



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»

на тему: «Работа интерпретатора Lisp»

Студент ИУ7-61Б
(Группа)

(Подпись, дата)

Д. В. Шубенина
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Н. Б. Толпинская
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Ю. В. Строганов
(И. О. Фамилия)

2024 г.

СОДЕРЖАНИЕ

1	Практические задания	3
1.1	Задание 1	3
1.2	Задание 2	3
1.3	Задание 3	3
1.4	Задание 4	3
1.5	Задание 5	4
1.6	Задание 6	4
1.7	Задание 7	4
1.8	Задание 8	4
1.9	Задание 9	5

1 Практические задания

1.1 Задание 1

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

```
1 | (defun evenGe (x)
2 |   (if (evenp x)
3 |       x
4 |       (+ x 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 oneMore (x)
2 |   (if (< x 0)
3 |       (- x 1)
4 |       (+ x 1)))
```

1.3 Задание 3

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

```
1 | (defun ordered (x y)
2 |   (if (< x y)
3 |       (cons x
4 |           (cons y Nil))
5 |       (cons y
6 |           (cons x Nil))))
```

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 between (x y z)
2 |   (or (<= y x z)
3 |       (<= z x y)))
```

1.5 Задание 5

Каков результат вычисления следующих выражений?

Выражение	Результат
(and 'fee 'fie 'foe)	FOE
(or nil 'fie 'foe)	FIE
(and (equal 'abc 'abc) 'yes)	YES
(or 'fee 'fie 'foe)	FEE
(and nil 'fie 'foe)	NIL
(or (equal 'abc 'abc) 'yes)	T

1.6 Задание 6

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

```
1 | (defun gep (x y)
2 |   (>= x y))
```

1.7 Задание 7

Какой из следующих двух вариантов предиката ошибочен и почему?

```
(defun pred1 (x)                                (defun pred2 (x)
  (and (numberp x) (plusp x)))                  (and (plusp x) (numberp x)))
```

Неправильным является второй вариант, т. к. в нем сначала проверяется знак аргумента (предикат **plusp**) и только потом, является ли аргумент числом. Предикат **plusp** принимает только числовые аргументы, поэтому при вызове **pred2** с нечисловым аргументом интерпретатор выдаст ошибку.

1.8 Задание 8

Решить задачу 4, используя для ее решения конструкции: только IF, только COND, только AND/OR (решение приведено в пункте 1.4).

Листинг 1.1 – Только IF

```
1 | (defun between-if (x y z)
2 |   (if (<= x y)
```

```

3 |         (>= x z)
4 |         (<= x z)))

```

Листинг 1.2 – Только COND

```

1 | (defun between-cond (x y z)
2 |   (cond ((<= x y) (<= z x))
3 |         ((<= x z) (<= y x))))

```

1.9 Задание 9

Переписать функцию how-alike, приведенную в лекции и использующую COND, используя только конструкции IF, 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)))

```

Листинг 1.3 – Только IF

```

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

```

Листинг 1.4 – Только AND/OR

```

1 | (defun how_alike_andor (x y)
2 |   (or (and (or (= x y) (equal x y)) 'the_same)
3 |       (and (and (oddp x) (oddp y)) 'both_odd)
4 |       (and (and (evenp x) (evenp y)) 'both_even)
5 |       (and T 'difference)))

```