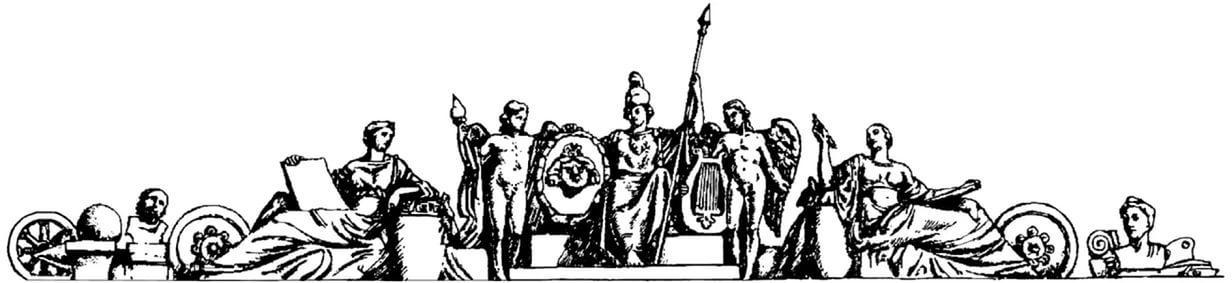
Министерство образования Российской Федерации

Московский Государственный Технический Университет им. Н.Э. Баумана

Отчет по лабораторной работе №7

По курсу «Функциональное и логическое

программирование»

Студент Сергеева М.В.

Группа ИУ7-65 Преподаватель Толпинская Н.Б.

2019

### Практическая часть.

### Задание 1.

Чем принципиально отличаются функции cons, list, append?

«Cons» создает только одну списковую ячейку, первый указатель которой указывает на первый аргумент функции «cons», а второй указатель на второй. «List» создает столько списковых ячеек, сколько аргументов было передано, создавая таким образом список из аргументов. Функция «append» производит объединение двух списков-аргументов в один список.

Пусть (setf lst1 '( a b)) (setf lst2 '( c d))

Каковы результаты вычисления следующих выражений:

1) > (cons lst1 lst2)

((A B) C D)

2) > (list lst1 lst2)

((A B) (C D))

3) > (append lst1 lst2)

(A B C D)

### Задание 2.

Каковы результаты вычисления следующих выражений:

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))

### Задание 3.

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

> (defun mylast1 (x) (car (reverse x)))

> (defun mylast2 (x) (cond ((<= (length x) 1) (car x)) (T (mylast2 (cdr x)))))

### Задание 4.

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

> (defun no-last1 (x) (reverse (cdr (reverse x))))

> (defun no-last2 (x) (cons (car x) (if (> (length (cdr x)) 1) (no-last2 (cdr x)) 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 получает право снова бросить кости, во всех остальных

случаях ход переходит ко второму игроку, но запоминается сумма выпавших очков.

Если второй игрок не выигрывает абслютно, то выигрывает тот игрок, у которого

больше очков. Результат игры и значения выпавших костей выводить на экран с

помощью функции print.

> (defun dice () (+ (random 6) 1))

> (defun roll () (list (dice) (dice)))

> (defun reroll (lst) (or (equal lst '(1 1)) (equal lst '(6 6))))

> (defun turn () (let ((first (roll))) (if (reroll first) (roll) first)))

> (defun summ-roll (lst) (+ (car lst) (cadr lst)))

> (defun win (lst) (let ((sum (summ-roll lst))) (or (= sum 11) (= sum 7))))

> (defun who-win (roll1 roll2) (cond

((win roll1) "Player 1 instante win")

((win roll2) "Player 2 instante win")

((> (summ-roll roll1) (summ-roll roll2)) "Player 1 win")

((< (summ-roll roll1) (summ-roll roll2)) "Player 2 win")

(T "Dead heat")))

> (defun game () (let ((roll1 (turn)) (roll2 (turn))) (print (list roll1 roll2 (who-win roll1 roll2)))))

## 

## 

## 

## 

## Задание 9(6).

Написать рекурсивную версию с именем rec-nth функции nth.

