

7. Массивы и коллекции

7.1. Одномерные массивы

7.1-0 Написать программу, выполняющую вейвлет-преобразование Хаара для заданного массива с длиной, равной степени 2. Само преобразование выполняется следующим образом. В первую половину массива записываются полусуммы пар соседних элементов, а во вторую — полуразности. После этого преобразование повторяется для первой половины массива до тех пор, пока не останется только одна полусумма.

Например, для массива $(2, 8, 6, 0)$ после первого шага получим

$$\left(\frac{2+8}{2}, \frac{6+0}{2}, \frac{2-8}{2}, \frac{6-0}{2} \right) = (5, 3, -3, 3).$$

На втором шаге преобразование будет применено только к полусуммам — $(5, 3)$. В итоге получим массив

$$\left(\frac{5+3}{2}, \frac{5-3}{2}, -3, 3 \right) = (4, 1, -3, 3).$$

Так как осталась одна полусумма, это и есть ответ.

7.1-1 Написать программу, случайным образом перемешивающую элементы массива при помощи алгоритма Фишера — Йетса.

Идея метода следующая. На первом шаге первый элемент обменивается со случайно выбранным элементом из массива (им может оказаться и сам первый элемент). На втором шаге второй элемент обменивается со случайно выбранным элементом кроме первого. На третьем — третий элемент со случайным кроме первых двух и так далее. Операция повторяется для каждого элемента массива.

7.1-2 Написать программу, заменяющую все составные числа в массиве последовательных чисел от 2 до указанного числа нулями при помощи алгоритма, называемого решето Эратосфена. Идея алгоритма следующая: перебираются все элементы, если элемент не нулевой, то заменяются нулями все, идущие после него с шагом, равным значению этого элемента. Алгоритм продолжается до элемента со значением $\lceil \sqrt{N} \rceil$, где N — значение последнего элемента.

Например, для массива $(2, 3, 4, 5, 6, 7, 8, 9)$ последовательность преобразований имеет вид (текущий элемент подчеркнут):

$$\begin{aligned} &(\underline{2}, 3, 4, 5, 6, 7, 8, 9), \\ &(2, \underline{3}, 0, 5, 6, 7, 0, 9), \\ &(2, 3, \underline{0}, 5, 0, 7, 0, 0). \end{aligned}$$

7.1-3 Написать программу для так называемой «пузырьковой сортировки» массива по неубыванию. Идея метода следующая. Просматриваются все пары соседних элементов и, если их порядок неверный, они обмениваются местами. Проходы по массиву повторяются, пока не окажется, что массив отсортирован.

7.1-4 Написать программу для сортировки массива по неубыванию простым выбором. Идея метода следующая. В массиве ищется наименьший элемент и меняется местами с первым, затем наименьший среди оставшихся меняется местами со вторым и так далее.

7.1-5 Написать программу, переставляющую элементы массива так, чтобы сначала были записаны значения, меньшие некоторого числа p , потом равные ему, а затем остальные.

7.1-6 Написать программу, реализующую так называемое правило 30 для клеточных автоматов. Дан массив заданной длины, элементы которого равны нулю кроме среднего, который равен единицы. Программа должна выполнить указанное число шагов преобразования массива выводя на каждом шаге результат.

В ходе преобразования проверяются все ячейки массива, кроме крайних. На следующем шаге ячейка становится равной 1 тогда и только тогда, когда она со своими соседями образует одну из четырёх комбинаций: 100, 010, 001 или 011. В противном случае ячейка на новом шаге будет равна 0.

Например для массива размера 7 первые три шага имеют вид:

0001000,
0011100,
0110010.

7.1-7 Написать программу, разбивающую массив на три подмассива с сохранением порядка. При этом суммы элементов в них должны быть как можно ближе друг к другу: разница между наибольшей и наименьшей суммами должна быть минимальной. Программа должна вывести номера первых элементов второго и третьего подмассивов.

Например, массив $(1, 2, 3, 2, 2)$ разбивается на подмассивы $(1, 2)$, (3) и $(2, 2)$. Если элементы нумеруются с нуля, то искомые индексы — 2 и 3.

