



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, искусственный интеллект и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №4
по курсу «Функциональное и логическое
программирование»

«Использование управляющих структур, работа со списками»

Студент группы **ИУ7-64Б**

(Подпись, дата)

Д.С. Чепиго

(И.О. Фамилия)

Преподаватели

(Подпись, дата)

Н.Б. Толпинская

(И.О. Фамилия)

(Подпись, дата)

Ю.В. Строганов

(И.О. Фамилия)

2023 г.

Содержание

1	Практические задания	3
1.1	Задание №1	3
1.2	Задание №2	3
1.3	Задание №3	4
1.4	Задание №4	4
1.5	Задание №5	5
1.6	Задание №6	5
1.7	Задание №7	7
1.8	Задание №8	7
1.9	Задание №9	8

1 Практические задания

1.1 Задание №1

Пусть (setf lst1 '(a b c))

(setf lst2 '(d e))

Каковы результаты вычисления следующих выражений?

1. (cons lst1 lst2)
((A B C) D E)
2. (list lst1 lst2)
((A B C) (D E))
3. (append lst1 lst2)
(A B C D E)

1.2 Задание №2

Каковы результаты вычисления следующих выражений, и почему?

- 1) (reverse '(a b c))
(C B A) – обычный разворот списка
- 2) (reverse '(a b (c (d))))
((C (D)) B A) – так как reverse работает относительно элементов списка верхнего уровня
- 3) (reverse '(a))
(A)
- 4) (reverse ())
NIL
- 5) (reverse '((a b c)))
((A B C)) – так как reverse работает относительно элементов списка верхнего уровня, а (A B C) единственный элемент, то он останется на своем месте
- 6) (last '(a b c))
(C) – так как является последним элементом

7) (last '(a))

(A) – так как является единственным, то есть и первым и последним элементом

8) (last '((a b c)))

((A B C)) – так как last работает относительно элементов списка верхнего уровня, а (A B C) единственный элемент, то есть и первый и последний

9) (last '(a b (c)))

((C)) – последний элемент списка верхнего уровня

10) (last ())

(NIL)

1.3 Задание №3

Написать, по крайней мере, два варианта функции, которая возвращает последний элемент своего списка-аргумента.

```
1 (defun lastVar1 (x)
2   (car (reverse x)))
3
4 (defun lastVar2 (x)
5   (if (null (cdr x))
6       (car x)
7       (lastVar2 (cdr x))))
```

1.4 Задание №4

Написать, по крайней мере, два варианта функции, которая возвращает свой список аргумент без последнего элемента.

```
1 (defun withoutLast1 (x)
2   (reverse (cdr (reverse x))))
3
4 (defun withoutLast2 (x)
5   (if (not (cdr x))
```

```

6         NIL
7         (cons (car x) (withoutLast2 (cdr x)))))

```

1.5 Задание №5

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

```

1  (defun myLast (x)
2      (if (not x)
3          x
4          (if (not (cdr x))
5              x
6              (myLast (cdr x)))))
7
8  (defun changeLast (x el)
9      (if (not x)
10         x
11         (if (not (cdr x))
12             (cons el Nil)
13             (cons (car x) (changeLast (cdr x) el)))))
14
15  (defun swap-first-last (x)
16      (changeLast (cons (car (myLast x)) (cdr x)) (car x)))

```

1.6 Задание №6

Написать простой вариант игры в кости, в котором бросаются две правильные кости. Если сумма выпавших очков равна 7 или 11 — выигрыш, если выпало (1,1) или (6,6) — игрок имеет право снова бросить кости, во всех остальных случаях ход переходит ко второму игроку, но запоминается сумма выпавших очков. Если второй игрок не выигрывает абсолютно, то выигрывает тот игрок, у которого больше очков. Результат игры и значения выпавших костей выводить на экран с помощью функции `print`.

```

1  (defun roll_dice ()
2      ( setf *random-state* (make-random-state t))
3      (+ (random 6) 1))
4
5  (defun check_continue_game (result)
6      (not (or (= result 7) (= result 11))) )
7
8  (defun make_a_move (player_i)
9      (let ((dice1 (roll_dice)) (dice2 (roll_dice)))
10
11          (if (and
12              (print (list 'Игрок player_i 'бросает 'кости))
13              (= dice1 dice2) (or (= dice1 1) (= dice1 6)))
14              (and
15                  (print (list 'Выпало dice1 dice2 'повтор))
16                  (make_a_move player_i )
17              )
18              (and
19                  (print (list 'Выпало dice1 dice2))
20                  (+ dice1 dice2))))))
21
22  (defun compare_results (res1 res2)
23      (if (check_continue_game res2)
24          (and
25              (print (list 'Сравнение 'по 'очкам))
26              (print ( list 'Игрок 1 'набрал res1))
27              (print ( list 'Игрок 2 'набрал res2))
28              (cond
29                  ((< res1 res2)
30                   (and
31                     (print (list 'Игрок 2 'выиграл 'по 'очкам))
32                     2 ))
33

```

```

34         (> res1 res2)
35         (and
36         (print (list 'Игрок 1 'выиграл 'по 'очкам))
37         1 ))
38         ((and (print 'Ничья ()) 0))))
39
40     (and
41         (print (list 'Игрок 2 'набрал res2 'очков
42         'и 'выиграл 'абсолютно)) 2 )))
43
44 (defun game ()
45     (let ((res1 (make_a_move 1)))
46         (if (check_continue_game res1)
47             (compare_results res1 (make_a_move 2))
48             (and (print (list 'Игрок 1 'набрал res1 'очков
49             'и 'выиграл 'абсолютно)) 1))))
50
51 (game)

```

1.7 Задание №7

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

```

1 (defun listEuale(x)
2     (equal x (reverse x)))

```

1.8 Задание №8

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

```

1 (defun countryByCapital (table capital)
2     (if (equal capital (cdr (car table)))

```

```

3      (car (car table))
4      (countryByCapital (cdr table) capital)))
5
6 (defun capitalByCountry (table country)
7   (if (equal country (car (car table)))
8       (cdr (car table))
9       (capitalByCountry (cdr table) country)))

```

1.9 Задание №9

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.

```

1 (defun mulFirstNumbers (lst number)
2   (cons (* number (car lst)) (cdr lst)))
3
4 (defun mulFirstAny (lst number)
5   (cond ((null lst) lst)
6         ((numberp (car lst))
7          (cons (* (car lst) number) (cdr lst)))
8         (T (cons (car lst)
9                  (mulFirstAny (cdr lst) number)))))
10

```