



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.
Тема	Работа с функционалами
Студент	Степанов А. О.
Группа	ИУ7-63Б
Оценка (баллы)	
Преподаватель	Толпинская Н.Б.

Москва, 2020 г.

ЗАДАНИЕ 1

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

```
1 (defun is_palindrom (lst)
2   (equal lst (reverse lst)))
3 )
4
5 (is_palindrom "abccba") ;;; T
6 (is_palindrom "abcgba") ;;; Nil
```

ЗАДАНИЕ 2

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

```
1 (defun swap-first-last (lst)
2   (cond
3     ((null lst) Nil)
4     ((eql (length lst) 1) lst)
5     (T
6       (append
7         (last lst)
8         (mapcar #'(lambda (el _) el) (cdr lst) (cddr lst))
9         (list (first lst))
10      )
11    )
12  )
13 )
14
15 (swap-first-last ()) ;;; Nil
16 (swap-first-last '(1)) ;;; (1)
17 (swap-first-last '(1 2)) ;;; (2 1)
18 (swap-first-last '(1 2 3 4 5)) ;;; (5 2 3 4 1)
```

ЗАДАНИЕ 3

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

```
1 (defun swap-two-elements (a b lst)
2   (cond
3     ((eql a b) lst)
4     ((> a b) (swap-two-elements b a lst))
```

```

5      (T
6      (append
7      (subseq lst 0 a)
8      (list (nth b lst))
9      (subseq lst (+ a 1) b)
10     (list (nth a lst))
11     (subseq lst (+ b 1))
12     )
13   )
14 )
15 )
16
17 (swap-two-elements '2 '4 '(1 2 3 4 5)) ;;; (1 2 5 4 3)
18 (swap-two-elements '1 '4 '(1 2 3 4 5)) ;;; (1 5 3 4 2)
19 (swap-two-elements '3 '2 '(1 2 3 4 5)) ;;; (1 2 4 3 5)

```

ЗАДАНИЕ 4

Напишите две функции, `swap-to-left` и `swap-to-right`, которые производят круговую перестановку в списке-аргументе влево и вправо, соответственно (на `k` позиций).

```

1 (defun swap-to-left-one (lst)
2   (append (mapcar #'(lambda (el _) el) (cdr lst) lst) (list (car lst)))
3 )
4
5 (defun swap-to-left (lst k)
6   (cond
7     ((<= k 0) lst)
8     (T (swap-to-left (swap-to-left-one lst) (- k 1)))
9   )
10 )
11
12 (swap-to-left '(1 2 3 4 5) 2) ;;; (3 4 5 1 2)
13 (swap-to-left '(1 2 3 4 5) 5) ;;; (1 2 3 4 5)
14
15 (defun swap-to-right-one (lst)
16   (append (last lst) (mapcar #'(lambda (el _) el) lst (cdr lst)))
17 )
18
19 (defun swap-to-right (lst k)
20   (cond
21     ((<= k 0) lst)

```

```

22      (T (swap-to-right (swap-to-right-one lst) (- k 1)))
23    )
24  )
25
26  (swap-to-right '(1 2 3 4 5) 2) ;;; (4 5 1 2 3)
27  (swap-to-right '(1 2 3 4 5) 5) ;;; (1 2 3 4 5)

```

ЗАДАНИЕ 5

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

- а) все элементы списка – числа,
- б) элементы списка – любые объекты.

```

1  (defun number_multiplication_car (lst n)
2    (mapcar #'(lambda (el) (* el n)) lst)
3  )
4
5  (number_multiplication_car '(1 2 3 4) '5) ;;; (5 10 15 20)
6
7  (defun multiplication_car (lst n)
8    (mapcar #'(lambda (el) (if (numberp el) (* el n) el)) lst)
9  )
10
11 (multiplication_car '(1 2 3 4) '5) ;;; (5 10 15 20)
12 (multiplication_car '(a 2 b 4) '5) ;;; (A 10 B 20)

```

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).

```

1  (defun select_between (a b lst)
2    (sort (reduce #'
3      (lambda (lst el)
4        (append
5          (if (numberp lst)
6            (if (and (<= a lst) (>= b lst))

```

```

7             (list lst)
8         )
9         lst
10    )
11    (if (and (<= a el) (>= b el))
12        (list el)
13    )
14 )
15 ) lst
16 ) #'<)
17 )
18
19 (select_between '1 '10 '(40 5 2 -6 3 1 8)) ;;; (1 2 3 5 8)
20 (select_between '-10 '5 '(-6 -2 -11 8 0 1 -10 -7 100 20 -4))
21 ;;; (-10 -7 -6 -4 -2 0 1)

```