|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Лабораторная работа №8*

*По предмету: «Функциональное и логическое программирование»*

**Тема:** Определение функций пользователя.

Студент: Юмаев А.Р.,

Группа: ИУ7-65Б

Преподаватель: Толпинская Н. Б.

Москва, 2020 г.

№1

Является ли список палиндромом?

(defun func(lst len)

(cond ((<= len 0) t)

((and (equalp (car lst) (last lst)) (func (cdr (butlast lst)) (- len 2))) t)

)

)

№4

Написать функцию swap-first-last:

(defun swap-first-last (x)

(append (last x) (cdr (butlast x)) (cons (car x) nil))

)

№5

Написать функции swap-to-element:

(defun swap-two-element (lst f s)

(let ((temp (nth f lst)))

(setf (nth f lst) (nth s lst ))

(setf (nth s lst) temp))

lst

)

№6

Написать функции swap-to-left и swap-to-right:

(defun swap-to-left (lst)

(append (cdr lst)

(cons (first lst) nil)

)

)

(defun swap-to-right (lst)

(append (last lst)

(butlast lst)

)

)

№7

Функция, которая умножает на заданное число все числа из списка, когда:  
 а) все элементы списка – числа

б) элементы списка – любые объекты

а) (defun mltpl (lst mul)

(mapcar #'(lambda (x) (\* x mul)) lst

)

)

б) (defun multiply-all (lst mul)

(mapcar #'(lambda (x)

(cond ((numbperp x) (\* x mul))

((listp x) (multiply-all x mul))

(t x)

)

)

lst) )

№8

Написать функцию select-between, которая из списка-аргумента выбирает те числа, которые расположены между двумя границами-аргументами (возвращает результат в виде списка):

(defun select-between (lst left right)

(remove-if #'(lambda (x) (or (< x left) (> x right)))

lst))