

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Ганыгин Максим Олегович  
Должность: и.о. декана факультета фундаментальной и прикладной информатики  
Дата подписания: 21.09.2023 12:56:21  
Уникальный программный ключ:  
65ab2aa0d384efe8480e6a4c688eddbc475e411a

## МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра программной инженерии

УТВЕРЖДАЮ

Проректор по учебной работе

« 24 » 12



### ОРГАНИЗАЦИЯ РЕКУРСИВНЫХ ВЫЧИСЛЕНИЙ НА ЯЗЫКЕ LISP

Методические указания по выполнению лабораторной работы по  
дисциплине «Функциональное и логическое программирование» для  
студентов направления подготовки 09.03.04 «Программная инженерия»

Курск 2017

УДК 004.652

Составители: В.Г. Белов, Т.М. Белова

Рецензент

Кандидат технических наук, доцент кафедры программной инженерии ЮЗГУ И.Н. Ефремова

**Организация рекурсивных вычислений на языке LISP:** методические указания по выполнению лабораторной работы по дисциплине "Функциональное и логическое программирование" для студентов направления подготовки 09.03.04 "Программная инженерия" / Юго-Зап. гос. ун-т; сост.: В.Г. Белов, Т.М. Белова, – Курск, 2017. – 46 с.: ил. 20, табл. 6.

Рассмотрены методы составления рекурсивных определений предикатов с использованием нисходящей и восходящей рекурсий на языке LISP.

Материал предназначен для студентов направления подготовки 09.03.04 «Программная инженерия» специализации «Разработка программно-информационных систем»

Подписано в печать 94.12.17. Формат 60x84 1/16.  
Усл. печ. л. 2.4. Уч.-изд. л. 2,2 Тираж 100 экз. Заказ 4379. Бесплатно.  
Юго-Западный государственный университет

305040, Курск, ул.50 лет Октября, 94.

## Содержание

1 ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ .....	4
2 МЕТОДИКА СОСТАВЛЕНИЯ РЕКУРСИВНЫХ ОПРЕДЕЛЕНИЙ.....	5
2.1. Нисходящая рекурсия .....	5
2.2. Примеры составления определений с нисходящей рекурсией .....	5
2.3. Восходящая рекурсия .....	7
3. ВЫПОЛНЕНИЕ ЛАБОРАТОРНОЙ РАБОТЫ .....	12
3.1 Задачи на нисходящую рекурсию .....	12
3.2 Задачи на восходящую рекурсию .....	17
4. СОДЕРЖАНИЕ ОТЧЕТА .....	24
5. КОНТРОЛЬНЫЕ ВОПРОСЫ К ЗАЩИТЕ ЛАБОРАТОРНОЙ РАБОТЫ ....	25
Приложение .....	26

## **1 ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ**

Целью лабораторной работы является приобретение навыков в составлении рекурсивных определений функций с использованием нисходящей и восходящей рекурсий.

## 2 МЕТОДИКА СОСТАВЛЕНИЯ РЕКУРСИВНЫХ ОПРЕДЕЛЕНИЙ

### 2.1. Нисходящая рекурсия

2.1.1. Составить определения функции для терминальных случаев. Т.е. определить чему будет равно значение функции, когда список станет или будет пустым, когда число станет или будет равно 0 или когда останется или будет одноэлементный список.

2.1.2. Для нетерминального случая предположить, что уже имеется определение разрабатываемой функции. Используя вызов этого определения, примененный к остатку списка, и элемент, предшествующий этому остатку, составить выражение, которое вычисляет значение функции для этого случая.

2.1.3. Объединить 2.1.1. и 2.1.2. в одном условном выражении, причем, 2.1.1. после "то", а 2.1.2. после "если".

### 2.2. Примеры составления определений с нисходящей рекурсией

#### 2.2.1. Количество элементов в списке

Исходные данные: список LST.

Результат: значение функции DL.

##### 2.2.1.1. Терминальный случай

Список LST пуст, т.е. (NULL LST) имеет значение истина. В этом случае значение функции DL будет 0, так как в списке нет ни одного элемента.

##### 2.2.1.2. Нетерминальный случай

Предположим, что определение функции DL уже составлено и можно осуществлять его вызов. Тогда длина остатка списка (CDR LST) имеет значение (DL (CDR LST)).

Если рассмотреть элемент (CAR LST), который предшествует остатку списка (CDR LST), то длина всего списка LST будет иметь значение (+ 1 (DL (CDR LST))).

### 2.2.1.3. Объединение терминального и нетерминального случаев

(DEFUN DL (LST)

(COND

((NULL LST) 0) ; терминальный случай

(T(+ 1 (DL (CDR LST)))) ; нетерминальный случай.

### 2.2.2. Сумма элементов в списке чисел

Исходные данные: список чисел LST.

Результат: значение функции SUM.

#### 2.2.2.1. Терминальный случай

Список LST пуст, т.е. (NULL LST) имеет значение истина. В этом случае значение функции SUM будет 0, так как в списке нет ни одного числа.

#### 2.2.2.2. Нетерминальный случай

Предположим, что определение функции SUM уже составлено и можно осуществлять его вызов. Тогда сумма чисел остатка списка (CDR LST) имеет значение (SUM (CDR LST)).

Если рассмотреть элемент (CAR LST), который предшествует остатку списка (CDR LST), то сумма чисел всего списка LST будет иметь значение (+ (CAR LST) (SUM (CDR LST))).

### 2.2.2.3. Объединение терминального и нетерминального случаев

(DEFUN SUM (LST)

(COND

((NULL LST) 0) ; терминальный случай

(T(+ (CAR LST) (SUM (CDR LST)))))) ; нетерминальный случай

### 2.2.3. Максимальный элемент в списке чисел

Исходные данные: список чисел LST.

Результат: значение функции MAX.

#### 2.2.3.1. Терминальные случаи

Список LST пуст, т.е. (NULL LST) имеет значение истина. В этом случае значение функции MAX будет NIL, так как в списке нет ни одного числа. Список LST состоит из одного элемента, т.е. (NULL (CDR LST)) имеет значение истина. В этом случае значение функции MAX будет (CAR LST). Очевидно, что второй случай содержит в себе и первый.

#### 2.2.3.2. Нетерминальный случай

Предположим, что определение функции MAX уже составлено и можно осуществлять его вызов. Тогда максимальное значение среди чисел остатка списка (CDR LST) имеет значение (MAX (CDR LST)).

Если рассмотреть элемент (CAR LST), который предшествует остатку списка (CDR LST), то максимальное значение среди чисел всего списка LST будет иметь значение: (CAR LST), если (CAR LST) больше или равно (MAX (CDR LST)); или (MAX (CDR LST)), если (CAR LST) меньше (MAX (CDR LST)).

#### 2.2.3.3. Объединение терминального и нетерминального случаев

```
(DEFUN MAX (LST)
  (COND
    ((NULL (CDR LST)) (CAR LST))
    ((>= (CAR LST) (MAX (CDR LST))) (CAR LST))
    (T(MAX (CDR LST)))).
```

### 2.3. Восходящая рекурсия

2.3.1. Составляется список параметров в определении вспомогательной функции, в которую помимо основных параметров вводятся дополнительные параметры, в одном из которых формируется значение разрабатываемой функции.

2.3.2. Составляется определение вспомогательной функции для терминального случая (или случаев), в котором ее значение будет равно значению дополнительного параметра, в котором формируется результат.

2.3.3. Для нетерминального случая в определении вспомогательной функции записывается ее вызов, в котором значения дополнительных параметров определяются на основании их предыдущих значений (записывается имя параметра) и значения очередного элемента в списке.

2.3.4. Пункты 2.3.2. и 2.3.3. объединяются в одно условное выражение, причем, 2.3.2. после "то", а 2.3.3. после "если".

