



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»

## ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

По курсу: "Функциональное и Логическое программирование"

Тема      Использование управляющих структур, работа со списками.

Группа      ИУ7-63Б (ИУ7и-676)

Студент      Тэмуужин Я.

Преподаватель      Толпинская Н.Б.

Преподаватель      Строганов Ю. В.

**Задание 1.** Пусть `(setf lst1 '(a b)) (setf lst2 '(c d))`. Каковы результаты вычисления следующих выражений?

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

**Задание 2.** Каковы результаты вычисления следующих выражений?

```
(reverse ())          ;; Nil
(last ())              ;; Nil
(reverse '(a))         ;; (A)
(last '(a))            ;; (A)
(reverse '((a b c)))   ;; ((A B C))
(last '((a b c)))      ;; ((A B C))
```

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

```
(defun f-l1 (lst)
  (last lst))
```

```
(defun f-l2 (lst)
  (car (reverse lst)))
```

```
(defun f-l3 (lst)
  (if (null (cdr lst)) (car lst)
      (f-l3 (cdr lst))))
```

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

```
(defun f-no-l1 (lst)
  (reverse (cdr (reverse lst))))
```

```
(defun f-no-l2 (lst)
  (butlast lst))
```

```
(defun f-no-l3-ins (lst res)
  (if (null (cdr lst)) res
      (f-no-l3-ins (cdr lst)
                    (append res (cons (car lst) nil)))))

(defun f-no-l3 (lst) (f-no-l3-ins lst ()))
```

**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.

```
(defun random-dice-pair ()
  (list (+ 1 (random 6)) (+ 1 (random 6))))

(defun dice-pair-sum (pair)
  (+ (car pair) (cadr pair)))

(defun is-win-sum (sum)
  (or (equal sum 7) (equal sum 11)))

(defun is-retry-pair (pair)
  (or (and (equal 1 (car pair)) (equal 1 (cadr pair)))
      (and (equal 6 (car pair)) (equal 6 (cadr pair))) ))

(defun play ()
  (let* ((pair1 (random-dice-pair))
        (sum1 (dice-pair-sum pair1)))
    (format t "Player~1~pair:~s~%Player~1~pair~sum:~s~%~%" pair1 sum1)
    (cond ((is-win-sum sum1) (princ "Player~1~is~winner"))
          ((is-retry-pair pair1) (format t "Retry~%" (play)))
          (t (let* ((pair2 (random-dice-pair))
                    (sum2 (dice-pair-sum pair2)))
               (format t "Player~2~pair:~s~%Player~2~pair~sum:~s~%~%" pair2
                        sum2)
               (cond ((is-win-sum sum2) (princ "Player~2~is~winner"))
                     ((> sum1 sum2) (princ "Player~1~is~winner"))
                     ((< sum1 sum2) (princ "Player~2~is~winner"))
                     (t (princ "Draw~!")))))))
```

### Результаты:

```
CL-USER> (play)
Player 1 pair: (3 3)
Player 1 pair sum: 6

Player 2 pair: (5 3)
Player 2 pair sum: 8

Player 2 is winner
```

```
CL-USER> (play)
Player 1 pair: (5 5)
Player 1 pair sum: 10

Player 2 pair: (5 2)
Player 2 pair sum: 7

Player 2 is winner
```

```
CL-USER> (play)
Player 1 pair: (1 4)
Player 1 pair sum: 5

Player 2 pair: (2 3)
Player 2 pair sum: 5

Draw !
```

```
CL-USER> (play)
Player 1 pair: (6 6)
Player 1 pair sum: 12

Retry
Player 1 pair: (2 1)
Player 1 pair sum: 3

Player 2 pair: (4 1)
Player 2 pair sum: 5

Player 2 is winner
```

```
CL-USER> (play)
Player 1 pair: (6 5)
Player 1 pair sum: 11

Player 1 is winner
```

```
CL-USER> (play)
Player 1 pair: (4 3)
Player 1 pair sum: 7

Player 1 is winner
```