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 №6

по курсу «Функциональное и логическое программирование»

на тему: «Рекурсивные функции»

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.

1 Практические задания

1.1 Задание 1

В листинге 1.1 представлена хвостовая рекурсивная функция `my-reverse`, которая развернет верхний уровень своего списка - аргумента `lst`.

Листинг 1.1 – Хвостовая рекурсивная функция `my-reverse`

```
(defun my-reverse(lst res)
       (cond
            ((null lst)
                        res)
            (t
                        (my_reverse (cdr lst)
                                    (cons (car lst)
                                          res)))))
```

1.2 Задание 2

В листинге 1.2 представлена функция, которая возвращает первый элемент списка - аргумента, который сам является непустым списком.

Листинг 1.2 – Функция, которая возвращает первый элемент списка - аргумента

```
(defun f (lst)
       (cond
            ((null lst)
                                Nil)
            ((listp (car lst))
                                (car lst))
            (t
                                (f (cdr lst)))))
```

1.3 Задание 3

В листинге 1.3 представлена функция, которая выбирает из заданного списка только те числа, которые больше 1 и меньше 10.

Листинг 1.3 – Функция, которая выбирает из заданного списка только те числа, которые больше 1 и меньше 10

```
(defun select-nums (lst)
       (cond ((null lst)
                                    Nil)
             ((listp (car lst))
                                    (select-nums (car lst)))
             ((< 1 (car lst) 10)
                                    (cons (car lst)
                                          (select-nums (cdr
                                             lst))))
             (t
                                    (select-nums (cdr lst)))))
```

1.4 Задание 4

В листинге 1.4 представлена функция, которая умножает на заданное число - аргумент все числа из заданного списка - аргумента, когда:

1) все элементы списка — числа;

2) элементы списка — любые объекты.

Листинг 1.4 – Функция, которая умножает на заданное число - аргумент все числа из заданного списка - аргумента

```
(defun multi (lst n)
       (cond
            ((null lst)
                                Nil)
            ((listp (car lst))
                                (multi (car lst) n))
            ((numberp (car lst))
                                (cons (* (car lst) n)
                                      (multi (cdr lst) n)))
            (t
                                (cons (car lst)
                                      (multi (cdr lst) n)))))
```

1.5 Задание 5

В листинге 1.5 представлена функция `select-between`, которая из списка - аргумента, содержащего только числа, выбирает только те, которые расположены между двумя указанными числами — границами - аргумента и возвращает их в виде списка (упорядоченного по возрастанию).

Листинг 1.5 – Функция `select-between`

```
(defun select-between (lst a b)
      (cond
           ((null lst)
                                    Nil)
           ((listp (car lst))
                                    (select-between (car lst) a
                                       b))
           ((and (< a (car lst)) (> b (car lst)))
                                    (cons (car lst)
                                          (select-between (cdr
                                             lst) a b)))
           (t
                                    (select-between (cdr lst) a
                                       b))))
```

1.6 Задание 6

В листинге 1.6 представлена рекурсивная версия (с именем `rec-add`) вычисления суммы чисел заданного списка:

1) одноуровневого смешанного;

2) структурированного.

Листинг 1.6 – Функция `rec-add`

```
(defun rec-add (lst res)
       (cond
            ((null lst) res)
            ((listp (car lst))
                              (rec-add (car lst) res))
            ((numberp (car lst))
                              (rec-add (cdr lst) (+ res (car
                                 lst))))
            (t (rec-add (cdr lst) res))))
```

1.7 Задание 7

В листинге 1.7 представлена рекурсивная версия с именем `recnth` функции `nth`.

Листинг 1.7 – Функция `recnth`

```
(defun recnth (n lst)
       (cond
            ((null lst)
                              Nil)
            ((< (length lst) (- n 1))
                              Nil)
            ((= n 0)
                              (car lst))
            (t
                              (recnth (- n 1) (cdr lst)))))
```

1.8 Задание 8

В листинге 1.8 представлена рекурсивная функция `allodd`, которая возвращает `t`, когда все элементы списка нечетные.

Листинг 1.8 – Функция `allodd`

```
(defun allodd (lst)
       (cond
            ((null lst)
                         t)
            ((listp (car lst))
                         (allodd (car lst)))
            ((oddp (car lst))
                         (allodd (cdr lst)))
            (t
                         Nil)))
```

1.9 Задание 9

В листинге 1.9 представлена рекурсивная функция, которая возвращает первое нечетное число из списка, возможно, создавая некоторые вспомогательные функции.

Листинг 1.9 – Функция, которая возвращает первое нечетное число из списка

```
(defun first-odd (lst)
       (cond
            ((null lst)
                                  Nil)
            ((listp (car lst))
                                  (first-odd (car lst)))
            ((oddp (car lst))
                                  (car lst))
            (t
                                  (first-odd (cdr lst)))))
```

1.10 Задание 10

В листинге 1.10 представлена функция, которая получает как аргумент список чисел, а возвращает список квадратов этих чисел в том же порядке.

Листинг 1.10 – Функция, которая возвращает список квадратов переданных чисел

```
(defun pow-2 (lst)
       (cond
            ((null lst)
                        Nil)
            (t
                        (cons (* (car lst) (car lst))
                              (pow-2 (cdr lst))))))
```