



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»

О Т Ч Е Т

по лабораторной работе № 6

Дисциплина: ***Функциональное и логическое программирование***

Студент

ИУ7И-64Б

(Группа)

(Подпись, дата)

Динь Вьет Ань

(И.О. Фамилия)

Преподаватель

Толпинская Н. Б.

(Подпись, дата)

(И.О. Фамилия)

Москва, 2023

Задание 1. Написать хвостовую рекурсивную функцию my-reverse, которая развернет верхний уровень своего списка-аргумента lst.

```
(defun move (lst res)
  (cond ((null lst) res)
        (T (move (cdr lst) (cons (car lst) res))))
  )
)
```

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

Задание 2. Написать функцию, которая возвращает первый элемент списка - аргумента, который сам является непустым списком.

```
(defun work2 (lst)
  (cond ((atom lst) lst)
        ((atom (car lst)) work2 (cdr lst))
        (t (car lst)))
  )
)
```

Задание 3. Напишите рекурсивную функцию, которая умножает на заданное число-аргумент все числа из заданного списка-аргумента, когда

а) все элементы списка --- числа

```
(defun work3_a (num lst)
  (cond ((null lst) nil)
        (T (cons (* num (car lst)) (work3_a num (cdr lst)))))
  )
)
```

b) элементы списка -- любые объекты.

```
(defun work3_b (num lst)
  (cond ((null lst) nil)
        ((numberp (car lst)) (cons (* num (car lst)) (work3_a num (cdr lst))))
        (T (cons (car lst) (work3_a num (cdr lst)))))
  )
)
```

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)).

```
(defun work4 (a b lst)
  (cond ((null lst) nil)
        ((and (< a (car lst)) (< (car lst) b)) (sort (cons (car lst) (work4 a b (cdr lst))) #'<))
        (T (sort (work4 a b (cdr lst)) #'<)))
  )
)
```

Задание 5. Написать рекурсивную версию (с именем rec-add) вычисления суммы чисел заданного списка:

a) одноуровневого смешанного

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

б) структурированного.

```
(defun work5 (lst res)
  (cond ((null lst) res)
        ((numberp (car lst)) (work5 (cdr lst) (+ res (car lst))))
        ((atom (car lst)) (work5 (cdr lst) res))
        (T (work5 (cdr lst) (work5 (car lst) res))))
  )
)
```