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 №9 по курсу «Функциональное и логическое программирование»

Тема: Использование функционалов и рекурсии

Студент: Сорокин А. П. ИУ7-66Б

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

2021 г.

6.4. Функция, которая выбирает из заданного списка только те числа, которые находятся между двумя заданными границами.

```
(defun is-between (x a b)
    (and (<= a x) (<= x b))
)

(defun sel-between-r (lst a b res)
    (if (null lst) res
        (sel-between-r (cdr lst) a b
            (cond
                ((listp (car lst)) (sel-between-r (car lst) a b res))
                ((and (numberp (car lst)) (is-between (car lst) a b))
                    (cons (car lst) res))
                (T res)
            )
        )
    )
)

(defun select-between (lst a b)
    (cond
        ((null lst) Nil)
        ((> a b) (sel-between-r lst b a Nil))
        (T (sel-between-r lst a b Nil))
    )
)

(defun sel-between-f (lst a b res)
(reduce #'(lambda (xlst x)
            (cond
                ((listp x) (sel-between-f x a b xlst))
                ((and (numberp x) (is-between x a b)) (cons x xlst))
                (T xlst)
            )
        )
     lst
        :initial-value res
    )
)

(defun select-between-f (lst a b)
    (cond
        ((null lst) Nil)
        ((> a b) (sel-between-f lst b a Nil))
        (T (sel-between-f lst a b Nil))
    )
)
```

6,5. Функция, вычисляющая декартово произведение двух своих списков-аргументов.

```
(defun add-to-set (lst el)
    (cond
        ((null lst) (cons el Nil))
        ((equalp el (car lst)) lst)
        (t (cons (car lst) (add-to-set (cdr lst) el)))
    )
)

(defun lst-to-set (lst)
    (reduce #'add-to-set lst :initial-value Nil)
)

(defun decart (a b)
     (mapcan    #'(lambda (x)
            (mapcar  #'(lambda (y) (cons x y)
                ) (lst-to-set b)
            )
          ) (lst-to-set a)
     )
)
```

6.7. Функция, которая вычисляет сумму длин всех элементов list-of-list (список, состоящий из списков).

```
(defun my-length (lst)
    (reduce #'(lambda (x y) (+ x 1))
        lst
        :initial-value 0
    )
)


(defun sum-len (list-of-list)
    (reduce #'+
        (mapcar #'(lambda (x)
                (if (listp x) (my-length x) 1)
            )
            list-of-list
        )
    )
)
```

Ответы на вопросы

1. Классификация рекурсивных функций.

Классификация №1:

- простая рекурсия: единственный вызов в теле функции;
- рекурсия первого порядка: вызовов несколько в теле функции;
- взаимная рекурсия: две или несколько функций вызывают друг друга.

Классификация №2:

- хвостовая рекурсия: все необходимые действия выполнены до вызова рекурсии, возвращает результат;
- дополняемая рекурсия: использует дополнительную функцию, которая используется не в аргументе вызова, а вне её;
    - частный случай – cons-дополняемая рекурсия.