



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	<u>ИУ7-63Б</u>	_____	<u>Лагутин Д. В.</u>
	(Группа)	(Подпись, дата)	(Фамилия И. О.)
Преподаватель		_____	<u>Толпинская Н. Б.</u>
		(Подпись, дата)	(Фамилия И. О.)

Москва, 2023 г.

1 Задание

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

```
1 (defun my-reverse-inner (lst tmp)
2   (cond ((not lst) tmp)
3         ((my-reverse-inner (cdr lst) (cons (car lst) tmp)))))
4
5 (defun my-reverse (lst)
6   (my-reverse-inner lst nil))
```

2 Задание

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

```
1 (defun my-listp (lst)
2   (cond ((not lst) t)
3         ((atom lst) nil)
4         ((my-listp (cdr lst))))))
5
6 (defun get-first-list (lst)
7   (cond ((not lst) nil)
8         ((and (my-listp (car lst)) (/= 0 (length (car lst)))) (car lst))
9         ((get-first-list (cdr lst)))))
```

3 Задание

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

- 1) все элементы списка — числа,
- 2) элементы списка — любые объекты.

```
1 (defun mul-list-inner (lst val tmp)
2   (cond ((not lst) tmp)
3         ((numberp (car lst))
4          (mul-list-inner (cdr lst)
5                           val
6                           (append tmp
7                                   (cons (* val (car lst))
8                                         nil))))
9         ((mul-list-inner (cdr lst)
10                          val
11                          (append tmp
12                                  (cons (car lst)
13                                        nil))))))
14
15 (defun mul-list (lst val) (mul-list-inner lst val nil))
```

4 Задание

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.

```
1 (defun select-between-inner (lst min max tmp)
2   (cond ((not lst) tmp)
3         ((< min (car lst) max) (select-between-inner (cdr lst) min max
4               (cons (car lst) tmp)))
5         ((select-between-inner (cdr lst) min max tmp))))
6 (defun select-between (lst min max)
7   (select-between-inner lst min max nil))
```

5 Задание

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

- 1) одноуровневого смешанного,
- 2) структурированного.

```
1 (defun rec-add-1-inner (lst sum)
2   (cond ((not lst) sum)
3         ((numberp (car lst)) (rec-add-1-inner (cdr lst) (+ sum (car
4           lst)))))
5         ((rec-add-1-inner (cdr lst) sum))))
6 (defun rec-add-1 (lst)
7   (rec-add-1-inner lst 0))
8
9 (defun rec-add-2-inner (lst sum)
10  (cond ((not lst) sum)
11        ((numberp (car lst)) (rec-add-2-inner (cdr lst) (+ sum (car
12          lst)))))
13        ((my-listp (car lst)) (rec-add-2-inner (cdr lst) (+ sum
14          (rec-add-2-inner (car lst) 0)))))
15        ((rec-add-2-inner (cdr lst) sum))))
16 (defun rec-add-2 (lst)
17   (rec-add-2-inner lst 0))
```