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_

Дисциплина Функциональные и логические языки программирования

Студент Козырев Марк

Группа ИУ7-63

Оценка (баллы) _____

Преподаватель _____

Москва.
2020 г.

1. Написать функцию, которая по своему списку-аргументу lst определяет является ли он полиндромом

```
(defun 1_polindrom (lst1 lst2)
	(check_t
		(mapcar #'equalp
			lst1
			(reverse lst2)
		)
	)
)
```

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

```
(defun 4_swap_first_last (lst)
	(append
		(last lst)
		(butlast (cdr lst))
		(list (car lst))
	)
)
```

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

```
(defun 5_swap_two_ellement (n1 n2 lst)
	(append
		(rec_mid 0 n1 lst 0)
		(list (nth n2 lst))
		(rec_mid (+ n1 1) n2 lst 0)
		(list (nth n1 lst))
		(rec_mid (+ n2 1) (length lst) lst 0)
	)
)
(defun rec_mid (n1 n2 lst current)
	(cond
		((null lst) nil)
		((= n2 current) nil)
		((<= n1 current) (append (list(car lst)) (rec_pre_nth n1 n2 (cdr lst) (+ current 1))))
		(t (rec_pre_nth n1 n2 (cdr lst) (+ current 1)))
	)
)
```

6. Напишите две функции swap-to-left и swap-to-right которые производят круговую замену в списке-аргументе влево и вправо соответственно

```
(defun 6_swap_to_left (lst)
	(append
		(mapcar #'eval (cdr lst))
		(list (car lst))
	)
)

(defun 6_swap_to_right (lst)
	(append
		(last lst)
		(mapcar #'eval (butlast lst))
	)
)
```

7. Написать функцию, которая умножает на заданное число-аргумент все числа из заданного списка-аргумента

```
(defun 7_multiply_number (n lst)
	(mapcar #'
		(lambda (lst_elem) (* lst_elem n))
		lst
	)
)
```

8. Написать 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

```
(defun 8_middle (n1 n2 lst)
	(rec_mid n1 n2 lst 0)
)

(defun rec_mid (n1 n2 lst current)
	(cond
		((null lst) nil)
		((= n2 current) nil)
		((<= n1 current) (append (list(car lst)) (rec_pre_nth n1 n2 (cdr lst) (+ current 1))))
		(t (rec_pre_nth n1 n2 (cdr lst) (+ current 1)))
	)
)
```