



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 №6.1
по дисциплине «Функциональное и логическое
программирование»

Тема: Рекурсивные функции.

Студент: Карпова Е. О.

Группа: ИУ7-62Б

Оценка (баллы): _____

Преподаватели: Толшинская Н. Б., Строганов Ю. В.

Москва — 2023 г.

1. Практическая часть

1.1. Задание №1

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

Листинг 1.1 — Задание №1

```
(defun get_reversed (l r)
  (cond
    ((null l) r)
    (t (get_reversed (cdr l) (cons (car l) r))))
)

(defun my_reverse (l)
  (get_reversed l nil)
)

; ИЛИ

(defun my_reverse (l &optional (r nil))
  (cond
    ((null l) r)
    (t (get_reversed (cdr l) (cons (car l) r))))
)
)
```

1.2. Задание №2

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

Листинг 1.2 — Задание №2

```

(defun get_inside (l)
  (
    cond ((null l) nil)
          ((atom (car l)) (car l))
          (t (get_inside (car l))))
  )
)

(defun get_first_atom (l)
  (
    cond ((null l) nil)
          ((atom (car l)) (get_first_atom (cdr l)))
          (t (get_inside (car l)))
  )
)
))

```

1.3. Задание №3

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

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

Листинг 1.3 — Задание №3

```

(defun mul_numbers1 (l num acc)
  (
    cond ((null l) acc)
          (t (mul_numbers1 (cdr l) num (cons (* (car l) num) acc)))
  )
)

(defun my_mul1 (l num)
  (mul_numbers1 l num nil))

```

Листинг 1.4 — Задание №3

```
(defun mul_numbers2 (l num acc)
  (
    cond ((null l) acc)
    ((numberp (car l)) (mul_numbers2 (cdr l) num
                                     (cons (* (car l) num) acc)))
    ((atom (car l)) (mul_numbers2 (cdr l) num (cons (car l) acc)))
    (t (mul_numbers2 (cdr l) num (cons (mul_numbers2 (car l) num nil)
                                       acc)))
  ))

(defun my_mul2 (l num)
  (mul_numbers2 l num nil)
)
```

1.4. Задание №4

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.

Листинг 1.5 — Задание №4

```
(defun select_between1 (lst l r acc)
  (cond ((null lst) acc)
        ((or (< l (car lst) r) (< r (car lst) l))
         (select_between1 (cdr lst) l r (cons (car lst) acc)))
        (t (select_between1 (cdr lst) l r acc)))
  ))

(defun select_between (lst l r)
  (select_between1 lst l r nil))
```

1.5. Задание №5

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

- одноуровневого смешанного;
- структурированного.

Листинг 1.6 — Задание №5

```
(defun add (lst acc)
  (
    cond ((null lst) acc)
          ((numberp (car lst)) (add (cdr lst) (+ (car lst) acc)))
          (t (add (cdr lst) acc))
  ))

(defun rec_add (lst) (add lst 0))
```

Листинг 1.7 — Задание №5

```
(defun rec_add (lst)
  (
    cond ((null lst) 0)
          ((numberp (car lst)) (+ (car lst) (rec_add (cdr lst))))
          ((atom (car lst)) (rec_add (cdr lst)))
          (t (+ (rec_add (car lst)) (rec_add (cdr lst))))
  ))
```