2.3.5. Составляется определение разрабатываемой функции, в которой осуществляется вызов вспомогательной функции с начальными значениями вспомогательных параметров.

## 2.4. Примеры составления определений с восходящей рекурсией

### 2.4.1. Количество элементов в списке

Исходные данные: список LST.

Результат: значение функции DLW.

#### 2.4.1.1. Список параметров вспомогательной функции DLWS

В качестве дополнительного параметра, в котором будет формироваться результат вспомогательной функции DLWS, будет использоваться L.

#### 2.4.1.2. Терминальный случай

Список LST пуст, т.е. (NULL LST) имеет значение истина. В этом случае значение функции DLWS будет L.

#### 2.4.1.3. Нетерминальный случай

Записываем вызов функции DLWS, в котором формируем новое значение для LST на основании старого: это будет (CDR LST); и новое значение для L на основании старого: это будет (+ 1 L).

Вызов функции будет иметь вид

(DLWS (CDR LST) (+ 1 L)) .

#### 2.4.1.4. Объединение терминальных и нетерминальных случаев

(DEFUN DLWS (LST L)

(COND

((NULL LST) L) ;терминальный случай

(T (DLWS (CDR LST) (+ 1 L)))) ; нетерминальный случай.

2.4.1.5. Определение функции DLW для вызова вспомогательной функции DLWS с начальным значением для L

(DEFUN DLW (LST)

(DLWS LST 0) ) .

#### 2.4.2. Сумма элементов в списке чисел

Исходные данные: список LST.

Результат: значение функции SUMW.

##### 2.4.2.1. Список параметров вспомогательной функции SUMWS

В качестве дополнительного параметра, в котором будет формироваться результат вспомогательной функции SUMWS, будет использоваться S.

##### 2.4.2.2. Терминальный случай

Список LST пуст, т.е. (NULL LST) имеет значение истина. В этом случае значение функции SUMWS будет S.

##### 2.4.2.3. Нетерминальный случай

Записываем вызов функции SUMWS, в котором формируем новое значение для LST на основании старого: это будет (CDR LST); и новое значение для L на основании старого: это будет (+ (CAR LST) S). Вызов функции будет иметь вид (SUMWS (CDR LST) (+ (CAR LST) S)) .

#### 2.4.2.4. Объединение терминальных и нетерминальных случаев (DEFUN SUMWS (LST S)

(COND

((NULL LST) S) ; терминальный случай

(T (SUMWS (CDR LST) (+ (CAR LST) S))))); нетерминальный случай.

#### 2.4.2.5. Определение функции SUMW для вызова вспомогательной функции DLWS с начальным значением для L

(DEFUN SUMW (LST)

(SUMWS LST 0) )

#### 2.4.3. Максимальный элемент в списке чисел

Исходные данные: список чисел LST.

Результат: значение функции MAXW.

##### 2.4.3.1. Список параметров вспомогательной функции MAXWS

В качестве дополнительного параметра, в котором будет формироваться результат вспомогательной функции MAXWS, будет использоваться MAX.

##### 2.4.3.2. Терминальный случай

Список LST пуст, т.е. (NULL LST) имеет значение истина. В этом случае значение функции MAXWS будет S.

##### 2.4.3.3. Нетерминальный случай

Записываем вызовы функции MAXWS, в которых формируем новое значение для LST на основании старого: это будет (CDR LST). Новое значение для MAX на основании старого будет:

- MAX, если MAX больше или равно (CAR LST);
- (CAR LST), если MAX меньше (CAR LST).

Вызов функции будет иметь вид

```
(( >= MAX (CAR LST)) (MAXWS (CDR LST) MAX))
(MAXWS (CDR LST) (CAR LST)) .
```

2.4.3.4. Объединение терминальных и нетерминальных случаев

```
(DEFUN MAXWS (LST MAX)
(COND
((NULL LST) MAX) ;терминальный случай
(( >= MAX (CAR LST)) (MAXWS (CDR LST) MAX))
(T (MAXWS (CDR LST) (CAR LST))))); нетерминальный случай.
```

2.4.3.5. Определение функции MAXW для вызова вспомогательной функции MAXWS с начальным значением для MAX

```
(DEFUN MAXW (LST)
((NULL LST) NIL)
(MAXWS LST (CAR LST)) )
```

Следует учесть, что при применении восходящей рекурсии в алгоритмах на построение различного вида списков результат, который формируется в дополнительном параметре в общем случае, называемым аккумулятором, будет получен в инверсном порядке. Это значит, что в терминальном случае аккумулятор должен быть реверсирован.

### 3. ВЫПОЛНЕНИЕ ЛАБОРАТОРНОЙ РАБОТЫ

После выполнения приведенных выше программ требуется разработать программы в соответствии с индивидуальным заданием список, которых приведен ниже.

#### 3.1 Задачи на нисходящую рекурсию

Задачи первого уровня сложности.

1. Дан список из 10 различных целых чисел. Найдите сумму чисел этого списка, расположенных между максимальным и минимальным числами (в сумму включить и оба эти числа).
2. Даны координаты  $n$  точек на плоскости:  $x_1, y_1, \dots, x_n, y_n$  ( $n=10$ ), которые являются вершинами ломаной. Найдите номера 2-х точек, расстояние между которыми наибольшее. Считайте, что такая пара точек единственная.
3. Дан список из 15 целых чисел, среди которых могут быть одинаковые числа. Минимальные элементы этого списка замените целой частью среднего арифметического всех элементов списка, остальные члены списка оставить без изменения.
4. Дан список из 10 различных целых чисел. Найдите произведение чисел в этом списке, расположенных между максимальным и минимальным числами (в произведение включить оба эти числа).
5. Дан целочисленный список  $a[15]$ . Найдите наименьшее и наибольшее из четных чисел в последовательности чисел  $a[15]$ . Предусмотрите ситуацию, когда четных чисел в последовательности нет.
6. Дан целочисленный список  $a[15]$ . Найдите наибольшее из нечетных чисел и количество четных чисел в списке  $a[15]$ . Предусмотрите ситуацию, когда нечетных чисел в последовательности нет.
7. Дана последовательность из 15 целых чисел. Найдите наибольшее и наименьшее числа в этой последовательности и поменяйте эти числа местами.
8. Дана последовательность из 15 вещественных чисел. Поменяйте наименьшее число в этой последовательности с первым числом и наибольшее число с последним числом.
9. Дан список вещественных чисел, содержащий 10 элементов. Если минимальный элемент или максимальный элемент этого списка равен 3.5,

то поменяйте эти элементы местами.

10. Дан список целых чисел, содержащий 10 элементов. Если минимальный элемент в этом списке находится на первом месте, а максимальный – на последнем месте, то поменяйте местами эти элементы.

11. Дан список из 15 вещественных чисел. Если минимальный элемент списка предшествует максимальному элементу, то подсчитайте сумму максимального и минимального элементов, иначе – вычислите разность.

12. Дан список из 15 вещественных чисел. Если максимальный элемент списка предшествует минимальному элементу, то подсчитайте произведение минимального и максимального элементов, иначе – среднее арифметическое.

13. Дан список  $b$  из 15 целых чисел. Найдите наименьшее из чисел, расположенных от начала списка до максимального элемента списка  $b$ .

14. Дан список из 10 целых чисел. Если минимальный элемент списка расположен на 4 месте, а максимальный – рядом (слева или справа), то подсчитайте среднее арифметическое максимального и минимального элементов списка.

15. Дан список из 15 целых чисел. Если минимальный элемент списка меньше максимального в 3 раза, то каждый элемент списка увеличьте в 3 раза.

16. Дан список  $b$  из 15 целых чисел. Найдите наибольшее из чисел, расположенных от минимального элемента до конца списка  $b$ .

