

## Лабораторная работа № 4. Работа со списками

Задание 1. Реализовать предикат чтения списка с клавиатуры и предикат вывода списка на экран. При дальнейшей разработке пользоваться указанными предикатами.

Задание 2. Реализовать предикат `sum_list_down(+List, ?Summ)`, который проверяет, является ли `Summ` суммой элементов списка или записывает в эту переменную сумму элементов. Для построения воспользоваться рекурсией вниз. Реализовать программу, читающую список, считающую сумму элементов и выводющую сумму на экран, с использованием предикатов из задания 1 и построенного предиката `sum_list_down /2`.

Задание 3. Построить предикат `sum_list_up(+List, ?Summ)`, который проверяет, является ли `Summ` суммой элементов списка или записывает в эту переменную сумму элементов. Для построения воспользоваться рекурсией вверх.

Задание 4. Построить предикат, `list_el_numb(+List, ?Elem, ?Number)`. Если задано значение `Elem`, то предикат записывает в `Number` номер первого вхождения `Elem`. Если задано значение `Number`, то предикат записывает в `Elem` значение, находящееся в списке под номером `Number`. Если заданы оба значения, то предикат проверяет, находится ли элемент `Elem` в списке под номером `Numb`.

Задание 4. Реализовать программу, которая читает список, читает элемент и находит номер первого вхождения элемента в список. В случае, если элемент отсутствует, пользователю выводится соответствующее сообщение. Воспользоваться предикатами из заданий 1 и 3.

Задание 5. Реализовать программу, которая читает список, читает номер элемента и находит соответствующий элемент. В случае, если номер некорректный, пользователю выводится соответствующее сообщение.

Задание 6. Реализовать предикат `min_list_up(+List, ?Min)`, который записывает минимальный элемент списка `List` в переменную `Min` или

проверяет, является ли значение в переменной Min минимальным элементом в списке List. Реализацию провести рекурсией вверх.

Задание 7. Реализовать предикат `min_list_down(+List, ?Min)`, который записывает минимальный элемент списка List в переменную Min или проверяет, является ли значение в переменной Min минимальным элементом в списке List. Реализацию провести рекурсией вниз.

Задание 8. Реализовать программу, которая читает список, находит и выводит на экран его минимальный элемент. Воспользоваться предикатами из заданий 1, 6 или 7.

Задание 9. Построить предикат, который возвращает true, если элемент есть в списке.

Задание 10. Построить предикат, который переворачивает список.

Задание 11. Построить предикат `p(Sublist,List)`, который возвращает true, если элементы Sublist встречаются в List в том же порядке.

Задание 12. Построить предикат, который удаляет элемент с заданным номером из списка.

Задание 13. Построить предикат, который удаляет все элементы, равные данному.

Задание 14. Построить предикат, который проверяет, встречаются ли все элементы в списке ровно 1 раз.

Задание 15. Построить предикат, который строит новый список, составленный из уникальных элементов введенного, то есть убирает все повторы, например из списка `[1,1,2,3,3,3]` получает список `[1,2,3]`.

Задание 16. Построить предикат, который получает для данного элемента количество раз, которое он встречается в списке.

Задание 17. Построить предикат, получающий длину списка.

Задание 18. Решить предложенные задачи по вариантам. В задаче необходимо выделить отдельно предикат чтения, предикат логики работы и предикат вывода. Предикат логики работы программы должен иметь минимально возможное количество аргументов, то есть вызов

`main_predicat(List,0,Otv)` считается недопустимым, так как для выполнения задачи используется вспомогательная переменная со значением, равным 0. Такое выделение главного предиката считается недопустимым. Для каждого построенного предиката необходимо четко пояснить, для чего он используется в этой задаче, то есть какие аргументы унифицированы, а какие нет, на момент его вызова. И какие значения окажутся в переменных по завершении доказательства цели. Если какая-то из небольших подзадач была ранее реализована в одном из предикатов, **ОБЯЗАТЕЛЬНО** использование этого предиката вместо переписывания. Использование встроенных предикатов считается **НЕДОПУСТИМЫМ**.

