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 предмету: «Функциональное и логическое программирование»

Студент:
Коротков Андрей Владимирович
Группа: ИУ7-65Б

Преподаватели:
Толпинская Наталья Борисовна
Строганов Юрий Владимирович

Москва, 2020 г.

Задание №1. Чем принципиально отличаются функции cons, list, append?
Функция cons является базисной функцией и создает спусковую ячейку, устанавливая указатель car на первый аргумент, а указатель cdr на второй элемент. Функция list для каждого аргумента создает списковую ячейку, последовательно соединяя их между собой в список. Функция append переставляет cdr указатель 1-го списка на голову начала 2-го списка.

Пусть (setf lst1 `(a b)) (setf lst2 `(c d))
Каковы результаты вычисления следующих выражений?
(cons lst1 lst2)
Ответ: ((a b) c d)

(list lst1 lst2)
Ответ: ((a b) (c d))

(append lst1 lst2)
Ответ: (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 два варианта функции, которая возвращает последний элемент своего списка-аргумента.

(defun last1 (x) (car (last x)))
(defun last2(x) (car (reverse x)))

Задание №4. Написать два варианта функции, которая возвращает свой список-аргумент без последнего элемента.

(defun without_last1 (x) (reverse (cdr (reverse x))))
(defun without_last2 (x) (reverse (nthcdr 1 (reverse x))))

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 выигрывал абсолютно, то

выигрывает тот игрок, у которого больше очков. Результат игры и значения выпавших костей выводить на экран с помощью функции print.

```
(defun rand_num () (+ (random 6) 1))
(defun roll () (list (rand_num) (rand_num)))
(defun need_to_reroll (lst) (or (equal lst '(1 1)) (equal lst '(6 6))))
(defun summ_roll (lst) (+ (first lst) (second lst)))
(defun win (lst) (let ((sum (summ_roll lst))) (or (= sum 11) (= sum 7))))

(defun play(num)
  (let ((lst (roll))) (print (list lst "Num player:" num)) (if (need_to_reroll lst)
    (play num)
    (if (win lst)
    Nil
    lst))))

(defun start_game()
  (let ((player1 (play 1)) (player2 (play 2)))
    (cond ((null player1) (print "player1 absolute won"))
          ((null player2) (print "player2 absolute won"))
          ((> (summ_roll player1) (summ_roll player2)) (print "player1 won"))
          ((< (summ_roll player1) (summ_roll player2)) (print "player2 won"))
          ((= (summ_roll player1) (summ_roll player2)) (print "draw")))))
```