17. Дан список из 15 целых чисел. Если минимальный или максимальный элемент равен 0, то обнулите элементы списка, предшествующие минимальному элементу в списке.

18. Дан список из 15 целых чисел. Если минимальный элемент меньше максимального в 2 раза, то все элементы, предшествующие максимальному элементу, установите равными минимальному элементу.

19. Дан список из 10 целых чисел. Если максимальный элемент находится на 2 месте, а минимальный элемент – на 5 месте, то все элементы, расположенные от минимального элемента и до конца списка, установите равными максимальному элементу.

20. Даны натуральное число  $n$  и целочисленный список  $a[15]$ . Определите среднее арифметическое тех чисел, которые кратны  $n$ .

21. Дан целочисленный список  $a[15]$ . Найдите наименьшее из чисел, значения которых кратны 3, в последовательности чисел  $a[15]$ . Предусмотрите ситуацию, когда таких чисел в последовательности нет.

22. Дан целочисленный список  $a[10]$ . Найдите наибольшее из чисел, значения которых кратны 5 и не кратны 7, в последовательности чисел  $a[10]$ . Предусмотрите ситуацию, когда таких чисел в последовательности нет.
23. Дан список  $b$  из 15 целых чисел. Найдите наибольшее из чисел, расположенных от начала списка до минимального элемента списка  $b$ .
24. Дан список из 10 вещественных чисел. Максимальный элемент списка увеличьте в 2 раза, а минимальный элемент уменьшите в 2 раза. Выведите преобразованный список.
25. Дан список из 15 вещественных чисел. Максимальный элемент списка замените суммой минимального и максимального элементов, а минимальный элемент – разностью максимального и минимального элементов. Выведите преобразованный список.
26. Дан список из 10 целых чисел, среди которых могут быть одинаковые числа. Установите все элементы, имеющие наименьшее значение, равными среднему арифметическому значению положительных элементов списка. Предусмотрите ситуацию, когда в списке положительных чисел нет.
27. Дан список из 15 целых чисел, значения которых лежат в диапазоне от 1 до 1000. Определите индекс первого элемента тройки элементов списка, сумма которых максимальна. Например, 1, 2 и 3 элементы, 4, 5 и 6 элементы и т.д.
28. Дан список из 20 целых чисел. Определите индекс первого элемента пары элементов списка, сумма которых максимальна. Например, 1 и 2 элементы, 3 и 4 элементы и т.д.
29. Дан список из 20 целых чисел. Определите индекс первого элемента пары элементов списка, произведение которых минимально. Например, 1 и 2 элементы, 3 и 4 элементы и т.д.
30. Дан список из 15 целых чисел, значения которых лежат в диапазоне от 1 до 1000. Определите индекс первого элемента тройки элементов списка, сумма которых минимальна. Например, 1, 2 и 3 элементы, 4, 5 и 6 элементы и т.д.

### Задачи второго уровня сложности.

1. Даны натуральное число  $n$  и целые числа  $a_1, \dots, a_n$ . Подсчитать максимальную длину последовательности подряд расположенных элементов, имеющих одинаковые значения.

2. Даны натуральное число  $n$  и целые числа  $a_1, \dots, a_n$ . Подсчитать максимальную длину последовательности подряд расположенных элементов, каждый из которых больше предыдущего.
3. Даны натуральное число  $n$  и целочисленный список  $a[n]$ , среди элементов которого есть одинаковые. Создайте список  $b$  из различных элементов списка  $a$ .
4. Даны натуральное число  $n$  и целые числа  $a_1, \dots, a_n$ , среди которых могут быть как положительные, так и отрицательные числа. Создайте список  $b$ , в начале которого поместить отрицательные числа, а в конце – положительные.
5. Даны натуральное число  $n$  и символьный список  $a[n]$ , в котором хранятся латинские буквы и арабские цифры. Создайте символьный список  $b$ , поместив в него только латинские буквы из списка  $a$ .
6. Даны натуральное число  $n$  и символьный список  $a[n]$ , в котором хранятся латинские буквы и арабские цифры. Создайте целочисленный список  $b$ , поместив в него только цифры из списка  $a$ .
7. Даны натуральное число  $n$  и целочисленный список  $a[n]$ . Упорядочьте список  $a$  по возрастанию.
8. Даны натуральное число  $n$  и целочисленный список  $a[n]$ . Упорядочьте список  $a$  по убыванию.
9. Даны натуральное число  $n$  и символьный список  $a[n]$ , в котором хранятся латинские буквы и арабские цифры. Создайте символьный список  $b$ , поместив в начале которого числа, а в конце – латинские буквы из списка  $a$ .
10. Даны натуральное число  $n$  и целочисленный список  $a[n]$ , состоящий только из 0 и 1. Определите самое большое количество подряд идущих единиц и выведите на экран индексы начала и конца этого диапазона.
11. Даны натуральное число  $n$  и целочисленный список  $a[n]$ , состоящий только из 0 и 1. Определите самое большое количество подряд идущих нулей и выведите на экран индекс начала и количество нулей самой длинной серии нулей.
12. Даны натуральное число  $n$  и целочисленный список  $a[n]$ . Выведите значение *true* для случая, если список  $a[n]$  представляет знакочередующуюся последовательность, и *false* – в противном случае. Последовательность является знакочередующейся, если знак очередного элемента противоположен знакам его соседей.

13. Даны натуральное число  $n$  и целочисленные списоки  $a[n]$  и  $b[n]$ , упорядоченные по возрастанию (предыдущий элемент меньше последующего). Требуется получить третий упорядоченный по возрастанию список путем слияния первых двух.
14. Даны натуральное число  $n$  и целочисленный список  $a[n]$ . Выведите значение *true* для случая, если список  $a[n]$  представляет возрастающую или убывающую последовательность, и *false* – в противном случае.
15. Даны натуральное число  $n$  и целые числа  $a_1, \dots, a_n$ . Подсчитайте максимальную длину последовательности подряд расположенных элементов, имеющих отрицательные значения. Возможно, что таких чисел в списке  $a$  нет.
16. Даны натуральное число  $n$  и целые числа  $a_1, \dots, a_n$ . Подсчитайте максимальную длину последовательности подряд расположенных элементов, имеющих положительные значения. Возможно, что таких чисел в списке  $a$  нет.
17. Даны натуральное число  $n$  и целочисленные списоки  $a[n]$  и  $b[n]$ . Из списка  $a$  выберите элементы, которых нет в списке  $b$ . Возможно, что таких чисел в списке  $a$  и в списке  $b$  нет.
18. Даны натуральное число  $n$  и целочисленные списоки  $a[n]$  и  $b[n]$ . Из списка  $a$  выбрать элементы, которые есть в списке  $b$ . Возможно, что таких чисел в списке  $a$  и в списке  $b$  нет.
19. Даны натуральное число  $n$  и целочисленные списоки  $a[n]$  и  $b[n]$ . Из списка  $a$  выбрать элементы, которые встречаются в списке  $b$  более одного раза. Возможно, что таких чисел в списке  $a$  и в списке  $b$  нет.
20. Даны натуральное число  $n$  и целочисленные списоки  $a[n]$  и  $b[n]$ . Из списка  $a$  выбрать элементы, которые больше минимального элемента среди четных чисел списка  $b$ . Возможно, что таких чисел в списке  $a$  и в списке  $b$  нет.
21. Даны натуральное число  $n$  и целочисленные списоки  $a[n]$  и  $b[n]$ . Из списка  $a$  выбрать элементы, которые меньше максимального элемента среди нечетных чисел списка  $b$ . Возможно, что таких чисел в списке  $a$  и в списке  $b$  нет.
22. Даны натуральное число  $n$  и целочисленные списоки  $a[n]$  и  $b[n]$ . Из списка  $a$  выберите элементы, которые меньше максимального элемента среди чисел списка  $b$ , значения которых кратны 3. Возможно, что таких чисел в списке  $a$  и в списке  $b$  нет.
23. Даны натуральное число  $n$  и целочисленные списоки  $a[n]$  и  $b[n]$ . Из списка  $a$  выбрать элементы, которые больше минимального элемента

среди чисел списка  $b$ , значения которых не кратны 7. Возможно, что таких чисел в списке  $a$  и в списке  $b$  нет.

