|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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**

|  |  |
| --- | --- |
| **Тема**  Lisp  **Студент** Белоусова Ю.С.  **Группа** ИУ7-61Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель** Толпинская Н.Б. |  |

Москва.

2020 г.

**1. Чем принципиально отличаются функции cons, list, append?**

**cons** - создаёт одну списочную ячейку. CAR-указатель ячейки указывает на первый аргумент, CDR-указатель - на второй.

**list** *-* создаёт список, в котором car-указатель i-й списочной ячейки указывает на i-й аргумент.

**append** - копирует все аргументы, кроме последнего, и создает из них список по следующему принципу. CDR-указатель последнего элемента i-й копии указывает на первый элемент (i+1)-й копии. CDR-указатель последней копии указывает на последний (не скопированный) аргумент.

**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)

**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)) |

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

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

(defun mylast2 (x)

(if (equal (cdr x) nil)

(car x)

(mylast2 (cdr x))

)

)

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

(defun cut1 (x)

(reverse (cdr (reverse x)))

)

(defun cut2 (x)

(cons (car x)

(if (> (length (cdr x)) 1)

(cut2 (cdr x))

nil

)

)

)

**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. Результат игры и значения выпавших костей выводить на экран с помощью функции print.**

(defun roll-dice () (print (list (+ (random 6) 1) (+ (random 6) 1))) )

(defun reroll-if-need (x)

(if (or (equal x '(1 1))

(equal x '(6 6)))

(and (print "Reroll!") (roll-dice))

x

)

)

(defun roll () (reroll-if-need (roll-dice)) )

(defun roll-sum (r) (+ (first r) (second r)) )

(defun is-winner (res)

(if (or (= res 7)

(= res 11)

)

t

nil

)

)

(defun play ()

(cond ( (and (print "First player:")

(is-winner (setq p1 (roll-sum (roll))))

)

"First player won"

)

( (and (print "Second player:")

(is-winner (setq p2 (roll-sum (roll))))

)

"Second player won"

)

((= p1 p2) "Tie!")

((> p1 p2) "First player won")

(t "Second player won")

)

)