

*Государственное образовательное учреждение высшего профессионального
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»

Отчет по лабораторной работе №8

по курсу ФиЛП

Студент

(Подпись, дата) Якубаускайте М. А.

Преподаватель

(Подпись, дата) Толпинская Н.Б.

Москва 2019

Практическая часть

5.1

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) (equal lst (reverse lst)) )
```

5.4

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

```
1 (defun swap-first-last (lst)
2   (append (last lst) (cdr (butlast lst)) (cons (car lst) nil))
3 )
```

5.5

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

```
1 (defun swap-two-element (lst f s)
2   (let ((temp (nth f lst)))
3     (setf (nth f lst) (nth s lst))
4     (setf (nth s lst) temp))
5   lst
6 )
```

5.6

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

```
1 (defun swap-to-left (lst)
2   (append
3     (cdr lst)
4     (cons (first lst) nil)
5   )
6 )
7 (defun swap-to-right (lst)
8   (append
9     (last lst)
10    (butlast lst)
11  )
12 )
```

Сдвигает на k элементов влево

```
1 (defun << (l n)
2   (cond
3     ((minusp n) l)
4     ((zerop n) l)
5     (t (<< (append (cdr l) (list (car l))) (- n 1)))))
```

Сдвигает на k элементов вправо

```
1 (defun >> (l n)
2   (cond
3     ((minusp n) l)
4     ((zerop n) l)
5     (t (>> (append (list (car (reverse l))) (reverse (cdr (reverse
      l))))) (- n 1))))
```

5.7

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

```
1 (defun multiply-all (lst mul)
2   (mapcar #'(lambda (x) (* x mul))
3     lst
4   )
5 )
```

б) элементы списка - любые объекты.

```
1 (defun multiply-all (lst mul)
2   (mapcar #'(lambda (x) (* x mul))
3     lst
4   )
5 )
```

5.8

Напишите функцию select-between, которая из списка-аргумента, содержащего только числа, выбирает только те, которые расположены между двумя указанными границами-аргументами.

```
1 (defun select-between (lst left right)
2   (remove-if #'(lambda (x) (or (< x left) (> x right)))
3     lst)
4 )
```