



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 №6

По курсу: "Функциональное и Логическое программирование"

Тема Использование управляющих структур, работа со списками.

Группа ИУ7-63Б

Студент Сукочева А.

Преподаватель Толпинская Н.Б.

Преподаватель Строганов Ю. В.

Практическая часть

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-last-rec (lst)
  (cond ((null (cdr lst)) (car lst))
        (T (f-last-rec (cdr lst)))))
```

Беремен первый элемент от перевернутого списка.

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

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

```
(defun f1 (lst)
  (reverse (cdr (reverse lst))))
```

```
(defun f1-rec (lst)
  (cond ((null (cdr lst)) ())
        (T (cons (car lst) (f1-rec (cdr 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 получает 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.

Теоретическая часть

Структуроразрушающие и не разрушающие структуру списка функции

Функции для работы со списками делятся на две группы:

1. Не разрушающие структуру. Если сохраняется возможность работать с исходными списками, значит функции не разрушают структуру. (Пример: `append`, `reverse`, `length`, `subst ...`)
2. Разрушающие структуру. Если после использования какой-то стандартной функции после ее работы теряется возможность работы с теми списками, которые изначально были, значит их структура разрушилась. Чаще всего такие функции начинаются в буквы 'n (Пример: `ncons`, `nreverse`, `nsubst ...`)

Отличие в работе функции `cons`, `list`, `append` и в их результате