24. Даны натуральное число  $n$  и целочисленные списоки  $a[n]$  и  $b[n]$ . Из списка  $a$  выберите элементы, которые больше суммы минимального и максимального элементов среди чисел списка  $b$ , значения которых кратны 5. Возможно, что таких чисел в списке  $a$  и в списке  $b$  нет.

25. Даны натуральное число  $n$  и целочисленные списоки  $a[n]$  и  $b[n]$ . Из списка  $a$  выберите элементы, которые меньше произведения минимального и максимального элементов среди нечетных чисел списка  $b$ . Возможно, что таких чисел в списке  $a$  и в списке  $b$  нет.

26. Даны натуральное число  $n$  и целочисленные списоки  $a[n]$  и  $b[n]$ . Из списка  $a$  выберите элементы, которые меньше среднего арифметического минимального и максимального элементов среди четных чисел списка  $b$ . Возможно, что таких чисел в списке  $a$  и в списке  $b$  нет.

27. Даны натуральное число  $n$  и целочисленный список  $a[n]$ , элементами которого являются двузначные числа. Из списка  $a$  выберите элементы, у которых старшая цифра ровно в 2 раза меньше младшей цифры. Возможно, что таких чисел в списке  $a$  нет.

28. Даны натуральное число  $n$  и целочисленный список  $a[n]$ , элементами которого являются двузначные числа. Из списка  $a$  выберите элементы, значения которых не принадлежат диапазону  $[50, 60]$ . Возможно, что таких чисел в списке  $a$  нет.

29. Даны натуральное число  $n$  и символьный список  $a[n]$ , в котором хранятся буквы русского алфавита. Создайте символьный список  $b$ , поместив в него буквы из списка  $a$ , лежащие в диапазоне  $[\text{б}, \text{м}]$ . Возможно, что таких букв в списке  $a$  нет.

30. Даны натуральное число  $n$  и символьный список  $a[n]$ , в котором хранятся буквы латинского алфавита. Создайте символьный список  $b$ , поместив в него буквы из списка  $a$ , не лежащие в диапазоне  $[\text{d}, \text{p}]$ . Возможно, что таких букв в списке  $a$  нет.

### 3.2 Задачи на восходящую рекурсию

#### Задачи первого уровня сложности.

1. Дана вещественная матрица размером 4 строки, 5 столбцов. Переставляя ее строки и столбцы, добейтесь того, чтобы наибольший элемент (один из них) оказался в верхнем левом углу.

2. Определите, является ли заданная целочисленная квадратная матрица порядка 5 симметричной относительно главной диагонали.
3. Дана вещественная матрица размером 4 строки, 5 столбцов. Поменяйте местами максимальный и минимальный элементы матрицы.
4. Дана целочисленная квадратная матрица порядка 5. Найдите максимальный элемент среди элементов, лежащих ниже главной диагонали, и максимальный элемент среди элементов, лежащих выше главной диагонали, поменяйте их местами.
5. Дана целочисленная квадратная матрица порядка 5. Найдите максимальный элемент среди элементов, лежащих левее вспомогательной диагонали, и максимальный элемент среди элементов, лежащих правее вспомогательной диагонали, поменяйте их местами.
6. Среди строк целочисленной квадратной матрицы порядка 5 найдите строку с минимальной суммой элементов.
7. Дана целочисленная квадратная матрица порядка 5. Найдите минимальный элемент среди элементов, лежащих ниже главной диагонали, и максимальный элемент среди элементов, лежащих выше главной диагонали, вычислите их среднее арифметическое.
8. Дана целочисленная квадратная матрица порядка 5. Найдите минимальный элемент среди элементов, лежащих левее вспомогательной диагонали, и максимальный элемент среди элементов, лежащих правее вспомогательной диагонали, и вычислите их среднее геометрическое.
9. Среди столбцов целочисленной квадратной матрицы порядка 5 найдите столбец с максимальной суммой элементов.
10. Среди тех столбцов целочисленной матрицы размером 3 строки, 5 столбцов, которые содержат только такие элементы, значения которых по модулю не превышают 10, найдите столбец с минимальным произведением элементов.
11. Даны целые числа  $a_1, \dots, a_{10}$ , целочисленная квадратная матрица порядка 5. Замените нулями в матрице те элементы, для которых имеются равные числа среди  $a_1, \dots, a_{10}$ .
12. В двумерном целочисленном массиве размером 4 строки, 5 столбцов поменяйте местами строки, симметричные относительно середины массива (горизонтальной линии).

13. В двумерном целочисленном массиве размером 4 строки, 5 столбцов поменяйте местами столбцы, симметричные относительно середины массива (вертикальной линии).
14. Даны две вещественные квадратные матрицы порядка 4. Получите новую матрицу прибавлением к элементам каждого столбца первой матрицы минимального элемента соответствующего столбца второй матрицы.
15. В целочисленной квадратной матрице порядка 4 найдите наибольший по модулю элемент. Получите квадратную матрицу порядка 3 путем выбрасывания из исходной матрицы строки и столбца, на пересечении которых расположен элемент с найденным значением.
16. В данной вещественной квадратной матрице порядка 5 найдите наименьший элемент. Получите квадратную матрицу порядка 4 путем выбрасывания из исходной матрицы строки и столбца, на пересечении которых расположен элемент с найденным значением.
17. Дана вещественная матрица размером 4 строки, 5 столбцов. Получите новую матрицу путем вычитания из всех элементов данной матрицы наименьшего по модулю элемента.
18. Дана целочисленная квадратная матрица порядка 4. Найдите в каждой строке наибольший элемент и поменяйте его местами с элементом, расположенным на главной диагонали.
19. Дана целочисленная квадратная матрица порядка 5. Найдите в каждой строке наименьший элемент и поменяйте его местами с элементом, расположенным на вспомогательной диагонали.
20. Дана вещественная матрица размером 4 строки, 5 столбцов. Определите числа  $b_1, \dots, b_4$ , равные значениям средних арифметических элементов строк.
21. Дана вещественная матрица размером 4 строки, 5 столбцов. Определите числа  $b_1, \dots, b_5$ , равные значениям средних арифметических элементов столбцов.
22. Дана вещественная матрица размером 4 строки, 5 столбцов. Определите числа  $b_1, \dots, b_5$ , равные среднему арифметическому значению максимального и минимального элементов каждого столбца.

23. В данной вещественной квадратной матрице порядка 5 найдите сумму элементов строки, в которой расположен элемент с наименьшим значением. Предполагается, что такой элемент единственный.
24. В данной вещественной квадратной матрице порядка 5 найдите среднее геометрическое положительных элементов столбца, в котором расположен элемент с наименьшим значением. Предполагается, что такой элемент единственный.
25. Дана целочисленная квадратная матрица порядка 4. Для каждой строки найдите минимальный элемент или максимальный в зависимости от условия: если на главной диагонали находится элемент больше элемента, расположенного на вспомогательной диагонали, то найдите минимальный элемент, иначе – максимальный.
26. Дана целочисленная квадратная матрица порядка 4. Ниже главной диагонали найдите минимальный элемент среди четных элементов, а выше главной диагонали – среди нечетных элементов.
27. Дана целочисленная квадратная матрица порядка 4. Ниже главной диагонали найдите минимальный элемент среди положительных элементов, а выше главной диагонали – максимальный элемент среди отрицательных элементов.
28. Дана целочисленная квадратная матрица порядка 5. Левее вспомогательной диагонали найдите минимальный элемент среди четных элементов, а правее вспомогательной диагонали – среди нечетных элементов.
29. Дана целочисленная квадратная матрица порядка 5. Левее вспомогательной диагонали найдите минимальный элемент среди положительных элементов, а правее вспомогательной диагонали – максимальный элемент среди отрицательных элементов.
30. Дана целочисленная квадратная матрица порядка 5. Для каждого столбца найдите минимальный элемент или максимальный в зависимости от условия: если на главной диагонали находится элемент меньше элемента, расположенного на вспомогательной диагонали, то найдите минимальный элемент, иначе – максимальный.

