



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»

## Отчёт по лабораторной работе №7 по курсу «Функциональное и логическое программирование»

Тема Рекурсивные функции

Студент Прянишников А.Н.

Группа ИУ7-65Б

Оценка (баллы) \_\_\_\_\_

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

# Практическое задание

## Задание 1

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

```
1 (defun my-reverse (lst &optional (result nil))  
2 (if (null lst) result (my-reverse (rest lst) (cons (first lst) result))))
```

## Задание 2

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

```
1 (defun return-list (lst)  
2 (if (listp (car lst)) (car lst) (return-list (rest lst))))
```

## Задание 3

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

```
1 (defun get-between (a b lst &optional (result nil))  
2 (if (null lst) (my-reverse result)  
3     (if (and (>= (car lst) a) (<= (car lst) b))  
4         (get-between a b (cdr lst) (cons (first lst) result))  
5         (get-between a b (cdr lst) result)  
6     )))
```

## Задание 4

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

- все элементы списка — числа;
- элементы списка – любые объекты.

```
1 (defun mult (lst koef &optional (result nil))
2 (if (null lst) (my-reverse result)
3 (mult (cdr lst) koef
4   (if (numberp (car lst))
5     (cons (* (car lst) koef) result)
6     (cons (car lst) result) ))))
```

## Задание 5

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)).

```
1 (defun select-between (a b lst &optional (result nil))
2 (if (null lst) (sort result #'<)
3   (if (and (>= (car lst) a) (<= (car lst) b))
4     (get-between a b (cdr lst) (cons (first lst) result))
5     (get-between a b (cdr lst) result)
6 )))
```

## Задание 6

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

- одноуровневого смешанного;

- структурированного.

```
1 (defun rec-add (lst &optional (result 0))
2   (if (null lst) result (rec-add (cdr lst) (+ result (car lst)))))
3
4 (defun rec-add (lst &optional (result 0))
5   (if (null lst)
6       result
7       (if (listp (car lst))
8           (rec-add (cdr lst) (+ result (rec-add (car lst)))))
9           (rec-add (cdr lst) (+ result (car lst)))))
10  )))
```

## Задание 7

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

```
1 (defun recnth (lst n &optional (index 0))
2   (if (null lst) Nil
3       (if (= index n)
4           (car lst)
5           (recnth (cdr lst) n (+ 1 index)))
6   )))
```

## Задание 8

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

```
1 (defun allodd (lst)
2   (if (null lst) T
3       (if (oddp (car lst))
4           (allodd (cdr lst))
5           Nil)))
```

## Задание 9

Написать рекурсивную функцию, которая возвращает первое нечетное число из списка (структурированного), возможно создавая некоторые вспо-

могательные функции.

```
1 (defun find-first-odd (lst)
2   (if (null lst) Nil
3       (if (listp (car lst))
4           (let ((res (find-first-odd (car lst))))
5             (if (equal res Nil)
6                 (find-first-odd (cdr lst))
7                 res))
8           (if (oddp (car lst))
9               (car lst)
10              (find-first-odd (cdr lst))))
11   )
12 ))
```

## Задание 10

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.

```
1 (defun square (lst &optional (result nil))
2   (if (null lst)
3       (my-reverse result)
4       (square (cdr lst) (cons (* (first lst) (first lst)) result)))
5   ))
```