## 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, и почему?

**(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** last1**(**lst**)**

**(if** **(null** **(cdr** lst**))**

**(car** lst**)**

**(**last1 **(cdr** lst**))))**

**(defun** last2**(**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** f2**(**lst**)**

**(if** **(null** **(cdr** lst**))**

NIL

**(cons** **(car** lst**)** **(**f2 **(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 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**(**n**)**

**(let** **((**d1 **(+** 1 **(random** 6**)))** **(**d2 **(+** 1 **(random** 6**))))**

**(**format t "Player~A rolled ~A+~A=~A~%"

n d1 d2 **(+** d1 d2**))**

**(+** d1 d2**)))**

**(defun** check\_win**(**sum**)**

**(or** **(=** sum 7**)** **(=** sum 11**)))**

**(defun** check\_reroll**(**sum**)**

**(or** **(=** sum 2**)** **(=** sum 12**)))**

**(defun** turn**(**n**)**

**(let** **((**sum **(**roll n**)))**

**(cond** **((**check\_win sum**)** **nil)**

**((**check\_reroll sum**)** **(**turn n**))**

**(**T sum**))))**

**(defun** dice\_game**()**

**(cond** **((not** **(setf** p1 **(**turn 1**)))** **(**print **'(**Player1 won!**)))**

**((not** **(setf** p2 **(**turn 2**)))** **(**print **'(**Player2 won!**)))**

**(**T **(cond** **((>** p1 p2**)** **(**print **'(**Player1 won!**)))**

**((<** p1 p2**)** **(**print **'(**Player2 won!**)))**

**(**T **(**print **'Tie!))))))**