



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»

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

Тема Использование управляющих структур, работа со списками

Студент Калашков П. А.

Группа ИУ7-66Б

Оценка (баллы) _____

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

1 Практические задания

Задание 1. Чем принципиально отличаются функции `cons`, `list`, `append`? Пусть `(setf lst1 '(a b c))` и `(setf lst2 '(d e))`. Каковы результаты вычисления следующих выражений?

Листинг 1.1 – Задание 1

```
1 (setf lst1 '(a b c))
2 (setf lst2 '(d e))
3
4 (cons lst1 lst2) ; ((a b c) d e)
5 (list lst1 lst2) ; ((a b c) (d e))
6 (append lst1 lst2) ; (a b c d e)
```

Задание 2. Каковы результаты вычисления следующих выражений, и почему?

Листинг 1.2 – Задание 2

```
1 (reverse '(a b c)) ; (c b a)
2 (reverse '(a b (c (d)))) ; ((c (d)) b a)
3 (reverse '(a)) ; (a)
4 (last '(a b c)) ; (c)
5 (last '(a)) ; (a)
6 (last '((a b c))) ; ((a b c))
7 (reverse ()) ; Nil
8 (reverse '((a b c))) ; ((a b c))
9 (last '(a b (c))) ; ((c))
10 (last ()) ; Nil
```

Задание 3. Написать, по крайней мере, два варианта функции, которая возвращает последний элемент своего списка-аргумента.

Листинг 1.3 – Задание 3, вариант 1

```
1 (defun get_last (lst)
2   (
3     cond
4     (
5       (cdr lst)
6       (get_last (cdr lst))
7     )
8     (
9       T
10      (car lst)
11    )
12  )
13 )
```

Листинг 1.4 – Задание 3, вариант 2

```
1 (defun get_last (lst)
2   (car (last lst))
3 )
```

Листинг 1.5 – Задание 3, вариант 3

```
1 (defun get_last (lst)
2   (car (reverse lst))
3 )
```

Задание 4. Написать, по крайней мере, два варианта функции, которая возвращает свой список аргумент без последнего элемента.

Листинг 1.6 – Задание 4, вариант 1

```
1 (defun without_last (lst)
2   (
3     cond
4     (
5       (cdr lst)
6       (cons (car lst) (without_last (cdr lst)))
7     )
8     (
9       T
10      Nil
11    )
12  )
13 )
```

Листинг 1.7 – Задание 4, вариант 2

```
1 (defun without_last (lst)
2   (reverse (cdr (reverse lst))))
3 )
```

Задание 5. Напишите функцию `swap-first-last`, которая переставляет в списке-аргументе первый и последний элементы.

Листинг 1.8 – Задание 5

```
1 (defun swap-first-last (lst)
2   (append
3     (last lst)
4     (reverse (cdr (reverse (cdr lst)))))
5     (cons (car lst) Nil)
6   )
7 )
```

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`.

Листинг 1.9 – Задание 6

```
1 (defvar dices)
2 (defvar roll1)
3 (defvar roll2)
4
5 (defun roll_1_dice ()
6   (+ (random 6) 1)
7 )
8
9 (defun roll ()
10  (+ (roll_1_dice) (roll_1_dice))
11 )
12
13 (defun is_win (throw)
14   (cond
15     ((or (= throw 7) (= throw 11)) T)
16     (T Nil)
17   )
18 )
19
20 (defun is_luck (throw)
21   (cond
22     ((or (= throw 2) (= throw 12)) T)
23     (T Nil)
24   )
25 )
26
27 (defun turn ()
28   (terpri)
29   (setq dices (roll))
30   (princ "Rolled ")
31   (princ dices)
32
33   (cond
34     ((is_win dices) 0)
35     ((is_luck dices) (turn))
36     (T dices)
37   )
38 )
39
40 (defun game())
```

```

41 (princ "First player is rolling: ")
42 (setq roll1 (turn))
43 (terpri)
44
45 (cond
46   ((eq roll1 0) (princ "First player won absolutely!"))
47   (T ((lambda ()
48         (princ "Second player is rolling: ")
49         (setq roll2 (turn))
50         (terpri)
51         (cond
52          ((eq roll2 0) (princ "Second player won absolutely!"))
53          ((eq roll1 roll2) (princ "Draw!"))
54          ((> roll1 roll2) (princ "First player won!"))
55          (T (princ "Second player won!"))
56         )
57       )))
58   )))
59 )
60 (terpri)
61 )

```

Задание 7. Написать функцию, которая по своему списку-аргументу `lst` определяет является ли он палиндромом (то есть равны ли `lst` и `(reverse lst)`).

Листинг 1.10 – Задание 7

```

1 (defun is_palindrom (lst)
2   (equalp lst (reverse lst))
3 )

```

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.

Листинг 1.11 – Задание 8, IF

```
1 (defun show_on_map (lst item)
2   (cond
3     ((equal (caar lst) item) (cdar lst))
4     ((equal (cdar lst) item) (caar lst))
5     ((cdr lst) (show_on_map (cdr lst) item))
6   )
7 )
```

Задание 9.

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

- а) все элементы списка — числа,
- б) элементы списка – любые объекты.

Листинг 1.12 – Задание 9, а

```
1 (defun mul_if_all_num (lst num)
2   (cond
3     ((and
4       (numberp num)
5       (and
6         (numberp (car lst))
7         (and
8           (numberp (cadr lst))
9           (numberp (caddr lst))
10        )
11      )
12     ) (* (car lst) num))
13     (T Nil)
14   )
```

15 |)

Листинг 1.13 – Задание 9, b

```
1 (defun mul_first_num (lst num)
2   (cond
3     ((and (numberp (car lst)) (numberp num)) (* (car lst) num))
4     ((cdr lst) (mul_first_num (cdr lst) num))
5     (T Nil)
6   )
7 )
```