



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 №6 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

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

```
1 (defun minus10 (lst)
2   (mapcar #'(lambda (x)
3               (cond ((numberp x) (- x 10))
4                     ((atom x) x)
5                     (T (minus10 x)))) lst))
```

1.2 Задание №2

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

1. все элементы списка — числа,

```
1 (defun mul-num (lst num)
2   (mapcar #'(lambda (x) (* x num)) lst))
```

2. элементы списка — любые объекты.

```
1 (defun mul-num (lst num)
2   (mapcar #'(lambda (x)
3               (cond ((numberp x) (* x num))
4                     ((atom x) x)
5                     (T (mul-num x num)))) lst))
```

1.3 Задание №3

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

```
1 (defun my-reverse (lst)
2   (reduce #'(lambda (x y) (cons y x)) (cons nil lst)))
3
4 (defun is-palindrome (lst)
5   (equal lst (my-reverse lst)))
```

1.4 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.

```
1 (defun is-in (el st)
2   (reduce #'(lambda (x y) (or x y))
3     (mapcar #'(lambda (x) (equal x el)) st) :initial-value Nil))
4
5 (defun is-subset (st1 st2)
6   (reduce #'(lambda (x y) (and x y))
7     (mapcar #'(lambda (x) (is-in x st2)) st1) :initial-value T))
8
9 (defun set-equal (st1 st2)
10  (and (is-subset st1 st2) (is-subset st2 st1)))
```

1.5 Задание №5

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

```
1 (defun squares (lst)
2   (mapcar #'(lambda (x) (* x x)) lst))
```

1.6 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.

```
1 (defun select-between (lst begin end)
2   (remove-if-not #'(lambda (x) (< begin x end)) lst))
```

1.7 Задание №7

Написать функцию, вычисляющую декартово произведение двух своих списков-аргументов.

```

1 (defun cartesian (lst1 lst2)
2   (mapcar #'(lambda (el1)
3             (mapcar #'(lambda (el2) (list el1 el2)) lst2)) lst1))

```

1.8 Задание №8

Почему так реализовано reduce, в чем причина?

```

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

```

Функция проверяет список-аргумент. Если он содержит только один элемент и initial-value не задано, то возвращается элемент и функция не вызывается. Если список-аргумент пустой и initial-value не задано, то функция вызывается с нулевым количеством параметров.

1.9 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.

```

1 (defun get-length (lst)
2   (reduce #' + (mapcar #'(lambda (x) (cond ((atom x) 1)
3                                             (T (get-length x)))) lst))

```