Задачи второго уровня сложности.

1. Дана вещественная матрица размером  $n$  строк,  $m$  столбцов. Упорядочьте ее строки по возрастанию их первых элементов.
2. Дана вещественная матрица размером  $n$  строк,  $m$  столбцов. Упорядочьте ее строки по возрастанию значений элементов.
3. Дана вещественная матрица размером  $n$  строк,  $m$  столбцов. Для каждой строки найдите наибольший элемент и упорядочьте строки матрицы по возрастанию значений наибольших элементов.
4. Дана вещественная матрица размером  $n$  строк,  $m$  столбцов. Для каждой строки найдите наименьший элемент и упорядочьте строки матрицы по возрастанию значений наименьших элементов.
5. Дана вещественная матрица размером  $n$  строк,  $m$  столбцов. Для каждой строки найдите сумму элементов и упорядочьте строки матрицы по возрастанию сумм.
6. Дана вещественная матрица размером  $n$  строк,  $m$  столбцов. Для каждого столбца найдите сумму элементов и упорядочьте столбцы матрицы по возрастанию сумм.
7. Дана целочисленная матрица размером  $n$  строк,  $m$  столбцов. Для каждого столбца найдите сумму нечетных элементов и упорядочьте столбцы матрицы по возрастанию сумм.
8. Дана целочисленная матрица размером  $n$  строк,  $m$  столбцов. Для каждого столбца найдите сумму положительных элементов и упорядочьте столбцы матрицы по возрастанию сумм.
9. Дана целочисленная матрица размером  $n$  строк,  $m$  столбцов. Для каждого столбца найдите произведение элементов, значения которых лежат в диапазоне  $[1, 10]$ , упорядочьте столбцы матрицы по убыванию произведений.
10. Дана целочисленная матрица размером  $n$  строк,  $m$  столбцов. Для каждого столбца найдите наибольшие элементы и упорядочьте столбцы матрицы по убыванию наибольших элементов.
11. Дана вещественная матрица размером  $n$  строк,  $m$  столбцов. Упорядочьте ее столбцы по возрастанию их первых элементов.
12. Дана целочисленная матрица размером  $n$  строк,  $m$  столбцов. Найдите и выведите строку, в которой длина максимальной серии одинаковых элементов максимальна.

13. Дана целочисленная матрица размером  $n$  строк,  $m$  столбцов. Найдите и выведите столбец, в котором длина максимальной серии возрастающих по значению элементов максимальна.
14. Дана вещественная матрица размером  $n$  строк,  $m$  столбцов. Для каждого столбца найдите среднее арифметическое элементов и упорядочьте столбцы матрицы по возрастанию среднего арифметического элементов.
15. Дана вещественная матрица размером  $n$  строк,  $m$  столбцов. Для каждой строки найдите среднее арифметическое элементов и упорядочьте строки матрицы по возрастанию среднего арифметического элементов.
16. Дана вещественная матрица размером  $n$  строк,  $m$  столбцов. Для каждого столбца найдите среднее геометрическое положительных элементов и упорядочьте столбцы матрицы по возрастанию среднего геометрического элементов.
17. Дана вещественная матрица размером  $n$  строк,  $m$  столбцов. Для каждой строки найдите среднее геометрическое положительных элементов и упорядочьте строки матрицы по убыванию среднего геометрического элементов.
18. Дана вещественная матрица размером  $n$  строк,  $m$  столбцов. Найдите строку, содержащую максимальный элемент, и упорядочьте эту строки матрицы по возрастанию значений элементов.
19. Дана вещественная матрица размером  $n$  строк,  $m$  столбцов. Найдите строку, содержащую минимальный элемент, и упорядочьте эту строки матрицы по убыванию значений элементов.
20. Дана вещественная матрица размером  $n$  строк,  $m$  столбцов. Найдите столбец, содержащий минимальный элемент, и упорядочьте этот столбец матрицы по убыванию значений элементов.
21. Дана вещественная матрица размером  $n$  строк,  $m$  столбцов. Найдите столбец, содержащий максимальный элемент, и упорядочьте этот столбец матрицы по возрастанию значений элементов.
22. Дана целочисленная матрица размером  $n$  строк,  $m$  столбцов. Найдите столбец, сумма элементов которого минимальна, и упорядочьте этот столбец матрицы по убыванию значений элементов.

23. Дана целочисленная матрица размером  $n$  строк,  $m$  столбцов. Найдите строку, сумма элементов которой максимальна, и упорядочьте эту строку матрицы по убыванию значений элементов.
24. Дана целочисленная матрица размером  $n$  строк,  $m$  столбцов. Найдите строку, среднее арифметическое элементов которой максимально, и упорядочьте эту строку матрицы по возрастанию значений элементов.
25. Дана целочисленная матрица размером  $n$  строк,  $m$  столбцов. Найдите столбец, среднее арифметическое элементов которого максимально, и упорядочьте этот столбец матрицы по возрастанию значений элементов.
26. Дана целочисленная матрица размером  $n$  строк,  $m$  столбцов. Найдите строку, среднее геометрическое элементов которой максимально, и упорядочьте эту строку матрицы по убыванию значений элементов.
27. Дана целочисленная матрица размером  $n$  строк,  $m$  столбцов. Найдите столбец, среднее геометрическое элементов которого минимально, и упорядочьте этот столбец матрицы по убыванию значений элементов.
28. Дана целочисленная матрица размером  $n$  строк,  $m$  столбцов. Найдите строку, произведение элементов которой максимально, и упорядочьте эту строку матрицы по убыванию значений элементов.
29. Дана целочисленная матрица размером  $n$  строк,  $m$  столбцов. Найдите столбец, произведение элементов которого минимально, и упорядочьте этот столбец матрицы по убыванию значений элементов.
30. Дана целочисленная матрица размером  $n$  строк,  $m$  столбцов. Найдите строку, количество различных элементов в которой максимально, и упорядочьте эту строку матрицы по возрастанию значений элементов.

#### **4. СОДЕРЖАНИЕ ОТЧЕТА**

Оформление решения задач по индивидуальным заданиям, так как показано в приложении.

## 5. КОНТРОЛЬНЫЕ ВОПРОСЫ К ЗАЩИТЕ ЛАБОРАТОРНОЙ РАБОТЫ

5.1. Какая из рекурсий работает быстрее? Нисходящая или восходящая?

5.2. Составить определение функции при помощи восходящей рекурсии для следующей задачи.

Исходные данные: список  $Y$  и атом  $A$ .

Результат: список, у которого начало начала есть  $Y$ , а последний элемент есть атом  $A$ .

5.3. Чему равен результат функции в терминальном случае

## Приложение

### Вариант оформления индивидуального задания

#### 1.1 Реализация задачи первого уровня сложности

##### 1.1.1 Спецификация

Функция возвращает матрицу, в которой строки упорядочены по возрастанию суммы их элементов.

##### 1.1.2 Структура входных данных

Входными данными является целочисленная матрица (двумерный список)  $lst$ .

##### 1.1.3 Структура выходных данных

Выходными данными является целочисленная матрица.

