



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический  
университет имени Н.Э. Баумана»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления»

---

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

---

## Лабораторная работа № 6

По курсу «Функциональное и логическое программирование».

### Рекурсивные функции

Студент	Степанов А.О.
Группа	ИУ7-63Б
Преподаватель	Толпинская Н.Б.

Москва, 2020 г.

# ЗАДАНИЕ

Дано два списка: первый список название стран, второй – столиц.

— из двух списков создать список из двухэлементных списков

— из двух списков создать список из точечных пар

По полученным спискам по стране найти столицу и наоборот.

```
1 (setq countries '(Russia USA GB Belarus))
2 (setq cities '(Moscow Washington London Minsk))
```

## СПОСОБ 1: СПИСКИ

Создание списка, состоящего из двухэлементных списков.

```
1 (defun list_merge_cc (countries cities)
2   (cond
3     ((null countries) Nil)
4     ((null cities) Nil)
5     (T (cons
6         (list (car countries) (car cities))
7         (list_merge_cc (cdr countries) (cdr cities))
8       )))
9   )
10 )
11
12 (setq list_merged_cc (list_merge_cc countries cities))
```

Поиск страны по столице в списке из двухэлементных списков.

```
1 (defun list_find_country (list_cc city)
2   (cond
3     ((null list_cc) Nil)
4     ((equal (cadar list_cc) city) (caar list_cc))
5     (T (list_find_country (cdr list_cc) city))
6   )
7 )
8
9 (list_find_country list_merged_cc 'Moscow) ;;; Russia
10 (list_find_country list_merged_cc 'Minsk) ;;; Belarus
11 (list_find_country list_merged_cc 'Sidney) ;;; Nil
```

Поиск столицы по стране в списке из двухэлементных списков.

```

1 (defun list_find_city (list_cc country)
2   (cond
3     ((null list_cc) Nil)
4     ((equal (caar list_cc) country) (cadar list_cc))
5     (T (list_find_city (cdr list_cc) country))
6   )
7 )
8
9 (list_find_city list_merged_cc 'Russia) ;;; Moscow
10 (list_find_city list_merged_cc 'Belarus) ;;; Minsk
11 (list_find_city list_merged_cc 'Australia) ;;; Nil

```

## СПОСОБ 2: ТОЧЕЧНЫЕ ПАРЫ

Создание списка, состоящего из точечных пар.

```

1 (defun cons_merge_cc (countries cities)
2   (cond
3     ((null countries) Nil)
4     ((null cities) Nil)
5     (T (cons
6         (cons (car countries) (car cities))
7         (cons_merge_cc (cdr countries) (cdr cities))
8       ))
9   )
10 )
11
12 (setq cons_merged_cc (cons_merge_cc countries cities))

```

Поиск страны по столице в списке из двухэлементных списков.

```

1 (defun cons_find_country (list_cc city)
2   (cond
3     ((null list_cc) Nil)
4     ((equal (cdar list_cc) city) (caar list_cc))
5     (T (cons_find_country (cdr list_cc) city))
6   )
7 )
8
9 (cons_find_country cons_merged_cc 'Moscow) ;;; Russia
10 (cons_find_country cons_merged_cc 'Minsk) ;;; Belarus
11 (cons_find_country cons_merged_cc 'Sidney) ;;; Nil

```

Поиск столицы по стране в списке из двухэлементных списков.

```

1 (defun cons_find_city (list_cc country)
2   (cond
3     ((null list_cc) Nil)
4     ((equal (caar list_cc) country) (cdar list_cc))
5     (T (cons_find_city (cdr list_cc) country))
6   )
7 )
8
9 (cons_find_city cons_merged_cc 'Russia) ;;; Moscow
10 (cons_find_city cons_merged_cc 'Belarus) ;;; Minsk
11 (cons_find_city cons_merged_cc 'Australia) ;;; Nil

```

## ВЫВОДЫ

Удобней и эффективней использовать второй способ со списком, состоящим из точечных пар, так как для обращения к второму элементу пары необходимо использовать на одну функцию меньше. Также при использовании списка, состоящего из двухэлементных списков на каждую пару выделяется три списковых ячейки, а во втором случае – только две.