Вариант1. Задачи 1, 2, 13, 15, 25, 28, 37, 43, 49

Вариант2. Задачи 1, 2, 14, 16, 26, 29, 38, 44, 50

Вариант3. Задачи 3, 4, 13, 15, 27, 30, 39, 45, 51

Вариант4. Задачи 3, 4, 14, 16, 25, 28, 40, 46, 52

Вариант5. Задачи 5, 6, 17, 19, 26, 29, 41, 47, 53

Вариант6. Задачи 5, 6, 18, 20, 27, 30, 42, 48, 54

Вариант7. Задачи 7, 8, 17, 19, 31, 34, 37, 43, 55

Вариант8. Задачи 7, 8, 18, 20, 32, 35, 38, 44, 56

Вариант9. Задачи 9, 10, 21, 23, 33, 36, 39, 45, 57

Вариант10. Задачи 9, 10, 22, 24, 31, 34, 40, 46, 58

Вариант11. Задачи 11, 12, 21, 23, 32, 35, 41, 47, 59

Вариант12. Задачи 11, 12, 22, 24, 33, 36, 42, 48, 60

1.1 Дан целочисленный массив. Необходимо найти количество элементов, расположенных после последнего максимального.

1.2 Дан целочисленный массив. Необходимо найти индекс минимального элемента.

.3 Дан целочисленный массив и натуральный индекс (число, меньшее размера массива). Необходимо определить является ли элемент по указанному индексу глобальным максимумом.

1.4 Дан целочисленный массив. Вывести индексы массива в том порядке, в котором соответствующие им элементы образуют убывающую последовательность.

1.5 Дан целочисленный массив и натуральный индекс (число, меньшее размера массива). Необходимо определить является ли элемент по указанному индексу глобальным минимумом.

1.6 Дан целочисленный массив. Необходимо осуществить циклический сдвиг элементов массива влево на три позиции.

1.7 Дан целочисленный массив. Необходимо осуществить циклический сдвиг элементов массива вправо на две позиции.

1.8 Дан целочисленный массив. Необходимо найти индексы двух наименьших элементов массива.

1.9 Дан целочисленный массив. Необходимо найти элементы, расположенные перед последним минимальным.

1.10 Даны два массива. Необходимо найти количество совпадающих по значению элементов.

1.11 Дан целочисленный массив, в котором лишь один элемент отличается от остальных. Необходимо найти значение этого элемента.

1.12 Дан целочисленный массив. Необходимо переставить в обратном порядке элементы массива, расположенные между его минимальным и максимальным элементами.

1.13 Дан целочисленный массив. Необходимо разместить элементы, расположенные до минимального, в конце массива.

1.14 Дан целочисленный массив и интервал  $a..b$ . Необходимо найти количество элементов в этом интервале.

1.15 Дан целочисленный массив и натуральный индекс (число, меньшее размера массива). Необходимо определить является ли элемент по указанному индексу локальным минимумом.

1.16 Дан целочисленный массив. Необходимо найти элементы, расположенные между первым и вторым максимальным.

1.17 Дан целочисленный массив. Необходимо поменять местами минимальный и максимальный элементы массива.

1.18 Дан целочисленный массив. Необходимо найти элементы, расположенные перед первым минимальным.

1.19 Дан целочисленный массив. Необходимо осуществить циклический сдвиг элементов массива вправо на одну позицию.

1.20 Дан целочисленный массив. Необходимо найти все пропущенные числа.

1.21 Дан целочисленный массив. Необходимо найти элементы, расположенные после первого максимального.

1.22 Дан целочисленный массив и интервал  $a..b$ . Необходимо найти количество минимальных элементов в этом интервале.

1.23 Дан целочисленный массив. Необходимо найти два наименьших элемента.

1.24 Дан целочисленный массив. Необходимо найти два наибольших элемента.

1.25 Дан целочисленный массив и интервал  $a..b$ . Необходимо найти максимальный из элементов в этом интервале.

1.26 Дан целочисленный массив. Необходимо найти количество элементов между первым и последним минимальным.

1.27 Дан целочисленный массив. Необходимо осуществить циклический сдвиг элементов массива влево на одну позицию.

