



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)

Постнов С. А.
(Фамилия И. О.)

Преподаватель

(Подпись, дата)

Толпинская Н. Б.
(Фамилия И. О.)

Преподаватель

(Подпись, дата)

Строганов Ю.В.
(Фамилия И. О.)

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	4
1.5	Задание 5	4
1.6	Задание 6	4
1.7	Задание 7	5
1.8	Задание 8	5
1.9	Задание 9	6

1 Практические задания

1.1 Задание 1

В листинге 1.1 представлена функция, которая принимает целое число и возвращает первое четное число, не меньшее аргумента.

Листинг 1.1 – Функция, которая возвращает первое четное число, не меньшее аргумента

```
1 (defun f (x)
2   (cond ((oddp x) (+ x 1))
3         (t x)))
4
5 (f 13) ;; 14
6 (f 40) ;; 40
```

1.2 Задание 2

В листинге 1.2 представлена функция, которая принимает число и возвращает число того же знака, но с модулем на 1 больше модуля аргумента.

Листинг 1.2 – Функция, которая возвращает число на 1 большее по модулю

```
1 (defun f (x)
2   (cond ((> x 0) (+ 1 x))
3         (t (+ x -1))))
4
5 (f -13) ;; -14
6 (f 12)  ;; 13
```

1.3 Задание 3

В листинге 1.3 представлена функция, которая принимает два числа и возвращает список из этих чисел, расположенных по возрастанию.

Листинг 1.3 – Функция, которая возвращает список из чисел, расположенных по возрастанию

```
1 (defun f (x y)
2   (cond ((< x y) (cons x (cons y Nil)))
3         (t (cons y (cons x Nil)))))
4
5 (f 12 13) ;; 12 13
6 (f 13 12) ;; 12 13
```

1.4 Задание 4

В листинге 1.4 представлена функция, которая принимает три числа и возвращает Т только тогда, когда первое число расположено между вторым и третьим.

Листинг 1.4 – Функция, которая возвращает Т только тогда, когда первое число расположено между вторым и третьим

```
1 (defun f (x y z)
2   (or (and (< x y) (> x z)) (and (> x y) (< x z))))
3
4 (f 12 -3 13) ;; T
5 (f 13 -3 13) ;; Nil
6 (f 13 13 13) ;; Nil
```

1.5 Задание 5

В листинге 1.5 представлен результат вычисления выражений.

Листинг 1.5 – Результат вычисления 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

В листинге 1.6 представлен предикат, который принимает два числа-аргумента и возвращает Т, если первое число не меньше второго.

Листинг 1.6 – Предикат, который возвращает Т, если первое число не меньше второго

```
1 (defun f (a b)
2   (or (> a b) (= a b)))
3
4 (f -13 13)) ;; NIL
5 (f 13 13)) ;; T
6 (f 13 -13)) ;; T
```

1.7 Задание 7

В листинге 1.7 представлены предикаты `pred1` и `pred2`.

Листинг 1.7 – Предикаты `pred1` и `pred2`

```
1 (defun pred1 (x)
2   (and (numberp x) (plusp x)))
3
4 (defun pred2 (x)
5   (and (plusp x) (numberp x)))
6
7 (pred1 13)    ;; T
8 (pred1 -13)   ;; NIL
9 (pred1 'g)    ;; NIL
10
11 (pred2 13)    ;; T
12 (pred2 -13)   ;; NIL
13 (pred2 'g)    ;; Ошибка
```

1.8 Задание 8

В листинге 1.8 представлено решение задачи 1.4, используя для ее решения конструкции: только IF, только COND, только AND/OR.

Листинг 1.8 – Решение задачи 1.4, при помощи только IF, только COND, только AND/OR

```
1 (defun cond_solution (x y z)
2   (cond ((> x y) (< x z))
3         ((> x z) (< x y))))
4
5 (defun if_solution (x y z)
6   (if (< y x)
7       (if (< x z) t Nil)
8       (if (< x y)
9           (if (< z x) t Nil))))
10
11 (defun and_or_solution (x y z)
12   (or (and (< x y) (> x z)) (and (> x y) (< x z))))
```

1.9 Задание 9

В листинге 1.9 представлена функция `how-alike`, использующая только конструкции IF, AND/OR.

Листинг 1.9 – Функция `how-alike`, использующая только конструкции IF, AND/OR

```
1 (defun how_alike (x y)
2   (if (or (= x y) (equal x y))
3       'the_same
4       (if (and (oddp x) (oddp y))
5           'both_odd
6           (if (and (evenp x) (evenp y))
7               'both_even
8               'difference))))
9
10 (how_alike 13 13) ;; THE_SAME
11 (how_alike 12 14) ;; BOTH_EVEN
12 (how_alike 13 14) ;; DIFFERENCE
13 (how_alike 13 15) ;; BOTH_ODD
```