##### 1.1.4 Декларативное определение результата

Результатом применения функции является преобразованная матрица, в которой строки упорядочены по возрастанию суммы элементов.

##### 1.1.5 Функциональные тесты

В таблице 1 представлены тестовые случаи для функции.

Таблица 1 - Тестовые случаи для функции

Проверяемая ситуация	Вызов функции со входными значениями	Предполагаемый результат
Пустая матрица	(lab5.1 nil)	Nil
Целочисленная матрица	(lab5.1 '((5 -4 2) (2 1 -4) (-9 0 0)))	((-9 0 0) (2 1 -4) (5 -4 2))

## 1.1.6 Графическое решение

На рисунке 1 представлено графическое решение.

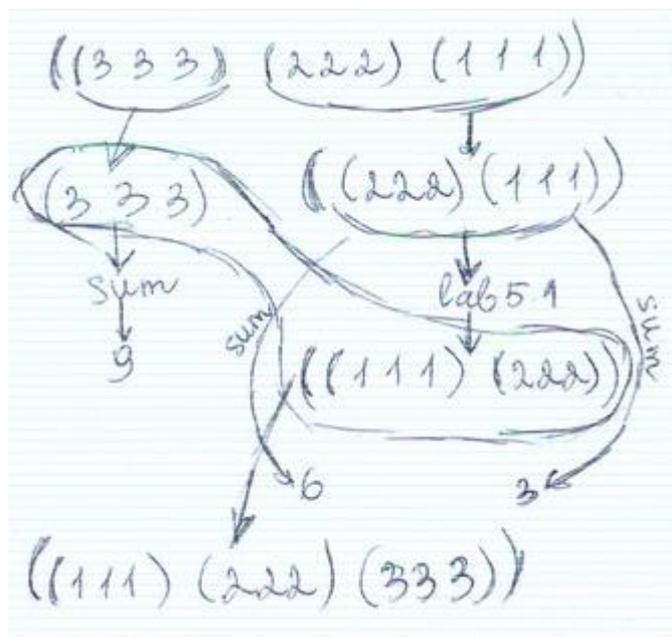


Рисунок 1 – Графическое решение для функции

### 1.1.7 Текст программы на языке Lisp

На рисунке 2 приведен текст программы на языке Lisp.

```

/* Функция возвращает сумму элементов строки */
(defun sum (lst)
  (cond
    ((null lst) 0)
    (T (+ (car lst) (sum (cdr lst))))))
/* Функция сортирует строки по их сумме */
(defun insert-row (matrix row)
  (cond
    ((null row) matrix)
    ((null matrix) (cons row nil))
    ((< (sum row) (sum (car matrix))) (cons row matrix))
    (T (cons (car matrix) (insert-row (cdr matrix) row)))))
/* Функция возвращает матрицу, в которой строки упорядочены
по возрастанию суммы их элементов. */
(defun lab5.1 (matrix)
  (cond
    ((null matrix) nil)
    (T (insert-row (lab5.1 (cdr matrix)) (car matrix)))))

```

Рисунок 2 – Текст программы

### 1.1.8 Результаты теста

На рисунке 3 представлен результат работы тестового драйвера для данной функции.

```
█_$(test5.1)
"test 5.1 OK"
_$(
```

Рисунок 3 – Результаты тестирования функции

## 1.2 Реализация задачи второго уровня сложности

### 1.2.1 Спецификация

Функция совершает подсчет максимальной длины последовательности подряд расположенных элементов, имеющих положительные значения. Возможно, что таких чисел в массиве нет.

### 1.2.2 Структура входных данных

Входными данными является список.

### 1.2.3 Структура выходных данных

Выходными данными является переменная целого типа.

### 1.2.4 Декларативное определение результата

Функция считает количество подряд идущих положительных чисел и список до первого отрицательного числа и сравнивает полученное количество с количеством после вызова этой функции от нового списка.

### 1.2.5 Функциональные тесты

В таблице 2 представлены тестовые случаи для функции.

Таблица 2 - Тестовые случаи для функции

Проверяемая ситуация	Вызов функции со входными значениями	Предполагаемый результат
Последовательность с подряд идущими положительными числами	(lab5.2 -3 -6 5 9 -4 -3)	2
Последовательность положительных чисел	(lab5.2 5 7 4 3 9 1 2)	7
Последовательность отрицательных чисел	(lab5.2 -1 -9 -3 -4 -6)	0
Знакопеременная последовательность	(lab5.2 -1 6 -9 6 -2 4)	1
Пустая последовательность	(lab5.2 nil)	0

### 1.2.6 Графическое решение

На рисунке 4 представлено графическое решение.

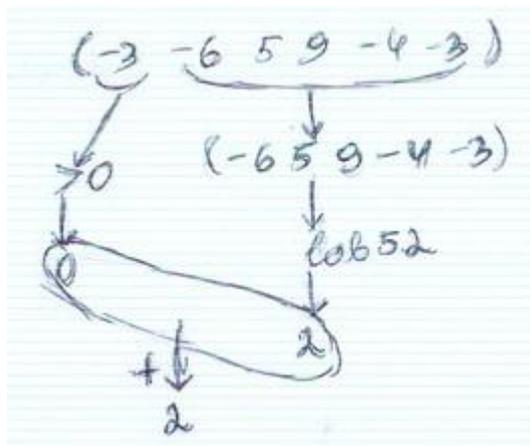


Рисунок 4 – Графическое решение для функции

### 1.2.7 Текст программы на языке Lisp

На рисунке 5 приведен текст программы на языке Lisp.

```

/* Функция совершает подсчет максимальной длины
последовательности подряд расположенных элементов,
имеющих положительные значения. Возможно, что таких чисел
в массиве нет. */
(defun findChainList (lst)
  (cond
    ((null lst) (cons 0 (cons lst '())))
    ((>= (car lst) 0) (cons (+ 1 (car (findChainList (cdr lst)))) (cons
(cadr (findChainList (cdr lst))) '())))
    (t (cons 0 (cons lst '())))))
/* Функция формирует результирующий список. */
(defun lab5.2(lst)
  (cond
    ((null lst) 0)
    ((< (car lst) 0) (lab5.2 (cdr lst)))
    ((> (car (findChainList lst)) (lab5.2 (cadr (findChainList lst))))
(car (findChainList lst)))
    (t (lab5.2 (cadr (findChainList lst))))))

```

Рисунок 5 – Текст программы

### 1.2.8 Результаты теста

На рисунке 6 представлен результат работы тестового драйвера для данной функции.

```
└─$ (test5.2) ` `
    "test 5.2 OK"
└─$ |
```

Рисунок 6 – Результаты тестирования функции

### 1.3 Реализация задачи первого уровня сложности восходящей рекурсией

#### 1.3.1 Спецификация

Функция возвращает матрицу, в которой строки упорядочены по возрастанию суммы их элементов.

#### 1.3.2 Структура входных данных

Входными данными является целочисленная матрица (двумерный список).

#### 1.3.3 Структура выходных данных

Выходными данными является целочисленная матрица.

#### 1.3.4 Декларативное определение результата

Результатом применения функции является преобразованная матрица, в которой строки упорядочены по возрастанию суммы элементов.

#### 1.3.5 Функциональные тесты

В таблице 3 представлены тестовые случаи для функции.

Таблица 3 - Тестовые случаи для функции

Проверяемая ситуация	Вызов функции со входными значениями	Предполагаемый результат
Пустая матрица	(lab5.3 nil)	nil
Целочисленная матрица	(lab5.3 '((5 -4 2) (2 1 -4) (-9 0 0)))	((-9 0 0) (2 1 -4) (5 -4 2))

### 1.3.6 Графическое решение

На рисунке 7 представлено графическое решение.

