## Задание 1

Напишите функцию, которая уменьшает на 10 все числа из списка-аргумента этой функции.

**(defun** f1**(**lst**)**

**(mapcar** #**'(lambda** **(**x**)**

**(if** **(numberp** x**)**

**(-** x 10**)**

x**))** lst**))**

## Задание 2

Напишите функцию, которая умножает на заданное число-аргумент все числаиз заданного списка-аргумента, когдаa) все элементы списка --- числа,б) элементы списка -- любые объекты.

;numbers

**(defun** mul-lst1**(**n lst**)**

**(mapcar** #**'(lambda** **(**x**)** **(\*** x n**))** lst**))**

;objcets

**(defun** mul-lst2**(**n lst**)**

**(mapcar** #**'(lambda** **(**x**)**

**(if** **(numberp** x**)**

**(\*** x n**)**

x**))** lst**))**

## Задание 3

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

**(defun** my-reverse **(**lst**)**

**(let** **((**res **nil))**

**(mapcar** #**'(lambda** **(**el**)**

**(setf** res **(cons** el res**)))** lst**)**

res**))**

**(defun** is-palindrom **(**lst**)**

**(equal** lst **(**my-reverse lst**)))**

## Задание 4

Написать предикат set-equal, который возвращает t, если два его множества-

аргумента содержат одни и те же элементы, порядок которых не имеет значения.

**(defun** set1-in-set2 **(**set1 set2**)**

**(if** **(member** NIL

**(mapcar** #**'(lambda** **(**el**)**

**(if** **(member** el set2**)** T**))**

set1**))**

NIL

T**))**

**(defun** set-equal**(**set1 set2**)**

**(and** **(**set1-in-set2 set1 set2**)**

**(**set1-in-set2 set2 set1**)))**

## Задание 5

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

**(defun** get-squares**(**lst**)**

**(mapcar** #**'(lambda** **(**el**)** **(\*** el el**))** lst**))**

## Задание 6

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 a b**)**

**(**sort

**(mapcan** #**'(lambda** **(**el**)**

**(cond** **((or** **(<=** a el b**)** **(<=** b el a**))** **`(**,el**))**

**(**T NIL**)))** lst**)** #'<**))**

## Задание 7

Написать функцию, вычисляющую декартово произведение двух своих списков-аргументов. ( Напомним, что А х В это множество всевозможных пар (a b), где а принадлежит А, принадлежит В.)

**(defun** decart **(**lstX lstY**)**

**(mapcan** #**'(lambda** **(**x**)**

**(mapcar** #**'(lambda** **(**y**)**

**(list** x y**))** lstY**))**

lstX**))**

## Задание 8

Почему так реализовано reduce, в чем причина?

(reduce #'+ '(0)) -> 0

(reduce #'+ ()) -> 0

**(**reduce #'+ **'(**0**))** == **(+** 0**)** => 0

**(**reduce #'+ **())** == **(+)** => 0

## Задание 9

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

**(defun** get-len**(**lst**)**

**(apply** #'+ **(mapcar** #'length lst**)))**