Б

Оценка (баллы) \_\_\_\_\_

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

## Задание 1

### Постановка задачи

Напишите функцию, которая уменьшает на 10 все числа из списка-аргумента этой функции.

```
1 (defun dec (lst) (mapcar #'(lambda (elem) (- elem 10)) lst))
```

## Задание №2

Напишите функцию, которая умножает на заданное число-аргумент все числа из заданного списка-аргумента.

Все элементы списка — числа.

```
1 (defun mul (lst1 num) (mapcar #'(lambda (elem) (* elem num)) lst1))
```

Элементы списка – любые объекты.

```
1 (defun mul (lst num)
2   (mapcar
3     #'(lambda (elem)
4       (cond ((numberp elem) (* elem num))
5             ((listp elem) (mul elem num))
6             ))
7     lst)
8 )
```

## Задание №3

Написать функцию, которая по своему списку-аргументу lst определяет является ли он палиндромом (то есть равны ли lst и (reverse lst)).

### Решение

Листинг 1: Решение задания №3

```
1 (defun to_pairs (lst1 lst2) (mapcar #'equal lst1 lst2))
2
3 (defun rdc (lst) (reduce #'(lambda (accum elem) (and accum elem)) lst :
4   initial-value T))
5
6 (defun is_same (lst1 lst2) (rdc (to_pairs lst1 lst2)))
7
8 (defun is_poly (lst) (is_same lst (reverse lst)))
```

## Задание №4

Написать предикат `set-equal`, который возвращает `t`, если два его множества- аргумента содержат одни и те же элементы, порядок которых не имеет значения.

### Решение

Листинг 2: Решение задания №4

```
1 (defun rdc_and (lst) (reduce #'(lambda (accum elem) (and accum elem)) lst :  
  initial-value T))  
2  
3 (defun rdc_or (lst) (reduce #'(lambda (accum elem) (or accum elem)) lst :  
  initial-value nil))  
4  
5 (defun set_equal (l1 l2)  
6 (cond ((eql (length l1) (length l2))  
7 (rdc_and (mapcar #'(lambda (x)  
8 (rdc_or (mapcar #'(lambda (y) (equal x y)) l2))  
9 ) l1)))  
10 ))
```

## Задание №5

Написать функцию которая получает как аргумент список чисел, а возвращает список квадратов этих чисел в том же порядке

### Решение

Листинг 3: Решение задания №5

```
1 (defun sqr (lst)  
2 (mapcar #'(lambda (x) (* x x)) lst)  
3 )
```

## Задание №6

Напишите функцию, `select-between`, которая из списка-аргумента, содержащего только числа, выбирает только те, которые расположены между двумя указанными границами- аргументами и возвращает их в виде списка (упорядоченного по возрастанию списка чисел (+ 2 балла)).

### Решение

#### Листинг 4: Решение задания №5

```
1 (defun select-between (lst left right) (mapcan #'(lambda (elem) (if (and (<
    elem right) (> elem left)) (list elem) nil)) lst))
```

## Задание №7

Написать функцию, вычисляющую декартово произведение двух своих списков- аргументов. ( Напомним, что  $A \times B$  это множество всевозможных пар  $(a, b)$ , где  $a$  принадлежит  $A$ , принадлежит  $B$ .)

## Решение

#### Листинг 5: Решение задания №5

```
1 (defun decart (lstx lsty) (mapcan #'(lambda (x) (mapcar #'(lambda (y) (list
    x y)) lsty)) lstx))
```

## Задание №8

Почему так реализовано reduce, в чем причина?

#### Листинг 6: Решение задания №5

```
1 (reduce #' +0) -> 0
2 (reduce #' + ()) -> 0
```

## Задание №9

Пусть list-of-list список, состоящий из списков. Написать функцию, которая вычисляет сумму длин всех элементов list-of-list, т.е. например для аргумента  $((1\ 2)\ (3\ 4)) \rightarrow 4$ .

## Решение

#### Листинг 7: Решение задания №5

```
1 (defun sum-sizes (lst)
2 (reduce #'(lambda (accum elem) (+ accum (length elem))) lst :initial-value
    0)
3 )
```