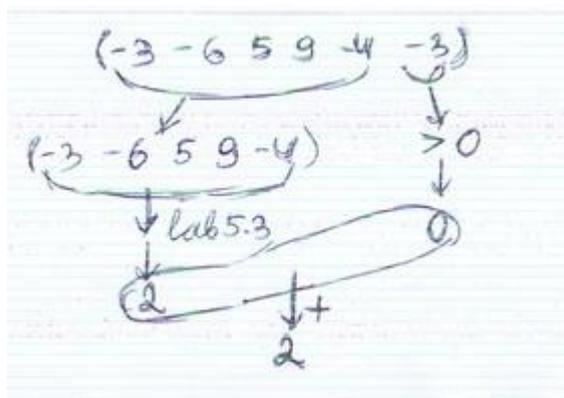


Рисунок 7 – Графическое решение для функции

## 1.3.7 Текст программы на языке Lisp

На рисунке 8 приведен текст программы на языке Lisp.

```

/* Функция вычисляет сумму элементов строки восходящей
рекурсией */
(defun w-sum (lst akk)
  (cond ((null lst) akk) (T (w-sum (cdr lst) (+ (car lst) akk)))))
/* Функция вызывает функцию w-sum (lst akk) */
(defun sum (lst) (w-sum lst 0))
/* Функция вызывает функцию w-reverse (lst akk) */
(defun my-reverse (lst) (w-reverse lst nil))
/* Функция, которая реверсирует входной список восходящей
рекурсией */
(defun w-reverse (lst akk)
  (cond ((null lst) akk) (T (w-reverse (cdr lst) (cons (car lst) akk)))))
/* Функция вставляет строчку в необходимое место */
(defun w-insert-row (matrix row akk)
  (cond ((null matrix) (my-reverse (cons row akk))) ((< (sum row)
(sum (car matrix))) (append (my-reverse akk) (cons row matrix)))
(T (w-insert-row (cdr matrix) row (cons (car matrix) akk)))))
/* Функция вызывает функцию w-insert-row (matrix row akk) */
(defun insert-row (matrix row) (w-insert-row matrix row nil))
/* Функция сортирует строки матрицы */
(defun w-sort-rows (matrix akk) (cond ((null matrix) akk) (T (w-sort-
rows (cdr matrix) (insert-row akk (car matrix)))))
/* Функция возвращает матрицу, в которой строки упорядочены
по возрастанию суммы их элементов. */
(defun lab5.3 (matrix) (w-sort-rows matrix nil))

```

Рисунок 8 – Текст программы

### 1.3.8 Результаты теста

На рисунке 9 представлен результат работы тестового драйвера для данной функции.

```

_ $ (test5.3)
"test 5.3 OK"
_ $ |

```

Рисунок 9 – Результаты тестирования функции

## 2 Лабораторная работа «Программирование с использованием двумерных массивов» на языке Lisp

### 2.1 Реализация задачи первого уровня сложности

#### 2.1.1 Спецификация

Функция возвращает список, в котором минимальный элемент заменен на целую часть среднего арифметического всех элементов списка.

#### 2.1.2 Структура входных данных

Входными данными является список целых чисел `lst`.

#### 2.1.3 Структура выходных данных

Выходными данными является список целых чисел.

#### 2.1.4 Декларативное определение результата

Результатом применения функции является преобразованный список, в котором минимальный элемент заменен на целую часть среднего арифметического всех элементов.

### 2.1.5 Функциональные тесты

В таблице 4 представлены тестовые случаи для функции.

Таблица 4 - Тестовые случаи для функции

Проверяемая ситуация	Вызов функции со входными значениями	Предполагаемый результат
Пустой список	(lab6.1 nil)	Nil
Список целых чисел	(lab6.1 '((5 1 4 -6 5 2 8)))	(5 1 4 2 5 2 8)

### 2.1.6 Графическое решение

На рисунке 10 представлено графическое решение.

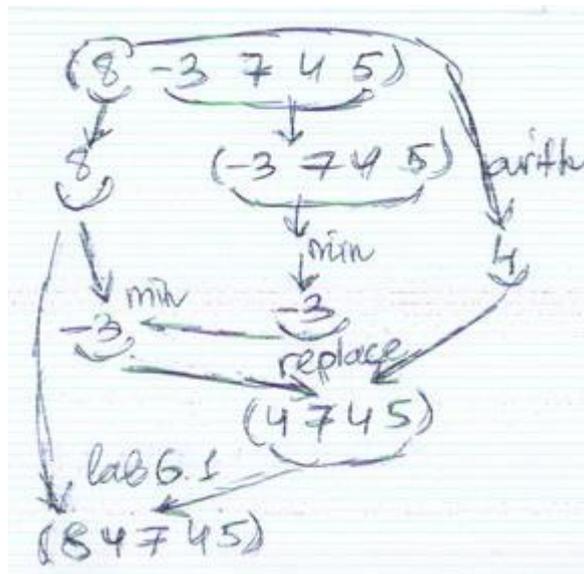


Рисунок 10 – Графическое решение для функции

### 2.1.7 Текст программы на языке Lisp

На рисунке 11 представлен текст программы на языке Lisp.

```

/* Функция возвращает сумму элементов списка */
(defun sum (lst)
  (cond ((null lst) 0) (T (+ (car lst) (sum (cdr lst))))))

/* Функция возвращает длину списка */
(defun my-length (lst)
  (cond ((null lst) 0) (T (+ 1 (length (cdr lst))))))

/* Функция возвращает целую часть среднего арифметического
всех элементов списка */
(defun arithmetic-mean-int (lst)
  (cond ((null lst) 0) (T (/ (sum lst) (my-length lst)))))

(defun my-min (lst)
  (cond
    ((null (cdr lst)) (car lst))
    ((> (car lst) (my-min (cdr lst))) (my-min (cdr lst)))
    (T (car lst))))

/* Функция меняет минимальный элемент на целую часть
среднего арифметического всех элементов списка. */
(defun replace (lst x y)
  (cond
    ((or (null lst) (not (numberp x)) (not (numberp y))) nil)
    ((= (car lst) x) (cons y (replace (cdr lst) x y)))
    (T (cons (car lst) (replace (cdr lst) x y)))))

/* Функция возвращает список, в котором минимальный элемент
заменен на целую часть среднего арифметического всех
элементов списка. */
(defun lab6.1 (lst)
  (cond ((null lst) nil) (T (replace lst (my-min lst) (arithmetic-mean-int

```

Рисунок 11 – Текст программы

### 2.1.8 Результаты теста

На рисунке 12 представлен результат работы тестового драйвера для данной функции.

```
└─$ (test6.1)
    "test 6.1 OK"
└─$ |
```

Рисунок 12 – Результаты тестирования функции

## 2.2 Реализация задачи второго уровня сложности

### 2.2.1 Спецификация

Дана вещественная матрица размером  $n$  строк,  $m$  столбцов. Функция для каждого столбца находит среднее геометрическое положительных элементов и упорядочит столбцы матрицы по возрастанию среднего геометрического элементов.

### 2.2.2 Структура входных данных

Входными данными является список списков `lst`.

### 2.2.3 Структура выходных данных

Выходными данными является список списков.

### 2.2.4 Декларативное определение результата

Результатом выполнения данной функции является транспонированная отсортированная по средним геометрическим значениям строк изначально транспонированной матрицы.

### 2.2.5 Функциональные тесты

В таблице 5 представлены тестовые случаи для функции.

Таблица 5 - Тестовые случаи для функции

Проверяемая ситуация	Вызов функции со входными значениями	Предполагаемый результат
Имеются перестановки	(lab6.2 '((4 1 0) (4 1 1) (6 3 0)))	((0 1 4) (1 1 4) (0 3 6))
Перестановок нет	(lab6.2 '((5 6 7) (8 9 10) (11 12 13)))	'((5 6 7) (8 9 10) (11 12 13))

### 2.2.6 Графическое решение

На рисунке 13 представлено графическое решение.

