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»

Студент Маслова Марина Дмитриевна

Группа ИУ7-63Б

Оценка (баллы) _____

Преподаватель Толпинская Наталья Борисовна

Преподаватель Строганов Юрий Владимирович

2022 г.

1 Практические задания

1.1 Задание №1

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

```
(defun move-to (lst res)
    (cond ((null lst) res)
          (T (move-to (cdr lst) (cons (car lst) res)))))

(defun my-reverse (lst)
    (move-to lst ()))
```

1.2 Задание №2

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

```
(defun get-first-list-el (lst)
    (cond ((null lst) Nil)
          ((and (car lst) (listp (car lst))) (caar lst))
          (T (get-list (cdr lst)))))
```

1.3 Задание №3

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

```
(defun sba (lst begin end res)
    (cond ((null lst) res)
          ((and (numberp (car lst))
                (or (< begin (car lst) end) (< end (car lst) begin)))
               (sba (cdr lst) begin end (cons (car lst) res)))
          ((atom (car lst)) (sba (cdr lst) begin end res))
          (T (sba (cdr lst) begin end (sba (car lst) begin end res)))))

(defun select-between-all (lst begin end)
    (sba lst begin end ()))
```

1.4 Задание №4

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

1. все элементы списка — числа,

```
(defun mul-nums-res (lst num res)
    (cond ((null lst) res)
          (T (mul-nums-res (cdr lst) num (cons (* (car lst) num) res)))))

(defun mul-nums (lst num)
    (mul-nums-res lst num ()))
```

2. элементы списка — любые объекты.

```
(defun mn (lst num res)
    (cond ((null lst) res)
          ((numberp (car lst))
              (mn (cdr lst) num (cons (* (car lst) num) res)))
          ((atom (car lst)) (mn (cdr lst) num (cons (car lst) res)))
          (T (mn (cdr lst) num (cons (mn (car lst) num ()) res)))))

(defun mul-num (lst num)
    (mn lst num ()))
```

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.

```
(defun sb (lst begin end res)
    (cond ((null lst) res)
          ((or (< begin (car lst) end) (< end (car lst) begin))
               (sb (cdr lst) begin end (cons (car lst) res)))
          (T (sb (cdr lst) begin end res))))

(defun select-between (lst begin end)
    (sb lst begin end ()))
```

1.6 Задание №6

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

1. одноуровневого смешанного;

```
(defun ra (lst res)
    (cond ((null lst) res)
          ((numberp (car lst)) (ra (cdr lst) (+ (car lst) res)))
          (T (ra (cdr lst) res))))

(defun rec-add (lst)
    (ra lst 0))
```

2. структурированного.

```
(defun ra (lst res)
    (cond ((null lst) res)
          ((numberp (car lst)) (ra (cdr lst) (+ (car lst) res)))
          ((atom (car lst)) (ra (cdr lst) res))
          (T (ra (cdr lst) (ra (car lst) res)))))

(defun rec-add (lst)
    (ra lst 0))
```

1.7 Задание №7

Написать рекурсивную версию с имененем recnth функции nth.

```
(defun rec-num-nth (n lst cur-ind)
    (cond ((null lst) Nil)
          ((= cur-ind n) (car lst))
          (T (rec-num-nth n (cdr lst) (+ cur-ind 1)))))

(defun recnth (n lst)
    (rec-num-nth n lst 0))
```

1.8 Задание №8

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

```
(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.

```
(defun first-odd (lst)
    (cond ((null lst) Nil)
          ((and (numberp (car lst)) (oddp (car lst))) (car lst))
          ((atom (car lst)) (first-odd (cdr lst)))
          (T (or (first-odd (car lst)) (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.

```
(defun squares (lst)
    (cond ((null lst) Nil)
          (T (cons (* (car lst) (car lst)) (squares (cdr lst))))))
```