



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»

на тему: «Рекурсивные функции»

Студент ИУ7-63Б
(Группа)

(Подпись, дата)

Лысцев Н. Д.
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Толпинская Н. Б.
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Строганов Ю. В.
(И. О. Фамилия)

2024 г.

1 Практические задания

1.1 Задание 1

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

Листинг 1.1 – Решение задания №1

```
(defun _my-reverse(lst res)
  (cond ((null lst) res)
        (t (_my-reverse (cdr lst) (cons (car lst) res)))))

(defun my-reverse(lst)
  (_my-reverse lst nil))
```

1.2 Задание 2

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

Листинг 1.2 – Решение задания №2

```
(defun get_first_noempt_lst(lst)
  (cond ((null lst) nil)
        ((and (listp (car lst)) (> (length (car lst)) 0)) (car lst))
        (t (get_first_noempt_lst (cdr lst)))))
```

1.3 Задание 3

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

Листинг 1.3 – Решение задания №3

```
(defun select_between1(a b lst)
  (cond ((null lst) nil)
        ((and (numberp (car lst)) (> (car lst) a) (< (car lst) b))
         (cons (car lst) (select_between1 a b (cdr lst))))
        ((atom (car lst)) (select_between1 a b (cdr lst)))
        (t (select_between1 a b (car lst)))))
```

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,
- б) элементы списка — любые объекты.

Листинг 1.4 – Решение задания №4

```
(defun mul_list_by_num(n lst) ; все элементы списка - числа
  (cond ((null lst) nil)
        (t (cons (* (car lst) n) (mul_list_by_num n (cdr lst))))))

(defun _mul_list_by_num2(n lst res) ; все элементы списка -
  любые объекты
  (cond ((null lst) res)
        ((numberp (car lst)) (_mul_list_by_num2 n (cdr lst)
          (cons (* (car lst) n) res)))
        ((atom (car lst)) (_mul_list_by_num2 n (cdr lst) (cons
          (car lst) res)))
        (t (_mul_list_by_num2 n (cdr lst) (cons
          (_mul_list_by_num2 n (car lst) ()) res)))))

(defun mul_list_by_num2(n lst)
  (reverse (_mul_list_by_num2 n lst ())))
```

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 списка чисел (+ 2 балла)).

Листинг 1.5 – Решение задания №5

```
(defun select_between(a b lst)
  (cond ((null lst) nil)
        ((and (numberp (car lst)) (> (car lst) a) (< (car lst) b))
          (cons (car lst) (select_between a b (cdr lst))))
        ((atom (car lst)) (select_between a b (cdr lst)))
        (t (select_between a b (car lst)))))
```

1.6 Задание 6

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

- а) одноуровневого смешанного;
- б) структурированного.

Листинг 1.6 – Решение задания №6

```
(defun rec-add(lst) ; одноуровневого смешанного
  (cond ((null lst) 0)
        ((numberp (car lst))
         (+ (car lst) (rec-add (cdr lst))))
        (t (rec-add (cdr lst)))))

(defun _rec-add2(lst end_sum); структурированного
  (cond ((null lst) end_sum)
        ((numberp (car lst)) (_rec-add2 (cdr lst) (+ (car lst)
                                                         end_sum)))
        ((atom (car lst)) (_rec-add2 (cdr lst) end_sum))
        (t (_rec-add2 (cdr lst) (_rec-add2 (car lst) end_sum)))))

(defun rec-add2(lst)
  (_rec-add2 lst 0))
```

1.7 Задание 7

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

Листинг 1.7 – Решение задания №7

```
(defun _recnth(n lst ind)
  (cond ((null lst) nil)
        ((= ind n) (car lst))
        (t (_recnth n (cdr lst) (+ ind 1)))))

(defun recnth(n lst)
  (_recnth n lst 0))
```

1.8 Задание 8

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

Листинг 1.8 – Решение задания №8

```
(defun allodd(lst)
  (cond ((null lst) t)
        ((evenp (car lst)) nil)
        (t (allodd (cdr lst)))))
```

1.9 Задание 9

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

Листинг 1.9 – Решение задания №9

```
(defun get_first_odd(lst)
  (cond ((and (numberp lst) (oddp lst)) lst)
        ((atom lst) nil)
        (t (or (get_first_odd (car lst))
                (get_first_odd (cdr lst))))))
```

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.

Листинг 1.10 – Решение задания №10

```
(defun get_list_square(lst_num)
  (cond ((null lst_num) nil)
        (t (cons (* (car lst_num) (car lst_num)) (get_list_square
                                                         (cdr lst_num))))))
```