



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 Использование рекурсии

Студент Богаченко А. Е.

Группа ИУ7-65Б

Оценка (баллы) _____

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

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

Листинг 1 – Задание 1

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

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

Листинг 2 – Задание 2

```
1 (defun first-sublist (lst)
2   (and lst (if (listp (car lst))
3               (car lst)
4               (first-sublist (cdr lst)))))
```

3. 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

Листинг 3 – Задание 3

```
1 (defun select-between (from to lst &optional (acc nil))
2   (if (null lst)
3       (sort acc #'<)
4       (let ((head (car lst)))
5         (if (and (< head to) (> head from))
6             (select-between from to (cdr lst) (cons head acc))
7             (select-between from to (cdr lst) acc))))))
```

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

Листинг 4 – Задание 4

```
1 (defun rec-add (lst &optional (acc 0))
2   (if (null lst)
3       acc
4       (rec-add (cdr lst) (+ acc (car lst)))))
```

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

Листинг 5 – Задание 5

```
1 (defun recnth (n lst)
2   (and lst (if (zerop n)
3               (car lst)
4               (recnth (- n 1) (cdr lst)))))
```

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

Листинг 6 – Задание 6

```
1 (defun allodd (lst)
2   (or (null lst)
3       (and (oddp (car lst))
4            (allodd (cdr lst)))))
```

7. 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

Листинг 7 – Задание 7

```
1 (defun square (lst)
2   (and lst (cons ((lambda (x) (* x x)) (car lst))
3                 (square (cdr lst)))))
```