



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"

Группа ИУ7-63Б (ИУ7и-676) _____

Студент Тэмуужин Я. _____

Преподаватель Толпинская Н.Б. _____

Преподаватель Строганов Ю. В. _____

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

```
(defun subtract-list-10 (lst) (mapcar #'(lambda (x) (- x 10)) lst))
```

2. Напишите функцию, которая умножает на заданное число-аргумент все числа из заданного списка-аргумента, когда а) все элементы списка — числа, (1 2 3) 10 -> (10 20 30)

```
(defun mult-randlist (lst num)
  (mapcar #'(lambda (x) (if (numberp x) (* x num) x)) lst))
```

б) элементы списка — любые объекты. (a 1 2 sdf 3) 10 -> (a 10 20 sdf 30)

```
(defun mult-numlist (num lst)
  (mapcar #'(lambda (x) (* num x)) lst))
```

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 res)
  (cond ((null lst) res)
        (T (my-reverse (cdr lst) (cons (car lst) res))) ) )

(defun is-palindrom (lst)
  (funcall #'(lambda (l1 l2) (equal l1 l2)) lst (my-reverse lst ())))
```

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 2 1 3 2) (3 2 1 2) T

```
(defun my-member (el set2)
  (cond ((null set2) nil)
        ((equal el (car set2)) T)
        (T (my-member el (cdr set2)))))

(defun my-subsetp (set1 set2)
  (cond ((my-member (car set1) set2)
        (if (not (null (cdr set1)))
            (my-subsetp (cdr set1) set2) T)) ))

(defun set-equal-1 (set1 set2)
  (and (my-subsetp set1 set2)
       (my-subsetp set2 set1)))

(defun set-equal (set1 set2)
  (not (or (mapcan #'(lambda (x1) (if (null x1) (not x1) nil))
                  (mapcar #'(lambda (x)
                              (my-member x set2))
                          set1))
          (mapcan #'(lambda (x2) (if (null x2) (not x2) nil))
                  (mapcar #'(lambda (y)
                              (my-member y set1))
                          set2)))))
```

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

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

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 is-between (x b1 b2)
  (or (and (>= x b1) (<= x b2))
      (and (<= x b1) (>= x b2))))

(defun select-between (lst b1 b2)
  (mapcan #'(lambda (x)
              (if (is-between x b1 b2) (list x) nil))
          lst ))
```

7. Написать функцию, вычисляющую декартово произведение двух своих списков- аргументов. (Напомним, что $A \times B$ это множество всевозможных пар (a, b) , где a принадлежит A , принадлежит B .)

```
a (2 3)
b (4 5 6)
((2 4) (2 5) (2 6))
(3 4) (3 5) (3 6))
```

```
(defun decart (lst1 lst2)
  (mapcar #'(lambda (x)
              (mapcar #'(lambda (y) (cons x y))
                      lst2))
          lst1))
```

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 $((1\ 2)\ (3\ 4)) \rightarrow 4$. $((1\ 2\ 3)\ (3\ 4\ 6\ 7)) \rightarrow 7$.

```
(defun lofl-len (lofl)
  (apply #'+
        (mapcan #'(lambda (x)
                    (if (listp x) (list (length x))))
                lofl)))
```