



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 №5 по курсу «Функциональное и логическое программирование»

Тема Использование управляющих структур, работа со списками

Студент Климов И.С.

Группа ИУ7-62Б

Оценка (баллы) _____

Преподаватели Толпинская Н.Б., Строганов Ю.В.

Москва — 2022 г.

Задание 1

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

Решение

```
(defun palindrome (lst)
  (cond ((null (listp lst)) NIL)
        (T (equal lst (reverse lst))))))
```

Задание 2

Написать предикат `set-equal`, который возвращает `t`, если два его множества-аргумента содержат одни и те же элементы, порядок которых не имеет значения.

Решение

```
(defun set_equal (lst1 lst2)
  (and (listp lst1) (listp lst2)
        (subsetp lst1 lst2) (subsetp lst2 lst1)))
```

Задание 3

Напишите свои необходимые функции, которые обрабатывают таблицу из 4-х точечных пар: (страна . столица), и возвращают по стране – столицу, а по столице – страну .

Решение

```
(defun capital_by_country (table country)
  (cond ((null table) NIL)
        ((equal (caar table) country) (cdar table))
        (T (capital_by_country (cdr table) country))))
```

```
(defun country_by_capital (table capital)
  (cond ((null table) NIL)
        ((equal (cdar table) capital) (caar table))
        (T (country_by_capital (cdr table) capital))))
```

Задание 4

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

Решение

```
(defun swap_first_last (lst)
  (cond ((< (length lst) 2) lst)
        (T (append (last lst) (butlast (cdr lst)) (list (car lst))))))
```

Задание 5

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

Решение

```
(defun butfirst (lst num)
  (car (remove NIL (maplist #'(lambda (x)
                                (if (= (length x) (- (length lst) num))
                                    x NIL)) lst))))

(defun swap_two_elements_help (lst i j)
  (cond ((or (< (length lst) 2) (= i j)) lst)
        (T (append (butlast lst (- (length lst) i))
                    (list (nth j lst))
                    (butlast (butfirst lst (+ i 1)) (- (length lst) j))
                    (list (nth i lst))
                    (butfirst lst (+ j 1))))))
```

```
(defun swap_two_elements (lst i j)
  (cond ((< j i) (swap_two_elements_help lst j i))
        (T (swap_two_elements_help lst i j))))
```

Задание 6

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

Решение

```
(defun swap_to_left (lst)
  (cond ((< (length lst) 2) lst)
        (T (append (butfirst lst 1) (list (car lst))))))

(defun swap_to_right (lst)
  (cond ((< (length lst) 2) lst)
        (T (append (last lst) (butlast lst)))))
```

Задание 7

Напишите функцию, которая добавляет к множеству двухэлементных списков новый двухэлементный список, если его там нет.

Решение

```
(defun add_two_element_list (lst new_pair)
  (cond ((null (listp lst)) NIL)
        ((null (notany #'(lambda (pair) (equal pair new_pair)) lst)) NIL)
        (T (nconc lst (list new_pair)))))
```

Задание 8

Напишите функцию, которая умножает на заданное число-аргумент первый числовой элемент списка из заданного 3-х элементного списка-аргумента, когда

- все элементы списка – числа,
- элементы списка – любые объекты.

Решение

```
(defun mul_first_number_v1 (lst k)
  (append (list (* (car lst) k)) (cdr lst)))

(defun mul_first_number_v2 (lst k)
  (let ((stop 0))
    (mapcar #'(lambda (x) (if (and (= stop 0) (numberp x))
                              (and (setf stop 1) (* x k))
                              x)) lst)))
```

Задание 9

Напишите функцию, `select-between`, которая из списка-аргумента из 5 чисел выбирает только те, которые расположены между двумя указанными границами-аргументами и возвращает их в виде списка (упорядоченного по возрастанию списка чисел (+ 2 балла)).

Решение

```
(defun select_between_help (lst left right)
  (sort (remove NIL (mapcar #'(lambda (num)
                                (if (and (> num left) (< num right)) num))
    lst)) #'<))

(defun select_between (lst left right)
  (cond ((< right left) (select_between_help lst right left))
        (T (select_between_help lst left right))))
```

