



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»

## **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3 по курсу «Функциональное и логическое программирование»**

Студент \_\_\_\_\_ Маслова Марина Дмитриевна

Группа \_\_\_\_\_ ИУ7-63Б

Оценка (баллы) \_\_\_\_\_

Преподаватель \_\_\_\_\_ Толпинская Наталья Борисовна

Преподаватель \_\_\_\_\_ Строганов Юрий Владимирович

2022 г.

# 1 Практические задания

## 1.1 Задание №1

Написать функцию, которая принимает целое число и возвращает первое четное число, не меньшее аргумента.

```
1 (defun first-greater-even (num)
2   (if (evenp num) num (+ num 1)))
```

## 1.2 Задание №2

Написать функцию, которая принимает число и возвращает число того же знака, но с модулем на 1 больше модуля аргумента.

```
1 (defun plus-minus-one (num)
2   (+ num (if (> num 0) 1 -1)))
```

## 1.3 Задание №3

Написать функцию, которая принимает два числа и возвращает список из этих чисел, расположенных по возрастанию.

```
1 (defun make-growing-list (a b)
2   (if (< a b) (list a b) (list b a)))
```

## 1.4 Задание №4

Написать функцию, которая принимает три числа и возвращает Т только тогда, когда первое число расположено между вторым и третьим.

```
1 (defun is-between (a b c)
2   (or (and (> a b) (< a c)) (and (> a c) (< a b))))
```

## 1.5 Задание №5

Каков результат вычисления следующих выражений?

```
1 (and 'fee 'fie 'foe)      ; foe
2 (or nil 'fie 'foe)       ; fie
3 (and (equal 'abc 'abc) 'yes) ; yes
4 (or 'fee 'fie 'foe)      ; fee
5 (and nil 'fie 'foe)      ; nil
6 (or (equal 'abc 'abc) 'yes) ; T
```

## 1.6 Задание №6

Написать предикат, который принимает два числа-аргумента и возвращает Т, если первое число не меньше второго.

```
1 (defun pred (a b)
2   (>= a b))
```

## 1.7 Задание №7

Какой из следующих двух вариантов предиката ошибочен и почему?

```
1 (defun pred1 (x)
2   (and (numberp x) (plusp x)))
3
4 (defun pred2 (x)
5   (and (plusp x) (numberp x)))
```

Ошибочным является второй вариант, так как функция `plusp` принимает на вход один аргумент типа `number`, из-за чего аргументы, не являющиеся числами, будут вызывать ошибку. При этом в первом варианте, если при проверке, является ли аргумент числом, получится значение `Nil`, то `and` вернет его в качестве результата, не продолжая дальнейшие вычисления, а при передаче числа будет проведена проверка на положительность.

## 1.8 Задание №8

Решить задачу 4, используя для ее решения конструкции `IF`, `COND`, `AND/OR`.

```

1 ;; if
2 (defun is-between-if (a b c)
3   (if (> a b) (< a c) (> a c)))
4
5 ;; cond
6 (defun is-between-cond (a b c)
7   (cond
8     ((> a b) (< a c))
9     ((< a b) (> a c))))
10
11 ;; and/or
12 (defun is-between (a b c)
13   (or (and (> a b) (< a c)) (and (> a c) (< a b))))

```

## 1.9 Задание №9

Переписать функцию how-alike, приведенную в лекции и использующую COND, используя только конструкции IF, AND/OR.

```

1 ;; cond
2 (defun how-alike (x y)
3   (cond ((or (= x y) (equal x y)) 'the_same)
4         ((and (oddp x) (oddp y)) 'both_odd)
5         ((and (evenp x) (evenp y)) 'both_even)
6         (t 'difference)))
7
8 ;; if
9 (defun how-alike-if (x y)
10   (if (or (= x y) (equal x y)) 'the_same
11       (if (and (oddp x) (oddp y)) 'both_odd
12           (if (and (evenp x) (evenp y)) 'both_even
13               'difference))))
14
15 ;; and/or
16 (defun how-alike-and-or (x y)
17   (or (and (or (= x y) (equal x y)) 'the_same)
18       (and (and (oddp x) (oddp y)) 'both_odd)
19       (and (and (evenp x) (evenp y)) 'both_even)
20       'difference))

```

## 2 Теоретические вопросы

### 2.1 Базис языка Lisp

Базис языка — минимальный набор конструкций и структур данных, с помощью которого можно решить любую задачу.

Базис Lisp образуют:

- атомы;
- структуры;
- базовые функции (atom, eq, cons, car, cdr);
- базовые специальные функции и функционалы (cond, quote, lambda, eval).

### 2.2 Классификация функций

Функции классифицируются на:

- чистые функции;
- рекурсивные;
- специальные функции или формы;
- псевдофункции;
- функции с вариантами значений;
- функционалы.

Классификация базовых функций:

- селекторы (car, cdr);
- конструкторы (cons, list);
- предикаты (null, consp, ...);

### 2.3 Способы создания функций

Лямбда определения:

```
1 (lambda <lambda-список> <форма>) ; lambda-выражение
2 ;; <lambda-список> -- список аргументов
3 ;; <форма> -- тело функции
```

Определение функций с именем:

```
1 (defun <имя> <lambda-выражение>)
```

## 2.4 Работа функций cond, if, and/or

### 2.4.1 Функция cond

Синтаксис:

```
1 (cond
2   (test1 body1)
3   (test2 body2)
4   ...
5   (testN bodyN)
6   [(T else-body)])
```

По порядку вычисляются и проверяются на равенство с Nil предикаты. Для первого предиката, который не равен Nil, вычисляется находящееся с ним в списке выражение и возвращается его значение. Если все предкаты вернут Nil, то и cond вернет Nil. Ветка «else» организуется явным указанием в качестве test — T.

### 2.4.2 Функция if

Синтаксис:

```
1 (if test t-body f-body)
```

Если вычисленный предикат не Nil, то выполняется t-body, иначе — f-body.

### 2.4.3 Функция and

Синтаксис:

```
1 (and arg1 arg2 ... argN)
```

Функция возвращает Nil при встрече первого (при вычислении слева направо) аргумента со значением Nil. Если все не Nil, то возвращается результат вычисления последнего аргумента.

## 2.4.4 Функция or

Синтаксис:

```
1 (or arg1 arg2 ... argN)
```

Функция возвращает первый `arg_i`, результат вычисления которого не `Nil`. Если все `Nil`, то возвращается `Nil`.