



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»

## Отчёт по лабораторной работе №4 по курсу

### «Функциональное и логическое программирование»

Тема Использование управляющих структур, работа со списками.

Студент Сироткина П.Ю.

Группа ИУ7-66Б

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

Москва — 2022 г.

# 1 Практические задания

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

```
1 (setf lst1 '(a b))
2 (setf lst2 '(c d))
3 (cons lst1 lst2)      ; ((A B) C D)
4 (list lst1 lst2)      ; ((A B) (C D))
5 (append lst1 lst2)    ; (A B C D)
```

**1.2** Каковы результаты вычисления следующих выражений, и почему?

```
1 (reverse ())          ; NIL
2 (last ())             ; NIL
3 (reverse '(a))        ; (A)
4 (last '(a))           ; (A)
5 (reverse '((a b c)))  ; ((A B C))
6 (last '((a b c)))     ; ((A B C))
```

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

```
1 (defun get_last(lst) (last lst))
```

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

```
1 (defun get_last(lst) (if (cdr lst) (get_last(cdr lst)) (car lst)))
```

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

```
1 (defun del_last(lst) (reverse (cdr (reverse lst))))
```

```
1 (defun del_last(lst) (if (cdr lst) (cons (car lst) (del_last (cdr lst)))
  Nil))
```

## 1.5 Написать простой вариант игры в кости.

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 в праве 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_dices()
2   (list (+ (random 6) 1) (+ (random 6) 1)))
3
4 (defun sum_points(dices)
5   (+ (first dices) (second dices)))
6
7 (defun is_absolute_winner(dices)
8   (or (= (sum_points dices) 7) (= (sum_points dices) 11)))
9
10 (defun need_reroll(dices)
11   (or (= (sum_points dices) 2) (= (sum_points dices) 12)))
12
13 (defun process(fdices sdices)
14   (setq fsum (sum_points fdices))
15   (setq ssum (sum_points sdices))
16
17   (if (need_reroll fdices) (print (and (print 'Reroll_for_player_#1) (setq
18     fdices (roll_dices)))))
19
20   (if (is_absolute_winner fdices)
21     'Player_#1_win
22     (if (is_absolute_winner sdices)
23       'Player_#2_win
24       (if (> (sum_points sdices) (sum_points fdices)) 'Player_#2_win 'Player_#1
25         _win)
26     )
27   )
28
29 (defun play()
30   (print '=====Roll_dices=====)
31   (setq fdices (roll_dices))
32   (setq sdices (roll_dices))
33
34   (print 'Player_#1)
35   (print fdices)
36   (terpri)
37
38   (print 'Player_#2)
39   (print sdices)
```

```
39 (terpri)
40
41 (print '=====Start_game=====')
42 (print (process fdices sdices))
43 (terpri)
44 )
45
46 (play)
```