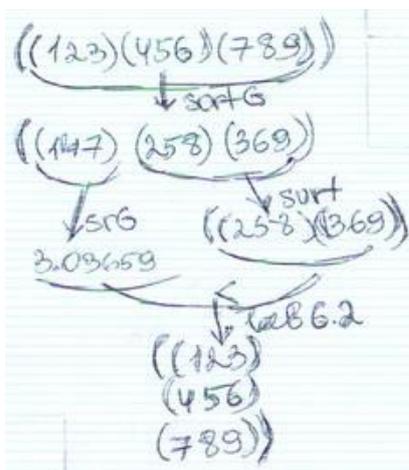


Рисунок 13 – Графическое решение для функции

## 2.2.7 Текст программы на языке Lisp

На рисунке 14 представлен текст программы на языке Lisp.

```

/* Функция возвращает корень из числа */
(defun sqrtn(a n) (expt a (/ 1.0 n)))

/* Функция для каждого столбца находит среднее
геометрическое положительных элементов и упорядочит
столбцы матрицы по возрастанию среднего геометрического
элементов */
(defun lab6.2(lst)
  (cond ((null lst) '(1 0)) ((>= (car lst) 0) (cons (* (car (times (cdr
lst))) (car lst)) (cons (+ (cadr (lab6.2 (cdr lst))) 1) '()))) (t (lab6.2 (cdr
lst)))))

/* Функция вставляет элемент в нужную позицию */
(defun insert (el a)
  (cond ((null a) (cons el '())) ((< (srG el) (srG (car a))) (cons el a))
(t (cons (car a) (insert el (cdr a)))))

/* Функция сортирует столбцы матрицы */
(defun sort(a)
  (cond ((null a) '()) (t (insert (car a) (sort (cdr a)))))

/* Функция для каждого столбца находит среднее
геометрическое положительных элементов */
(defun srG(lst) (sqrtn (car (sort lst)) (cadr (sort lst))))

/* Функция для каждого столбца упорядочит столбцы матрицы
по возрастанию среднего геометрического элементов */
(defun sortG(lst) (transp (sort (transp lst))))

```

Рисунок 14 – Текст программы

### 2.2.8 Результаты теста

На рисунке 15 представлен результат работы тестового драйвера для данной функции.

```
_$ (test6.2) ' "  
"test 6.2 OK"  
_$ |
```

Рисунок 15 – Результаты тестирования функции

## 2.3 Реализация задачи первого уровня сложности восходящей рекурсией

### 2.3.1 Спецификация

Функция возвращает матрицу, в которой строки упорядочены по возрастанию суммы их элементов.

### 2.3.2 Структура входных данных

Входными данными является целочисленная матрица (двумерный список) `lst`.

### 2.3.3 Структура выходных данных

Выходными данными является целочисленная матрица.

### 2.3.4 Декларативное определение результата

Результатом применения функции является преобразованный список, в котором минимальный элемент заменен на целую часть среднего арифметического всех элементов.

### 2.3.5 Функциональные тесты

В таблице 6 представлены тестовые случаи для функции.

Таблица 6 - Тестовые случаи для функции

Проверяемая ситуация	Вызов функции со входными значениями	Предполагаемый результат
Пустая матрица	(lab6.3 nil)	nil
Целочисленная матрица	(lab6.3 '(5 1 4 -6 5 2 8))	(5 1 4 2 5 2 8)

### 2.3.6 Графическое решение

На рисунке 16 представлено графическое решение.

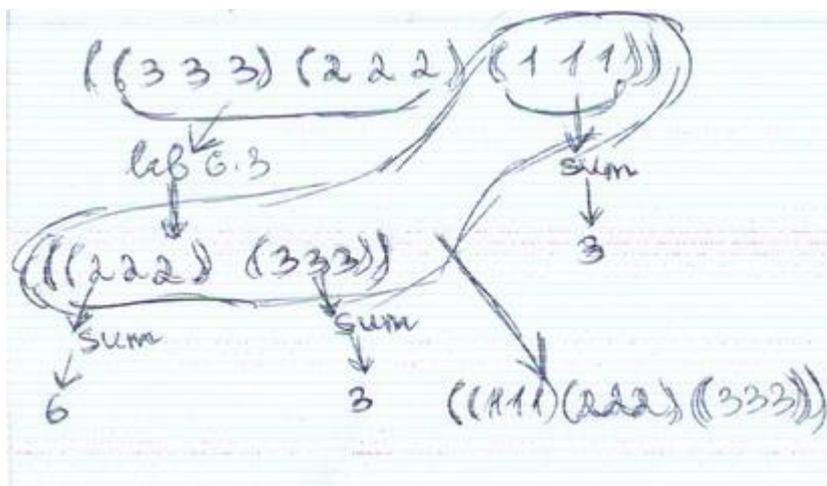


Рисунок 16 – Графическое решение для функции

## 2.3.7 Текст программы на языке Lisp

На рисунках 17-18 представлен текст программы на языке Lisp.

```

/* Функция вызывает функцию wsum (lst akk) */
(defun sum (lst) (wsum lst 0))

/* Функция возвращает сумму элементов списка */
(defun wsum (lst akk)
  (cond ((null lst) akk) (T (wsum (cdr lst) (+ (car lst) akk)))))

/* Функция вызывает функцию wlength (lst akk) */
(defun my-length (lst) (wlength lst 0))

/* Функция возвращает длину списка */
(defun wlength (lst akk)
  (cond ((null lst) akk) (T (wlength (cdr lst) (+ akk 1)))))

/* Функция возвращает среднее арифметическое элементов */
(defun arithmetic-mean-int (lst)
  (cond ((null lst) 0) (T (/ (sum lst) (my-length lst)))))

/* Функция вызывает функцию wmin (lst min-el) */
(defun my-min (lst) (wmin (cdr lst) (car lst)))

/* Функция возвращает минимальный элемент в списке */
(defun wmin (lst min-el)
  (cond ((null lst) min-el) ((< (car lst) min-el) (wmin (cdr lst) (car lst)))
    (T (wmin (cdr lst) min-el))))

```

Рисунок 17 – Текст программы

```

/* Функция возвращает реверсированный список */
(defun wreverse (lst akk)
  (cond ((null lst) akk) (T (wreverse (cdr lst) (cons (car lst) akk)))))
/* Функция вызывает функцию wreplace (lst x y akk) */
(defun replace (lst x y) (wreplace lst x y nil))
/* Функция меняет местами необходимые элементы */
(defun wreplace (lst x y akk)
  (cond ((null lst) (my-reverse akk))
        ((= (car lst) x) (append (my-reverse (cons y akk)) (cdr lst)))
        (T (wreplace (cdr lst) x y (cons (car lst) akk))) ))
/* Функция возвращает матрицу, в которой строки упорядочены
по возрастанию суммы их элементов */
(defun lab6.3 (lst)
  (cond ((null lst) nil) (T (replace lst (my-min lst) (arithmetic-mean-int
lst)))))

```

Рисунок 18 – Текст программы

### 2.3.8 Результаты теста

На рисунке 19 представлен результат работы тестового драйвера для данной функции.

```

_ $ (test6.3)
"test 6.3 OK"
_ $ |

```

Рисунок 19 – Результаты тестирования функции

### 3 Тестовый драйвер

#### 3.1 Текст программы тестового драйвера

На рисунке 20 представлен текст программы тестового драйвера

```
/* Тестовый драйвер */  
(defun testdriver(func f)  
  (cond  
    ((= (car (car f)) (apply (read func) (print "ОК")))  
     (t (print "Ошибка")))))
```

Рисунок 20 – Текст программы тестового драйвера