7.1-8 Написать программу, находящую максимальное значение, которое можно получить, складывая последовательные элементы массива. Например, для массива $(1, -2, 3, 1, -1, 2)$ максимальная сумма равна $3 + 1 + (-1) + 2 = 5$.

7.1-9 Написать программу для тернарного поиска элемента в отсортированном массиве. Алгоритм похож на бинарный поиск. Массив делится двумя точками на три части приблизительно равного размера. Затем сравнивая искомое значение со значениями в точках, определяем, в какой из частей оно нахо-

дится. Затем поиск повторяется в этой части, и так далее, пока не будет найден элемент или не окажется, что его нет в массиве.

7.2. Многомерные массивы

7.2-0 Дан двумерный массив целых чисел. Написать программу, заменяющую нулями элементы, стоящие в строках и столбцах, в которых есть нули.

7.2-1 Дан двумерный массив действительных чисел. Написать программу, заменяющую каждый элемент арифметическим средним его смежных соседей по горизонтали, вертикали и диагонали.

7.2-2 Дан двумерный массив размера $M \times N$ действительных чисел. Написать программу для его нормализации по формуле

$$a_{ij}^{\text{норм}} = \frac{a_{ij} - \bar{a}}{\sigma},$$

где \bar{a} — математическое ожидание, а σ — стандартное отклонение, вычисляемые по формулам

$$\bar{a} = \frac{1}{MN} \sum_{i,j} a_{ij},$$
$$\sigma = \sqrt{\frac{1}{MN} \sum_{i,j} (a_{ij} - \bar{a})^2}.$$

7.2-3 Дан двумерный массив действительных чисел. Написать программу, проверяющую, есть ли в нём линейно зависимые строки и столбцы.

7.2-4 Написать программу, заполняющую двумерный массив значениями из треугольника Паскаля с вершиной в верхем левом углу.

В полученном массиве первая строка и первый столбец должны содержать только единицы, а остальные элементы должны быть равны сумме элемента выше и элемента левее.

Пример заполнения для массива 3×3 :

$$\begin{array}{ccc} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{array}$$

7.2-5 Дан двумерный массив целых чисел. Написать программу, проверяющую, сколько в нём квадратов размера 2×2 , состоящих только из чётных элементов.

7.2-6 Дан квадратный массив действительных чисел. Написать программу, формирующую массив с суммами элементов на его диагоналях.

7.2-7 Даны две двумерные матрицы. Написать программу, находящую их произведение по правилам матричного умножения, если это возможно.

7.2-8 Дан двумерный массив вещественных чисел. Написать программу, заменяющую каждый его элемент арифметическим средним элементов ниже и правее него.

7.2-9 Дан двумерный массив. Написать программу, записывающую его элементы в одномерный массив по строкам. При этом нечётные строки обходятся слева направо, а чётные справа налево.

7.3. Коллекции и их применение

7.3-0 Дана последовательность, содержащая шесть видов скобок: (,), [,], {, }. Написать программу, проверяющую, является ли скобочная запись, задаваемая последовательностью, корректной. Иными словами, программа должна проверить правильность вложенности пар скобок, отсутствие «незакрытых» и лишних «закрывающих» скобок.

Например, последовательность «([{}])» корректна, а «)[{}]]» нет.

7.3-1 Дана закодированная последовательность чисел. Написать декодирующую программу, работающую по следующему алгоритму. Если очередное число x положительное, то оно копируется в декодированную последовательность, если отрицательное, то копируется $|x|$ последних значений из декодированного потока.

Например, последовательность (1, 2, 3, -2, -3) будет декодирована как (1, 2, 3, 2, 3, 3, 2, 3).

7.3-2 Написать программу, заменяющую каждый элемент последовательности чисел на индекс его первого вхождения.

Например, последовательность (3, 5, 5, 3, 7) будет заменена на (0, 1, 1, 0, 4).

7.3-3 Написать программу, проверяющую, верно ли, что количество повторов какого-либо элемента последовательности равно его значению.

Например, последовательность (4, 2, 4, 2, 4, 1, 4) удовлетворяет этому условию.

7.3-4 Написать программу, которая каждый элемент последовательности дополнительно повторяет столько раз, сколько он встречался до этого.

Например, последовательность

(1, 2, 1, 2, 1)

будет преобразована в

(1, 2, 1, 1, 2, 2, 1, 1, 1).

