



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический
университет имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 8

Дисциплина	Функциональное и логическое программирование
Студент	Сиденко А.Г.
Группа	ИУ7-63Б
Преподаватель	Толпинская Н.Б.

Москва, 2020 г.

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

```
1 (defun palindrom (lst)
2   (equalp lst (reverse lst)) )
```

2. Напишите функцию `swap-first-last`, которая переставляет в списке-аргументе первый и последний элементы.

```
1 (defun last_elem (lst)
2   (if (NULL (cadr lst)) (car lst) (last_elem (cdr lst))))
3
4 (defun centr (lst)
5   (if (NULL (cadr lst)) (cons (car lst) (centr (cdr lst))))
6
7 (defun swap_first_last (lst)
8   ‘(, (last_elem lst) ,@(centr (cdr lst)) ,(car lst)) )
```

3. Напишите функцию `swap-two-ellement`, которая переставляет в списке-аргументе два указанных своими порядковыми номерами элемента в этом списке.

```
1 (defun swap_two_ellement (lst i j)
2   (cond ((= i j) lst)
3         ((> i j) (swap_two_ellement lst j i))
4         (t
5          ‘(, @(subseq lst 0 i), (nth j lst), @(subseq lst (+ i 1) j)
6            , (nth i lst) ,@(subseq lst (+ j 1))))))
```

4. Напишите две функции, `swap-to-left` и `swap-to-right`, которые производят круговую перестановку в списке-аргументе влево и вправо, соответственно.

```
1 (defun swap_to_left (lst k)
2   (cond ((= k 0) lst)
3         ((> k (length lst))
4          (swap_to_left lst (rem k (length lst))))
5         (t
6          ‘(, @(subseq lst k) ,@(subseq lst 0 k)))) )
7
8 (defun swap_to_right (lst k)
9   (cond ((= k 0) lst)
10         ((> k (length lst))
11          (swap_to_right lst (rem k (length lst))))
```

```

12      (t
13        '(@ (subseq lst (- (length lst) k))
14          ,@ (subseq lst 0 (- (length lst) k)))) )

```

5. Напишите функцию, которая умножает на заданное число-аргумент все числа из заданного списка-аргумента, когда все элементы списка — числа или элементы списка — любые объекты.

```

1 (defun multiplication (lst k)
2   (mapcar #'(lambda (x) (if (numberp x) (* x k) 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).

```

1 (defun select_between_nil (lst a b)
2   (mapcar #'(lambda (x) (if (and (> x a) (< x b)) x Nil)) lst))
3
4 (defun remove_null (lst)
5   (cond ((null lst) nil)
6         ((null (car lst)) (remove_null (cdr lst)))
7         (t (cons (car lst) (remove_null (cdr lst))))))
8
9 (defun select_between (lst a b)
10  (remove_null (select_between_nil lst a b)) )

```