



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**

**По дисциплине «Функциональное и логическое программирование»**

**Студент: Тимонин А. С.**

**Группа ИУ7-626**

**Преподаватель Толпинская Н. Б.**

Москва.  
2020 г.

## Практическая часть

### Задание 1.

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

```
(cons lst1 lst2) ; result: ((A B) C D)
(list lst1 lst2) ; result: ((A B) (C D))
(append lst1 lst2) ; result: (A B C D)
```

### Задание 2.

Каковы результаты вычисления следующих выражений?

```
(reverse ()) ; NIL
(last ()) ; result: NIL
(reverse '(a)) ; result: (A)
(last '(a)) ; result: (A)
(reverse '((a b c))) ; result: (C B A)
(last '((a b c))) ; result: ((A B C))
```

### Задание 3.

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

```
(defun getLastElement1 (lst)
  (if (null (cdr lst))
      (car lst)
      (getLastElement1 (cdr lst)))
)

(defun getLastElement2 (lst)
  (cond
    ((null (cdr lst)) (car lst))
    (t (getLastElement2 (cdr lst))))
)
```

#### Задание 4.

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

```
(defun getList1 (lst)
  (
    cons (car lst)
      (if (> (length (cdr lst)) 1)
        (getList1 (cdr lst))
        nil)
  )
)

(defun getList2 (lst)
  (
    cons (car lst)
      (cond
        ((> (length (cdr lst)) 1) (getList2 (cdr
lst)))
        (t nil)
      )
    )
  )
)
```

#### Задание 5.

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 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.

```

; Ролл 2ух кубиков
(defun roll-dice ()
  (list (+ (random 6) 1) (+ (random 6) 1)))

; Проверка на реролл
(defun is-reroll (lst)
  (or
    (and (equal 1 (car lst)) (equal 1 (cadr lst)))
    (and (equal 6 (car lst)) (equal 6 (cadr lst)))
  )
)

; Проверка на победу
(defun is-win (lst)
  (or
    (equal 7 (+ (car lst) (cadr lst)))
    (equal 11 (+ (car lst) (cadr lst)))
  )
)

; Сумма ролла
(defun sum-roll (lst)
  (+ (car lst) (cadr lst))
)

(defun play ()
  (setq player1 (roll-dice))
  (format t "Player1 have roll ~A. ~%" player1)
  (if (is-reroll player1)
    (play)
    (if (is-win player1)
      (format t "Player1 WIN with roll ~A. ~%" player1)
      (first(list
        (setq player2 (roll-dice))
        (format t "Player2 have roll ~A. ~%" player2)
        (if (is-win player2)
          (format t "Player2 WIN with roll ~A. ~%" player2)
          (if (= (sum-roll player1) (sum-roll player2))
            (play)
            (if (> (sum-roll player1) (sum-roll player2))
              (format t "Player1 WIN with roll ~A ~%" player1)
              (format t "Player2 WIN with roll ~A ~%" player2)
            )
          )
        )
      )
    )
  )
)

```