



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 курсу «Функциональное и логическое программирование»

Студент ИУ7-61Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Д. В. Шубенина  
(И. О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Н. Б. Толпинская  
(И. О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Ю. В. Строганов  
(И. О. Фамилия)

2024 г.

# СОДЕРЖАНИЕ

<b>1</b>	<b>Практические задания</b>	<b>3</b>
1.1	Задание 1 . . . . .	3
1.2	Задание 2 . . . . .	3
1.3	Задание 3 . . . . .	3
1.4	Задание 4 . . . . .	3
1.5	Задание 5 . . . . .	4
1.6	Задание 6 . . . . .	4
1.7	Задание 7 . . . . .	5
1.8	Задание 8 . . . . .	5
1.9	Задание 9 . . . . .	6
1.10	Задание 10 . . . . .	6

# 1 Практические задания

## 1.1 Задание 1

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

```
1 (defun my-reverse (lst &optional (res Nil))
2   (cond ((null lst) res)
3         (t (my-reverse (cdr lst)
4                        (cons (car lst) res)))))
```

## 1.2 Задание 2

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

```
1 (defun get-first (lst)
2   (cond ((null lst) Nil)
3         ((and (listp lst)
4               (not (null (caar lst))))
5          (car lst))
6         (t (get-first (cdr lst)))))
```

## 1.3 Задание 3

Написать функцию, которая выбирает из заданного списка только те числа, которые больше 1 и меньше 10.

```
1 (defun sel-between (lst &optional (res Nil))
2   (cond ((null lst) res)
3         ((< 1 (car lst) 10)
4          (sel-between (cdr lst)
5                       (nconc res (cons (car lst) Nil))))
6         (t (sel-between (cdr lst) res))))
```

## 1.4 Задание 4

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

- а) все элементы списка — числа,

b) элементы списка — любые объекты.

```
1 (defun mul-lst-a (lst num)
2   (cond ((null lst) Nil)
3         (T (cons (* (car lst) num)
4                   (mul-lst-a (cdr lst) num)))))
5
6 (defun mul-lst-b (lst num)
7   (cond ((null lst) Nil)
8         ((numberp (car lst)) (cons (* (car lst) num)
9                                     (mul-lst-b (cdr lst) num)))
10        (T (cons (car lst)
11                  (mul-lst-b (cdr lst) num))))
12   ))
```

## 1.5 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).

```
1 (defun select-between (lst a b)
2   (cond ((null lst) Nil)
3         (T
4          (cond ((< a (car lst) b)
5                (cons (car lst)
6                      (select-between (cdr lst) a b)))
7                (T (select-between (cdr lst) a b)))
8          ))
9   ))
```

## 1.6 Задание 6

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

a) одноуровневого смешанного,

b) структурированного.

```

1 ;; пункт a)
2 (defun rec-add-a (lst)
3   (if (null lst)
4       0
5       (+
6         (if (numberp (car lst))
7             (car lst)
8             0)
9         (rec-add-a (cdr lst))
10      )
11   ))
12
13 ;; пункт b)
14 (defun rec-add-b (lst)
15   (if (null lst)
16       0
17       (+
18         (if (numberp (car lst))
19             (car lst)
20             (rec-add-b (car lst)))
21         (rec-add-b (cdr lst))
22      )
23   ))

```

## 1.7 Задание 7

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

```

1 (defun recnth-impl (n lst)
2   (cond ((null lst) Nil)
3         ((= n 0) (car lst))
4         ((> n 0) (recnth-impl (- n 1) (cdr lst))))
5 )

```

## 1.8 Задание 8

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

```

1 (defun allodd (lst)
2   (cond ((null lst) T)
3         ((or (not (numberp (car lst)))
4              (evenp (car lst))) Nil)

```

```
5 | (T (allodd (cdr lst)))
6 | ))
```

## 1.9 Задание 9

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

```
1 | (defun first-odd (lst)
2 |   (cond ((null lst) Nil)
3 |         ((numberp lst)
4 |          (if (oddp lst)
5 |              lst))
6 |         (T (or (first-odd (car lst))
7 |                (first-odd (cdr lst)))))
8 | ))
```

## 1.10 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 f (lst)
2 |   (cond ((null lst) Nil)
3 |         (T (cons (* (car lst) (car lst))
4 |                   (f (cdr lst)))))
5 | ))
```