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 № _7_

Дисциплина Функциональные и логические языки программирования

Студент Козырев Марк

Группа ИУ7-63

Оценка (баллы) _____

Преподаватель _____

Москва.
2020 г.

3. Написать два варианта функции которая возвращает последний элемент своего списка-аргумента

```
(defun 3_1_last_element (lst)
	(last_element lst 0 (length lst))
)

(defun last_element (lst current len)
	(cond
		((null lst) nil)
		((= (+ current 1) len) (car lst) )
		(t (last_element (cdr lst) (+ current 1) len))
	)
)

(defun 3_2_last_element(lst)
	(cond
		((null (cdr lst)) (car lst))
		(t (argum (cdr lst)))
	)
)

(defun 3_3_last_element(lst)
	(last lst)
)
```

4. Написать два варианта функции которая возвращает свой список-аргумент без последнего элемента

```
(defun 4_1_without_last (lst)
	(cond
		((null (cdr lst)) nil)
		(t (append (list (car lst)) (4_without_last (cdr lst))))
	)
)

(defun 4_2_without_last (lst)
	(reverse(cdr (reverse lst)))
)
```

5. Написать простой вариант игры в кости

```
(defun 5_game ()
	(write "first_player: ")
	(write (setf first_player (player)))
	(write "	second_player: ")
	(write (setf second_player (player)))
	(cond
		((or
			(= (summ_2 first_player) 7)
			(= (summ_2 first_player) 11))
		"first player win")
		((or
			(= (summ_2 second_player) 7)
			(= (summ_2 second_player) 11))
		"second player win")
		((<
			(summ_2 first_player)
			(summ_2 second_player))
		"second player win")
		((>
			(summ_2 first_player)
			(summ_2 second_player))
		"first player win")
		((=
			(summ_2 first_player)
			(summ_2 second_player))
		"draw")
	)
)
(defun check_1_6 (1_dice 2_dice)
	(cond
			((or
				(and
					(= 1_dice 1)
					(= 2_dice 1))
				(and
					(= 1_dice 6)
					(= 2_dice 6)))
				(setf 1_dice (random 6))
				(setf 2_dice (random 6))
				(check_1_6 1_dice 2_dice))
	)
	(list 1_dice 2_dice)

)

(defun summ_2 (lst)
	(+ (car lst) (cadr lst))
)
(defun player ()
```

```
        (setf 1_dice (random 6))
        (setf 2_dice (random 6))
        (check_1_6 1_dice 2_dice)
)
```