



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 №7 по курсу «Функциональное и логическое программирование»

Тема Рекурсивные функции

Студент Климов И.С.

Группа ИУ7-62Б

Оценка (баллы) \_\_\_\_\_

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

Москва — 2022 г.

### Задание 1 (1)

Написать хвостовую рекурсивную функцию my-reverse, которая развернет верхний уровень своего списка-аргумента lst.

#### Решение

```
(defun move_to (lst result)
  (cond ((null lst) result)
        (T (move_to (cdr lst) (cons (car lst) result)))))

(defun my_reverse (lst)
  (move_to lst ()))
```

### Задание 2 (3)

Написать функцию, которая возвращает первый элемент списка - аргумента, который сам является непустым списком.

#### Решение

```
(defun find_list (lst element)
  (cond ((or (null lst) element) element)
        ((and (listp (car lst)) (> (length lst) 0))
         (find_list (cdr lst) (car lst)))
        (T (find_list (cdr lst) element))))

(defun find_first_list (lst)
  (find_list lst NIL))
```

### Задание 3 (4)

Написать функцию, которая выбирает из заданного списка только те числа, которые больше 1 и меньше 10. (Вариант: между двумя заданными границами).

## Решение

```
(defun between (lst result left right)
  (cond ((null lst) result)
        ((and (> (car lst) left) (< (car lst) right))
         (between (cdr lst) (append result (list (car lst))) left right))
        (T (between (cdr lst) result left right))))

(defun between_1_and_10 (lst)
  (between lst () 1 10))
```

## Задание 4 (7)

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

- a) все элементы списка – числа,
- b) элементы списка – любые объекты.

## Решение

```
(defun mul_by_number_v1 (lst num &optional (result NIL))
  (cond ((null lst) result)
        (T (mul_by_number_v1 (cdr lst) num (append result (list (* (car lst) num)))))))

(defun mul_by_number_v2 (lst num &optional (result NIL))
  (cond ((null lst) result)
        ((numberp (car lst))
         (mul_by_number_v2 (cdr lst) num (append result (list (* (car lst) num)))))
        (T (mul_by_number_v2 (cdr lst) num (append result (list (car lst)))))))
```

## Задание 5 (8)

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

```
(defun my_remove (x lst &optional (deleted 0) (result NIL))
  (cond ((null lst) result)
        ((and (= deleted 0) (equal x (car lst)))
         (my_remove x (cdr lst) 1 result))
        (T (my_remove x (cdr lst) deleted (append result (list (car lst)))))))

(defun my_max (lst &optional (max NIL))
  (cond ((null lst) max)
        ((or (null max) (> (car lst) max)) (my_max (cdr lst) (car lst)))
        (T (my_max (cdr lst) max))))

(defun my_sort (lst &optional result)
  (cond ((null lst) result)
        (T (my_sort (my_remove (my_max lst) lst)
                     (cons (my_max lst) result)))))

(defun select_between (lst left right)
  (cond ((> left right) (select_between lst right left))
        (T (select_between lst left right))))
```

## Задание 6 (8)

Написать рекурсивную версию (с именем `rec-add`) вычисления суммы чисел заданного списка:

- a) одноуровневого смешанного,
- b) структурированного.

## Решение

```
(defun rec_add_v1 (lst &optional (sum 0))
  (cond ((null lst) sum)
        ((numberp (car lst)) (rec_add_v1 (cdr lst) (+ sum (car lst))))
        (T (rec_add_v1 (cdr lst) sum))))

(defun rec_add_v2 (lst)
  (cond ((numberp lst) lst)
        ((atom lst) 0)
        (T (+ (rec_add_v2 (car lst)) (rec_add_v2 (cdr lst))))))
```

## Задание 7 (9)

Написать рекурсивную версию с именем `recnth` функции `nth`.

### Решение

```
(defun recnth (n lst)
  (cond ((= n 0) (car lst))
        (T (recnth (- n 1) (cdr lst)))))
```

## Задание 8 (10)

Написать рекурсивную функцию `allodd`, которая возвращает `t` когда все элементы списка нечетные.

### Решение

```
(defun my_odd (x)
  (and (numberp x) (= (mod x 2) 1)))

(defun all_odd (lst)
  (cond ((null lst) T)
        ((my_odd (car lst)) (all_odd (cdr lst)))
        (T NIL)))
```

## Задание 9 (11)

Написать рекурсивную функцию, которая возвращает первое нечетное число из списка (структурированного), возможно создавая некоторые вспомогательные функции.

### Решение

```
(defun first_odd (lst)
  (cond ((my_odd lst) lst)
        ((atom lst) NIL)
        (T (or (first_odd (car lst)) (first_odd (cdr lst))))))
```

## Задание 10 (12)

Используя cons-дополняемую рекурсию с одним тестом завершения, написать функцию, которая получает как аргумент список чисел, а возвращает список квадратов этих чисел в том же порядке.

### Решение

```
(defun square_list (lst)
  (cond ((null lst) NIL)
        (T (cons (* (car lst) (car lst)) (square_list (cdr lst))))))
```