



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 дисциплине
"Функциональное и логическое программирование"

Тема Рекурсивные функции

Студент Малышев И. А.

Группа ИУ7-61Б

Оценка (баллы) _____

Преподаватель: Толпинская Н. Б.

Москва — 2022 г.

Практические задания

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

Листинг 1: Решение задания №1

```
1 (defun my-reverse (lst)
2   (my-rev lst ()))
3
4 (defun my-rev (lst acc)
5   (cond ((null lst) acc)
6         (t (my-rev (cdr lst) (cons (car lst) acc)))))
```

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

Листинг 2: Решение задания №2

```
1 (defun find-first-lst (lst)
2   (cond ((null lst) nil)
3         ((listp (car lst)) (car lst))
4         (t (find-first-lst (cdr lst)))))
```

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

Листинг 3: Решение задания №3

```
1 (defun a1b10 (x) (and (> x 1) (< x 10)))
2
3 (defun a1b10-lst (lst acc)
4   (cond ((null lst) acc)
5         ((a1b10 (car lst)) (a1b10-lst (cdr lst) (cons (car lst) acc)))
6         (t (a1b10-lst (cdr lst) acc))))
7
8 (defun above-one-below-ten (lst)
9   (my-reverse (a1b10-lst lst acc)))
```

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

- а) все элементы списка — числа,
- б) элементы списка – любые объекты.

Листинг 4: Решение задания №4

```
1 ; a)
2 (defun nums-mul (lst mul)
3   (cond ((null lst) nil)
4         (t (cons (* (car lst) mul) (nums-mul (cdr lst) mul)))))
5
6 ; b)
7 (defun nums-mul (lst mul)
8   (cond ((null lst) nil)
9         ((numberp (car lst)) (cons (* (car lst) mul) (nums-mul (cdr lst)
10                               mul))))
10  (t (cons (car lst) (nums-mul (cdr lst) mul)))))
```

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)).

Листинг 5: Решение задания №5

```
1 (defun select-between-internal (from to lst acc)
2   (cond ((null lst) acc)
3         (T (let ((head (car lst)))
4             (cond ((and (< head to) (> head from))
5                   (select-between-internal from to (cdr lst) (cons head
6                                                                     acc))))
7             (T (select-between-internal from to (cdr lst) acc))))))
8 (defun select-between (from to lst)
9   (my-reverse (select-between-internal from to lst nil)))
```

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

- а) одноуровневого смешанного,
- б) структурированного.

Листинг 6: Решение задания №6

```
1 ; a)
2 (defun rec-add-internal (lst sum)
3   (cond ((null lst) sum)
4         (t (rec-add-internal (cdr lst) (+ sum (car lst))))))
5
6 (defun rec-add (lst)
7   (rec-add-internal lst 0))
8
9
10 ; b)
11 (defun rec-add-internal (lst sum)
12   (cond ((null lst) sum)
13         ((numberp (car lst)) (rec-add-internal (cdr lst) (+ sum (car
14   lst)))))
15   ((listp (car lst)) (rec-add-internal (cdr lst) (rec-add-internal
16   (car lst) sum))))
17   (t (rec-add-internal (cdr lst) sum))))
18
19 (defun rec-add (lst)
20   (rec-add-internal lst 0))
```

7. Написать рекурсивную версию с именем recnth функции nth.

Листинг 7: Решение задания №7

```
1 (defun recnth (lst n)
2   (cond ((null lst) nil)
3         ((< n 0) nil)
4         ((= n 0) (car lst))
5         (t (recnth (cdr lst) (- n 1)))))
```

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

Листинг 8: Решение задания №8

```
1 (defun allodd (lst)
2   (cond ((null lst) nil)
3         ((null (cdr lst)) (oddp (car lst)))
4         ((oddp (car lst)) (allodd (cdr lst)))
5         (t nil)))
```

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

Листинг 9: Решение задания №9

```
1 (defun first-odd (lst)
2   (cond ((null lst) nil)
3         ((and (numberp (car lst)) (oddp (car lst))) (car lst))
4         ((listp (car lst)) (first-odd (car lst)))
5         (t (first-odd (cdr lst)))))
```

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.

Листинг 10: Решение задания №10

```
1 (defun square-list (lst)
2   (cond ((null lst) lst)
3         ((atom lst) (* lst lst))
4         (t (cons (square-list (car lst))
5                   (square-list (cdr lst)))))
```