1.28 Дан целочисленный массив. Необходимо найти элементы, расположенные между первым и последним максимальным.

1.29 Дан целочисленный массив и интервал  $a..b$ . Необходимо проверить наличие максимального элемента массива в этом интервале.

1.30 Дан целочисленный массив и натуральный индекс (число, меньшее размера массива). Необходимо определить является ли элемент по указанному индексу локальным максимумом.

1.31 Дан целочисленный массив. Найти количество чётных элементов.

1.32 Дан целочисленный массив. Найти количество его локальных максимумов.

1.33 Дан целочисленный массив. Проверить, чередуются ли в нем положительные и отрицательные числа.

1.34 Дан целочисленный массив и отрезок  $a..b$ . Необходимо найти элементы, значение которых принадлежит этому отрезку.

1.35 Дано вещественное число  $R$  и массив вещественных чисел. Найти элемент массива, который наиболее близок к данному числу.

1.36 Дан целочисленный массив. Необходимо найти максимальный нечетный элемент.

1.37 Дан целочисленный массив. Вывести индексы элементов, которые меньше своего левого соседа, и количество таких чисел.

1.38 Дан целочисленный массив и отрезок  $a..b$ . Необходимо найти количество элементов, значение которых принадлежит этому отрезку.

1.39 Дан целочисленный массив. Необходимо вывести вначале его элементы с четными индексами, а затем - с нечетными.

1.40 Дан целочисленный массив. Необходимо найти минимальный четный элемент.

1.41 Дан целочисленный массив. Найти среднее арифметическое модулей его элементов.

1.42 Дан целочисленный массив. Найти все элементы, которые меньше среднего арифметического элементов массива.

1.43 Дан целочисленный массив. Необходимо найти количество минимальных элементов.

1.44 Дан массив чисел. Необходимо проверить, чередуются ли в нем целые и вещественные числа.

1.45 Дан целочисленный массив и интервал  $a..b$ . Необходимо найти сумму элементов, значение которых попадает в этот интервал.

1.46 Дан целочисленный массив. Необходимо вывести вначале его положительные элементы, а затем - отрицательные.

1.47. Для введенного списка положительных чисел построить список всех положительных делителей элементов списка без повторений.

1.48. Для введенного списка построить список с номерами элемента, который повторяется наибольшее число раз.

1.49. Для введенного списка положительных чисел построить список всех положительных простых делителей элементов списка без повторений.

1.50. Для двух введенных списков L1 и L2 построить новый список, состоящий из элементов, встречающихся только в одном из этих списков и не повторяющихся в них.

1.51. Для введенного списка построить два списка L1 и L2, где элементы L1 это неповторяющиеся элементы исходного списка, а элемент списка L2 с номером  $i$  показывает, сколько раз элемент списка L1 с таким номером повторяется в исходном.

1.52. Для введенного числа построить список всех его простых делителей, причем если введенное число делится на простое число в степени  $p$ , то в итоговом списке число должно повторяться  $p$  раз. Результирующий список должен быть упорядочен по возрастанию.  $p \leq p \leq p$

1.53. Для введенного списка построить новый с элементами, большими, чем среднее арифметическое списка, но меньшими, чем его максимальное значение.

1.54. Для введенного списка построить список из элементов, встречающихся в исходном более трех раз.

1.55. Для введенного списка построить новый список, который получен из начального упорядочиванием по количеству встречаемости элемента,

То есть из списка [5,6,2,2,3,3,3,5,5,5] необходимо получить список [5,5,5,5,3,3,3,2,2,6].

1.56. Для введенного списка посчитать среднее арифметическое непроводных элементов, которые больше, чем среднее арифметическое простых.

1.57 Для введенного списка найти количество таких элементов, которые больше, чем сумма всех предыдущих.

1.58 Для введенного списка вывести количество элементов, которые могут быть получены как сумма двух любых других элементов списка.

1.59. Дан список. Построить новый список из квадратов неотрицательных чисел, меньших 100 и встречающихся в массиве больше 2 раз.

1.60. Дан список. Построить массив из элементов, делящихся на свой номер и встречающихся в исходном массиве 1 раз.