



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 дисциплине "Функциональное и логическое программирование"

Студент Кузнецова А. В.

Группа ИУ7-61Б

Оценка (баллы) _____

Преподаватели Толпинская Н. Б., Строганов Ю. В.

Москва — 2023 г.

1 Практические задания

Используя рекурсию:

1.1 Задание №1

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

```
1 (defun move-to (lst res)
2   (cond ((null lst) res)
3         (T (move-to (cdr lst) (cons (car lst) res)))))
4
5 (defun my-reverse (lst)
6   (move-to lst ()))
```

1.2 Задание №2

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

```
1 (defun get-first-list-el (lst)
2   (cond ((null lst) Nil)
3         ((and (car lst) (listp (car lst))) (caar lst))
4         (T (get-first-list-el (cdr lst)))))
```

1.3 Задание №3

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

а) все элементы списка — числа,

```
1 (defun mul-nums-res (lst num res)
2   (cond ((null lst) res)
```

```

3          (T (mul-nums-res (cdr lst) num (cons (* (car lst)
4              num) res)))))
5  (defun mul-nums (lst num)
6      (mul-nums-res lst num ()))

```

6) элементы списка — любые объекты.

```

1  (defun mn (lst num res)
2      (cond ((null lst) res)
3            ((numberp (car lst))
4              (mn (cdr lst) num (cons (* (car lst) num) res)))
5            ((atom (car lst)) (mn (cdr lst) num (cons (car lst)
6              res))))
7      (T (mn (cdr lst) num (cons (mn (car lst) num ())
8              res)))))
9  (defun mul-num (lst num)
      (mn lst num ()))

```

1.4 Задание №4

Напишите функцию, `select-between`, которая из списка-аргумента, содержащего только числа, выбирает только те, которые расположены между двумя указанными границами аргументами и возвращает их в виде списка (упорядоченного по возрастанию списка чисел (+ 2 балла)).

```

1  (defun sb (lst begin end res)
2      (cond ((null lst) res)
3            ((or (< begin (car lst) end) (< end (car lst) begin))
4              (sb (cdr lst) begin end (cons (car lst) res)))
5            (T (sb (cdr lst) begin end res))))
6
7  (defun select-between (lst begin end)
8      (sort (sb lst begin end ()) '<))

```

1.5 Задание №5

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

а) одноуровневого смешанного,

```
1  (defun ra (lst res)
2    (cond ((null lst) res)
3          ((numberp (car lst)) (ra (cdr lst) (+ (car lst) res)))
4          (T (ra (cdr lst) res))))
5
6  (defun rec-add (lst)
7    (ra lst 0))
```

б) структурированного.

```
1  (defun ra (lst res)
2    (cond ((null lst) res)
3          ((numberp (car lst)) (ra (cdr lst) (+ (car lst) res)))
4          ((atom (car lst)) (ra (cdr lst) res))
5          (T (ra (cdr lst) (ra (car lst) res)))))
6
7  (defun rec-add (lst)
8    (ra lst 0))
```