



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»

Отчёт по лабораторной работе №6.1
по дисциплине «Функциональное и логическое
программирование»

Тема: Рекурсивные функции

Студент: Княжев А. В.

Группа: ИУ7-62Б

Оценка (баллы): _____

Преподаватели: Толшинская Н. Б., Строганов Ю. В.

Москва — 2023 г.

1. Практическая часть

1.1. Задание 1

Задание

Написать хвостовую рекурсивную функцию `my-reverse`, которая развернет верхний уровень своего списка-аргумента `lst`.

Решение

```
(defun my-reverse1 (lst res)
  (cond
    ((null lst) res)
    (t (my-reverse1 (cdr lst) (cons (car lst) res))))
  )
)

(defun my-reverse (lst) (my-reverse1 lst NIL))
```

1.2. Задание 2

Задание

Написать функцию, которая возвращает первый элемент списка — аргумента, который сам является непустым списком.

Решение

```
(defun get_atom (x)
  (cond
    ((atom x) x)
    (t (get_atom (car x)))
  )
)

(defun f1 (x)
  (cond
    ((null x) NIL)
    ((atom (car x)) (f1 (cdr x)))
    (t (get_atom (car x)))
  )
)
```

1.3. Задание 3

Задание

Напишите рекурсивную функцию, которая умножает на заданное число-аргумент все числа из заданного списка-аргумента, когда

1. все элементы списка — числа;
2. элементы списка — любые объекты.

Решение

```
(defun f21 (x n res)
  (cond
    ((null x) res)
    (t (f21 (cdr x) n (cons (* (car x) n) res))))
  )
)

(defun f2 (x n)
  (f21 x n NIL)
)
```

```
(defun f31 (x n res)
  (cond
    ((null x) res)
    ((numberp (car x)) (f31 (cdr x) n (cons (* (car x) n) res)))
    ((atom (car x)) (f31 (cdr x) n (cons (car x) res)))
    (t (f31 (cdr x) n (cons (f31 (car x) n NIL) res)))
  )
)

(defun f3 (x n)
  (f31 x n NIL)
)
```

1.4. Задание 4

Задание

Напишите функцию, **select-between**, которая из списка-аргумента, содержащего только числа, выбирает только те, которые расположены между двумя указанными числами — границами-аргументами и возвращает их в виде списка.

Решение

```
(defun select-between1 (x a b)
  (cond
    ((null x) NIL)
    ((< a (car x) b) (cons (car x) (select-between1 (cdr x) a b)))
    (t (select-between1 (cdr x) a b))
  )
)

(defun select-between (x a b)
  (cond
    ((< a b) (select-between1 x a b))
    (t (select-between1 x b a))
  )
)
```

1.5. Задание 5

Задание

Написать рекурсивную версию (с именем `rec-add`) вычисления суммы чисел заданного списка:

1. одноуровневого смешанного;
2. структурированного.

Решение

```
(defun rec-add1 (x res)
  (cond
    ((null x) res)
    ((numberp (car x)) (rec-add (cdr x) (+ (car x) res)))
    (t (rec-add (cdr x) res)))
  )
)

(defun rec-add (x)
  (rec-add1 x NIL)
)
```

```
(defun rec-add (x)
  (cond
    ((null x) 0)
    ((numberp (car x)) (+ (car x) (rec-add (cdr x))))
    ((atom (car x)) (rec-add (cdr x)))
    (t (+ (rec-add (car x)) (rec-add (cdr x)))))
  )
)
```

Листинг 1..1: Some Code

```
1 (cond 1 2)
```