



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

Студент Лучина Е.Д

Группа ИУ7-61Б

Преподаватель Толпинская Н.Б.

Москва.
2020 г.

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

Пусть

```
(setf lst1 '( a b))
```

```
(setf lst2 '( d))
```

Каковы результаты вычисления следующих выражений?

```
(cons lst1 lst2) -> ((a b) d)
```

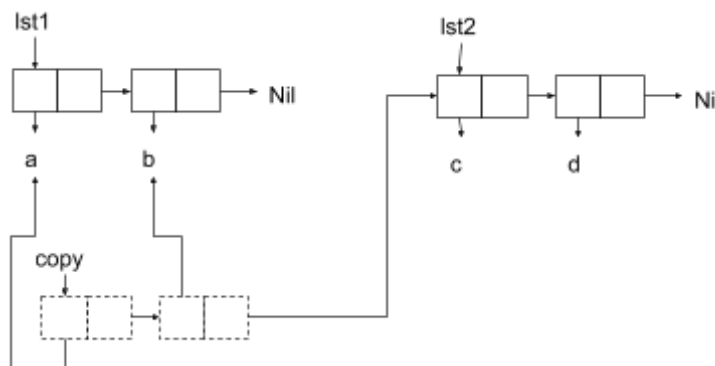
```
(list lst1 lst2) -> ((a b) (d))
```

```
(append lst1 lst2) -> (a b d)
```

Cons. Создает точечную пару. Принимает ровно два аргумента. Создает ровно одну списковую ячейку и расставляет указатели: car - первый аргумент, cdr - второй аргумент.

List. Создает список. Принимает любое количество аргументов. Выделяет столько списковых ячеек, сколько аргументов передано. Указатели car расставляет последовательно на переданные аргументы. Указатели cdr связывают ячейки в список, последний указывает на Nil.

Append. Добавляет один список в конец предыдущего. Принимает любое количество параметров - списков. Функция не разрушающая структуру своих аргументов. Создает копии всех своих аргументов, кроме последнего. Последние указатели каждого из копий-списков устанавливает на начало следующего списка.



2. Каковы результаты вычисления следующих выражений?

```
(reverse Q) - error Q has no value
```

```
(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 a (&rest arguments)
  (car (last arguments)))
```

```
(defun a (&rest arguments)
  (nth (- (length arguments) 1) arguments))
```

))
---	---

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

```
(defun func1(&rest arguments)
  (reverse (cdr (reverse arguments))))
)
```

```
(defun myloop(L)
  (if (null (rest L))
      nil
      (cons (first L) (myloop (rest L))))
)
)
```

```
(defun func(&rest arguments)
  (myloop arguments)
)
```

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 имеет 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.

<pre>(defun turn () (and (setq a (read)) (setq b (read)) (setq s1 (+ a b)) (print s1) (if (or (eq s1 7) (eq s1 11)) 13 (if (or (eq s1 2) (eq s1 12)) (and (print 'again) (turn)) s1))))</pre>	<pre>(defun play () (and (print '(turn of the first player)) (setq s1 (turn)) (if (< s1 13) (and (print '(turn of the second player)) (setq s2 (turn)) (if (< s2 13) (if (< s1 s2) (print '(second player won the game)) (print '(first player won the game))) (print '(second player won the game))))) (print '(first player won the game))))</pre>
--	---

)
--	---

(play) <== (1 1) (2 3) (6 6) (3 4)

(TURN OF THE FIRST PLAYER) 2

AGAIN 5

(TURN OF THE SECOND PLAYER) 12

AGAIN 7

(SECOND PLAYER WON THE GAME)