

Отчет по лабораторной работе №3 по курсу «Функциональное программирование»

Студент группы 8О-306 Кондратьев Егор, № по списку 14.

Контакты: egor.kondratev27@gmail.com

Работа выполнена: 12.04.2022

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806

Отчет сдан:

Итоговая оценка:

Подпись преподавателя:

1. Тема работы

Последовательности, массивы и управляющие конструкции Коммон Лисп.

2. Цель работы

Научиться создавать векторы и массивы для представления матриц, освоить общие функции работы с последовательностями, инструкции цикла и нелокального выхода.

3. Задание (вариант № 3.16)

Запрограммировать на языке Коммон Лисп функцию, принимающую два аргумента:

- A - двумерный массив, представляющий действительную матрицу,
- lis - список действительных чисел.

Функция должна возвращать новую матрицу, являющуюся копией A, но в которой заменены нулями элементы с четной суммой индексов, встречающиеся в списке lis.

Исходный массив A должен оставаться неизменным.

4. Оборудование студента

Ноутбук Asus ROG GL752VW, процессор QuadCore Intel Core i7-6700HQ, 3100 MHz , память 24474 МБ (DDR4 SDRAM), 64-разрядная система.

5. Программное обеспечение

ОС Windows 10, программа VSC

6. Идея, метод, алгоритм

Создаем новый массив ответов той же размерности, что и оригинальный массив. Проходим по оригинальному двумерному массиву рекурсивно, если сумма индексов элементов четна и элемент от этих индексов присутствует в списке, то записываем в новый массив под этими индексами ноль, иначе копируем элемент из изначального массива. Проходимся по списку рекурсивно, проверяя головной элемент списка и если он не подходит, то запускаем эту же функцию от хвоста списка.

7. Сценарий выполнения работы

8. Распечатка программы и ее результаты

Программа

;;; lab3 Egor Kondratev

;;; 3.16

```
(defun mprint (m)
  (loop for i below (car (array-dimensions m)) do
    (loop for j below (cadr (array-dimensions m)) do
      (let ((cell (aref m i j)))
        (format t "~a " cell)))
      (format t "~%"))))
```

```
(defun findel (l elem)
  (when l
    (if (= (car l) elem)
      t
      (findel (cdr l) elem))))
```

```
(defun relocation (ans arr lst n m i j)
  (if (< i n)
    (if (< j m)
      (progn
        (if (and (= (rem (+ i j) 2) 0) (findel lst (aref arr i j)))
          (setf (aref ans i j) 0)
          (setf (aref ans i j) (aref arr i j)))
        (relocation ans arr lst n m i (+ 1 j)))
      (relocation ans arr lst n m (+ 1 i) 0)
    )
    t))
```

```
(defun func (A lis)
  (let* ((n (car (array-dimensions A)))
        (m (car (cdr (array-dimensions A)))))
    (ans (make-array (list n m)))
    (relocation ans A lis n m 0 0)
    ans))
```

Результаты

```
* (mprint (func #2A((1 2 3) (4 5 6) (7 8 9)) '(1 3 5 7 8 9)))
```

```
0 2 0
```

```
4 0 6
```

```
0 8 0
```

```
*(mprint (func #2A((1 2 3 4) (5 6 7 8) (9 10 11 12)) '(1 6 8 9 11)))
```

```
0 2 3 4
```

```
5 0 7 0
```

```
0 10 0 12
```

9. Дневник отладки

№	Дата, время	Событие	Действие по исправлению	Примечание
1				

10. Замечания автора по существу работы

11. Выводы

Выполняя эту работу, я улучшил навыки работы с массивами и списками. Для просмотра матрицы лучше воспользоваться дополнительной функцией печати, которая выводит двумерный массив в более привычном формате.