



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 курсу "Функциональное и логическое программирование"

Тема Рекурсия

Студент Варин Д.В.

Группа ИУ7-66Б

Оценка (баллы) _____

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

Москва — 2022 г.

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

Листинг 1 – Задание 1

```
1 (defun my-reverse (lst &optional res)
2   (if lst
3       (my-reverse (cdr lst) (cons (car lst) res))
4       res))
5
6 (my-reverse '(1 2 3))
```

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

Листинг 2 – Задание 2

```
1 (defun find-first-deep (lst)
2   (cond ((and (listp (car lst)) lst) (caar lst))
3         (lst (find-first-deep (cdr lst)))
4         (T ())))
5
6 (find-first-deep '(1 2 3))
7 (find-first-deep '((1) 2 3))
```

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

Листинг 3 – Задание 3

```
1 (defun find-nums (lst n1 n2 &optional res)
```

```

2  (if lst
3      (find-nums (cdr lst) n1 n2 (if (< n1 (car lst) n2) (cons (car lst)
4          res) res))
5      (reverse res)))
6  (find-nums '(-1 -2 0 3 5 6 10 11) 1 10)

```

4. Напишите рекурсивную функцию, которая умножает на заданное число-аргумент все числа а) все элементы списка – числа б) элементы списка – любые объекты

Листинг 4 – Задание 4

```

1  (defun mult-nums (lst n &optional res)
2      (if lst
3          (mult-nums (cdr lst) n (cons (* (car lst) n) res))
4          (reverse res)))
5
6  (defun mult-nums (lst n &optional res)
7      (if lst
8          (mult-nums (cdr lst) n (cons (cond ((listp (car lst)) (mult-nums
9              (car lst) n))
10              ((numberp (car lst)) (* (car lst) n))
11              (T (car lst)))
12              res))
13          (reverse res)))
14  (mult-nums '(-1 (-2 0 a) 4 5) 10)

```

5. Напишите функцию select-between, которая из списка-аргумента из чисел выбирает те, которые расположены между двумя указанными границами-аргументами и возвращает их в виде списка (сортированного +2 балла)

Листинг 5 – Задание 5

```
1 (defun select-between (lst n1 n2 &optional res)
2   (if lst
3     (select-between (cdr lst) n1 n2 (if (< n1 (car lst) n2) (cons (
4       car lst) res) res))
5     res))
6 (select-between '(0 1 2 3 4) 0 3)
7 (select-between '(0 1 2 3 4) 3 3)
8 (select-between () 3 3)
9 (select-between '(0 1 2) 3 4)
```

6. Написать рекурсивную версию (с именем rec-add) вычисления суммы чисел заданного списка: а) одноуровневого смешанного б) структурированного

Листинг 6 – Задание 6

```
1 (defun rec-add (lst &optional (res 0))
2   (if lst
3     (rec-add (cdr lst) (+ res (car lst)))
4     res))
5
6 (rec-add '(1 2 3))
7
8 (defun rec-add (lst &optional (res 0))
9   (if lst
10     (rec-add (cdr lst) (cond ((listp (car lst)) (rec-add (car lst)
11       res))
12       ((numberp (car lst)) (+ (car lst) res))
```

```

12         (T res)))
13     res))
14
15 (rec-add '(1 2 3))
16 (rec-add '(1 (2 3) (4 5)))

```

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

Листинг 7 – Задание 7

```

1 (defun recnth (n lst &optional (cnt 0))
2   (if (= n cnt)
3       (car lst)
4       (recnth n (cdr lst) (1+ cnt))))
5
6
7 (defun recnth (n lst)
8   (if (= n 0)
9       (car lst)
10      (recnth (- n 1) (cdr lst))))
11
12 (recnth 2 '(0 1 2 3))

```

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

Листинг 8 – Задание 9

```

1 (defun allodd (lst)
2   (if lst
3       (if (oddp (car lst)) (allodd (cdr lst)) ())
4       T))
5
6 (allodd '(1 2 3))
7 (allodd '(1 3))

```

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

Листинг 9 – Задание 9

```
1 (defun _first-odd (lst)
2   (if lst
3     (cond ((listp (car lst)) (or (_first-odd (car lst)) (_first-odd (cdr lst))))
4     ((numberp (car lst)) (if (oddp (car lst)) (car lst) (_first-odd (cdr lst))))))
5
6 (_first-odd '(1 2 3))
7 (_first-odd '(2 4 5))
8 (_first-odd '((2 4) (3 5) 2))
```

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 – Задание 10

```
1 (defun sqr-lst (lst)
2   (if lst (cons (* (car lst) (car lst)) (sqr-lst (cdr lst))))
3
4 (sqr-lst '(-2 3 4))
```