

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра автоматизированных систем управления (АСУ)

Обработка списков
Лабораторная работа №1
по дисциплине
«Функциональное и логическое программирование»

Студент гр. 430-2

_____ А.А. Лузинсан

«____» _____ 2023 г.

Руководитель

Доцент кафедры АСУ

_____ С.М. Алфёров

«____» _____ 2023 г.

Томск 2023

Оглавление

Введение.....	3
1 Ход работы.....	4
1.1 Задание 1.....	4
1.2 Задание 2.....	4
1.3 Задание 3.....	4
1.4 Задание по варианту.....	5
Заключение.....	6

Введение

Цель: получить опыт обработки списков в LISP.

Задание №1: напишите функцию, вычисляющую предпоследний элемент списка

Задание №2: напишите функцию (fulllength x), считающую полное количество атомов (не равных nil) в списке x. Исходный список может быть многоуровневым.

Задание №3: напишите функцию от двух аргументов x и n , которая создает список из n раз повторенных списков ,состоящих из одного элемента x.

Задание №4, вариант 7: объединить позиции с одинаковым наименованием товаров, количество просуммировать, информацию о скидке и цене взять из первой позиции данного наименования. Таким образом сформировать новый список чека.

1 Ход работы

1.1 Задание 1

Входные данные: (3 8 92 847 35 82)

Полученный результат: 35

Код программы представлен в листинге 1.1.

Листинг 1.1 — код функции prelast и её вызов

```
(defun prelast(arr)
  (car (cdr (reverse arr))))
(print (prelast `(3 8 92 847 35 82)))
```

1.2 Задание 2

Входные данные: (2 7 8 2 3)

Полученный результат: 5

Код программы представлен в листинге 1.1.

Листинг 1.2 — код функции fulllength и её вызов

```
(defun fulllength(x)
  ( cond ((null x) 0)
        ((atom x) 1)
        (t (+ (fulllength (car x))
               (fulllength (cdr x))
              )
        )
  ))
(print (fulllength `(2 7 8 2 3)))
```

1.3 Задание 3

Входные данные: 3 5

Полученный результат (3 3 3 3 3)

Код программы представлен в листинге 1.1.

Листинг 1.3 — код функции mlst и её вызов

```
(defun mlst(x n)
  ( cond ((> n 0) (cons (list x) (mlst x (1- n))))
        (t nil)
  ))
(print (mlst 3 5))
```

1.4 Задание по варианту

Входные данные: ((item1 2 10.0 0.0) (item2 1 5.0 0.0) (item1 1 10.0 0.0) (item3 2 15.0 0.0))

Код программы представлен в листинге 1.4.

Листинг 1.4 — код функции `mlst` и её вызов

```
(defun merge-check-items (check)
  (if (null check)
      nil
      (let ((item (car check))
            (rest (cdr check)))
        (let ((name (first item))
              (quantity (second item))
              (price (third item))
              (discount (fourth item)))
          (let ((merged-item
                (reduce (lambda (acc x)
                        (if (equal name (first x))
                            (list name (+ quantity (second x)) price discount)
                            acc))
                      rest
                      :initial-value (list name quantity price discount))))
            (cons merged-item (merge-check-items (remove-if (lambda (x) (equal name
                                                                    (first x))) rest)))))))

(defvar *check*
  '((item1 2 10.0 0.0)
    (item2 1 5.0 0.0)
    (item1 1 10.0 0.0)
    (item3 2 15.0 0.0)))
(setq *check* (merge-check-items *check*))

; Выводим результат
(dolist (item *check*)
  (format t "Наименование: ~a, Количество: ~a, Цена: ~a, Скидка: ~a~%" (first
                                                                    item) (second item) (third item) (fourth item)))
```

Результат:

Наименование: ИТЕМ1, Количество: 3, Цена: 10.0, Скидка: 10.0
Наименование: ИТЕМ2, Количество: 1, Цена: 5.0, Скидка: 20.0
Наименование: ИТЕМ3, Количество: 2, Цена: 15.0, Скидка: 50.0

Заключение

В результате выполнения лабораторной работы я получила опыт обработки списков в LISP, изучила и использовала основные функции языка LISP, а также выполнила задание по варианту.