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 Криков А.В.

Группа ИУ7-63Б

Оценка (баллы) _____

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

Москва — 2022 г.

Практические задания

Задание 1

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

```
(defun my-reverse (lst &optional res)
  (if lst
    (my-reverse (cdr lst) (cons (car lst) res))
    res))
```

Задание 2

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

```
(defun find-first-deep (lst)
  (cond ((and (listp (car lst)) lst) (caar lst))
    (lst (find-first-deep (cdr lst)))
    (T ())))
```

Задание 3

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

```
(defun find-nums (lst n1 n2 &optional res)
  (if lst
    (find-nums (cdr lst) n1 n2 (if (< n1 (car lst) n2) (cons (car lst) res)
       res))
    (reverse res)))
```

Задание 4

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.

```
(defun mult-nums (lst n &optional res)
  (if lst
    (mult-nums (cdr lst) n (cons (* (car lst) n) res))
    (reverse res)))

(defun mult-nums (lst n &optional res)
  (if lst
    (mult-nums (cdr lst) n (cons (cond ((listp (car lst)) (mult-nums (car
       lst) n))
            ((numberp (car lst)) (* (car lst) n))
            (T (car lst)))
          res))
    (reverse res)))
```

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)).

```
(defun select-between (lst n1 n2 &optional res)
  (if lst
    (select-between (cdr lst) n1 n2 (if (< n1 (car lst) n2) (cons (car lst)
       res) res))
    res))

(defun select-between-sorted (lst n1 n2)
    (sort (select-between lst n1 n2) #'<))
```

Задание 6

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

1. одноуровнего смешанного;

2. структурированного.

```
(defun rec-add (lst &optional (res 0))
  (if lst
    (rec-add (cdr lst) (+ res (car lst)))
    res))
```

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

Задание 7

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

```
1 (defun recnth (n lst)
2   (if (= n 0) (car lst) (recnth (- n 1) (cdr lst))))
```

Задание 8

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

```
1 (defun allodd (lst)
2     (if lst (if (oddp (car lst)) (allodd (cdr lst)) ()) T))
```

Задание 9

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

```
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)))))))
```

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.

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