



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»

НАПРАВЛЕНИЕ ПОДГОТОВКИ _____ «09.03.04 Программная инженерия»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

Название: _____ Работа интерпретатора Lisp

Дисциплина: _____ Функциональное и логическое программирование

Студент	ИУ7-66Б	_____	А.Д. Ковель
	Группа	Подпись, дата	И. О. Фамилия

Преподаватель	_____	Н. Б. Толпинская
---------------	-------	------------------

Преподаватель	_____	Ю. В. Строганов
	Подпись, дата	И. О. Фамилия

Москва, 2023 г.

1 Практические задания

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

```
1      (defun make-even (x)
2        (
3          if (evenp x)
4            x
5            (if (> x 0)
6              (- x 1)
7              (+ x 1)
8            )
9        )
10     )
```

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

```
1      (defun calc-plus-one (x)
2        (
3          if (< x 0)
4            (- x 1)
5            (+ x 1)
6          )
7      )
8  )
```

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

```
1      (defun make-two-list (a b)
2        (
3          if (< a b)
4            (list a b)
5            (list b a)
6          )
7      )
8  )
```

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

```
1 (defun is-first-between (first second third)
2   (
3     if (or
4         (and (> first second) (< first third))
5         (and (> first third) (< first second)))
6       )
7     t
8     nil
9   )
10 )
11 )
```

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

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

```
1 (defun not-less (x y)
2   (
3     >= x y
4   )
5 )
```

7. Какой из следующих двух вариантов предиката ошибочен и почему?

- 1) (defun pred1 (x) (and (numberp x) (plusp x)))
- 2) (defun pred2 (x) (and (plusp x)(numberp x)))

Ответ: ошибочен вариант 2. Так как `and` вычисляет аргументы до тех пор, пока не будет ясно, какой ответ надо вернуть. `pred2` при первой проверке вернет `NIL` и завершит работы не вызывая `plusp`.

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

```
1 (defun x-from-y-to-z (x y z)
2   (if (< y x)
3     (if (< x z)
4       T Nil)
5     Nil))
```

```
1 (defun x-from-y-to-z (x y z)
2   (cond ((< y x) (cond ((< x z) T) (T Nil))) (T Nil)))
```

```
1 (defun x-from-y-to-z (x y z)
2   (and (< y x) (< x z)))
```

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

```
1 (defun how_alike (x y)
2   (cond ((or (= x y) (equal x y)) 'the_same)
3         ((and (oddp x) (oddp y)) 'both_odd)
4         ((and (evenp x) (evenp y)) 'both_even)
5         (T 'difference))
6   ))
```

```
1 (defun how_alike (x y)
2   (if (if (= x y)
3         (equal x y))
4       'the_same
5       (if (if (oddp x)
6               (oddp y))
7           'both_odd
8           (if (if (evenp x)
9                   (evenp y))
10              'both_even
11              'difference)
12         ))))
```

```
1 (defun how_alike (x y)
2 (or
3   (and (= x y) (equal x y) 'the_same)
4   (and (oddp x) (oddp y) 'both_odd)
5   (and (evenp x) (evenp y) 'both_even)
6   'difference
7 ))
```