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»

Лабораторная работа №5 по дисциплине "Функциональное и логическое программирование"

Тема Использование функционалов

Студент Светличная А.А.

Группа ИУ7-63Б

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

Москва — 2023 г.

1 Практические задания

1.1 Задание №1

Напишите функцию, которая уменьшает на 10 все числа из списка-аргумента этой функции, проходя по верхнему уровню списковых ячеек. (*Список смешанный структурированный)

Листинг 1.1 – Выполнение задания №1

```
(defun my-add(lst)
    (mapcar
        #'(lambda (x)
            (cond
                ((numberp x) (- x 10))
                (T x))
            )
        lst
    )
)
```

1.2 Задание №2

Написать функцию которая получает как аргумент список чисел, а возвращает список квадратов этих чисел в том же порядке.

Листинг 1.2 – Выполнение задания №2

```
(defun my-square(lst)
    mapcar #'(lambda(x) (* x x) lst))
```

1.3 Задание №3

Напишите функцию, которая умножает на заданное число-аргумент все числа из заданного списка-аргумента, когда а) все элементы списка —

числа, б) элементы списка — любые объекты.

Листинг 1.3 – Выполнение задания №3

```
(defun my-multiply(lst)
    mapcar #'(lambda(x) (* x num) lst))

(defun my-multiply(lst num)
    (mapcar
        #'(lambda (x)
            (cond
                ((numberp x) (* x num))
                ((listp x) (multiply-by x num))
                (T x))
            )
        lst
    )
)
```

1.4 Задание №4

Написать функцию, которая по своему списку-аргументу lst определяет является ли он палиндромом (то есть равны ли lst и (reverse lst)), для одноуровнего смешанного списка.

Листинг 1.4 – Выполнение задания №4

```
(defun compare (lst1 lst2)
    (reduce #'(lambda (x y) (and x y))
        (mapcar #'eql lst1 lst2)))

(defun palindrome (lst)
    (apply #'(lambda (lst1 lst2) (compare lst1 lst2))
        (list lst (reverse lst))))
```

1.5 Задание №5

Используя функционалы, написать предикат set-equal, который возвращает t, если два его множества-аргумента (одноуровневые списки) содержат одни и те же элементы, порядок которых не имеет значения.

Листинг 1.5 – Выполнение задания №5

```
(defun set-include-set(lst1 lst2)
   (reduce #'(lambda(x y) (and x y))
           (mapcar #'(lambda (el) (member el lst2)) lst1)))

(defun compare-set(lst1 lst2)
    (if (and (set-include-set lst1 lst2)
             (set-include-set lst2 lst1) ) T NIL ))
```

1.6 Задание №6

Напишите функцию, select-between, которая из списка-аргумента, содержащего только числа, выбирает только те, которые расположены между двумя указанными числами – границами-аргументами и возвращает их в виде списка.

Листинг 1.6 – Выполнение задания №6

```
(defun select-between(lst num1 num2)
    (remove-if #'(lambda(x)
                        (not (or (and (> x num1) (< x num2))
                                 (and (< x num1) (> x num2))
                             )
                        )
                 )
    )
)
```

1.7 Задание №7

Написать функцию, вычисляющую декартово произведение двух своих списковаргументов. (Напомним, что A x B это множество всевозможных пар (a b), где a принадлежит A, принадлежит B).

Листинг 1.7 – Условие задания №7

```
(defun decart-mult (lst1 lst2)
    (mapcan #'(lambda (x)
        (mapcar #'(lambda (y) (list x y))
            lst2)) lst1))
```

1.8 Задание №8

Почему так реализовано reduce, в чем причина?

Листинг 1.8 – Условие задания №8

```
(reduce #'+ ()) -> 0
(reduce #'* ()) -> 1
```

Если подпоследовательность пуста и начальное значение не задано, то функция вызывается с нулевыми аргументами (для сложения — ноль, для умножения — единица).

1.9 Задание №9

Пусть list-of-list список, состоящий из списков. Написать функцию, которая вычисляет сумму длин всех элементов list-of-list (количество атомов), т.е. например для аргумента `((1 2) (3 4)) -> 4`.

Листинг 1.9 – Выполнение задания №9

```
(defun len-list-of-list(lst)
    (reduce #'+
            (mapcar #'(lambda (x)
                              (cond
                                  ((listp x) (len-list-of-list x))
                                  (T 1)
                              )
                      )
                    lst
            )
    )
 )
```