7.3-5 Даны три числа a, b, N . Написать программу, выводящую все возможные числа $x \leq N$, которые можно получить из числа 1 при помощи операций «прибавить a » и «умножить на b ». Операции можно применять произвольное число раз в произвольном порядке.

Для решения этой задачи можно использовать очередь, хранящую числа, требующие проверки, удовлетворяют ли они условию задачи. Числа, производные от проверяемого, также добавляются в очередь.

Например, если $a = 3, b = 2, N = 8$, то программа должна вывести числа 1, 2, 4, 5, 7, 8.

7.3-6 Написать программу, рекурсивно вычисляющую n -й элемент последовательности f_n , заданный рекуррентным соотношением

$$f_n = \begin{cases} 2f_{n-1} + f_{\lfloor n/2 \rfloor}, & \text{если } n > 2; \\ 1, & \text{если } n \leq 1. \end{cases}$$

Ускорить вычисления с помощью мемоизации. Для этого все уже вычисленные промежуточные значения помещаются в словарь. Если требуется найти f_k , которое уже было посчитано ранее, используется значение из словаря.

7.3-7 Рассмотрим изменённый вариант задачи Иосифа Флавия. Числа от одного до некоторого N записаны по кругу. Затем каждое k -е (начиная с единицы) число вычёркивается, как в считалке с выбыванием. Процесс продолжается, пока не останется одно число. Написать программу, выводящую на экран последовательность вычеркивания чисел.

Например, если $N = 5, a k = 3$, то последовательность будет иметь вид

$$(3, 1, 5, 2, 4).$$

7.3-8 Даны два стека целых чисел. В первом содержатся некоторые целочисленные значения (данные), а во втором коды команд. Команды управляют процессом вычислений и могут быть одного из следующих видов:

- 1 — взять два значения из первого стека и положить обратно сумму;
- 2 — взять два значения из первого стека и положить обратно разность (верхушка — уменьшаемое);
- 3 — поменять местами два верхних значения первого стека;
- 4 — скопировать значение из второго стека в первый.

Перед выполнением команды извлекаются из второго стека. Если код команды отличается от перечисленных, она игнорируется.

Написать программу, выполняющую команды второго стека и затем выводящую на экран верхушку первого. Если какую-то команду выполнить невозможно — вывести на экран сообщение об ошибке.

Например, если стек данных имел вид (3, 2, 1), а стек команд — (1, 2) (верхушка слева), то результатом работы программы будет число 4.

7.3-9 Написать программу, реализующую следующий алгоритм слияния двух отсортированных последовательностей. Даны две последовательности чисел, отсортированных по возрастанию. Новая последовательность формируется при помощи многократного выбора из начала исходных последовательностей наименьшего элемента.

7.4. LINQ

7.4-0 В некоторой коллекции хранятся радиусы окружностей. Написать LINQ-выражение, выбирающее только те радиусы, для которых площади окружностей больше либо равны 4.

7.4-1 В некоторой коллекции хранятся значения углов в градусах. Написать LINQ-выражение, выбирающее только острые углы.

7.4-2 В некоторой коллекции хранятся целые числа. Написать LINQ-выражение, формирующее коллекцию их квадратов.

7.4-3 В некоторой коллекции хранятся углы в радианах. Написать LINQ-выражение, выбирающее только те углы, косинусы которых неотрицательны.

7.4-4 В некоторой коллекции хранятся целые числа. Написать LINQ-выражение, выбирающее только положительные двузначные числа.

7.4-5 В некоторой коллекции хранятся целые числа. Написать LINQ-выражение, выбирающее только чётные числа.

7.4-6 В некоторой коллекции хранятся вещественные числа. Написать LINQ-выражение, формирующее коллекцию с их целыми частями.

7.4-7 В некоторой коллекции хранятся целые числа. Написать LINQ-выражение, выбирающее только числа, заканчивающиеся на 7.

7.4-8 В некоторой коллекции хранятся целые числа. Написать LINQ-выражение, выбирающее только числа, делящиеся на 11.

7.4-9 В некоторой коллекции хранятся целые числа. Написать LINQ-выражение, формирующее коллекцию, содержащую их последние цифры.