

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМ. Р.Е. АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра «Информационные радиосистемы»

Обработка однородного набора структурированных данных

Методические указания
к контрольной работе по курсу
«Информационные технологии»
для студентов направления подготовки
бакалавра 11.03.01 «Радиотехника»
заочной формы обучения

Нижний Новгород 2025

Составитель: **С.Б. Сидоров**

УДК 519.6

Обработка однородного набора структурированных данных: Метод. указания к контрольной работе по курсу «Информационные технологии» для студентов направления подготовки бакалавра 11.03.01 «Радиотехника» заочной формы обучения / НГТУ; Сост.: С.Б. Сидоров. Н. Новгород, 2025.– 21 с.

Содержит методические материалы и подбор задач, предназначенных для отработки приёмов разработки программ. Определён порядок выполнения работы. Рассмотрен пример решения задачи и приведены требования к составлению отчёта. Может быть использовано студентами для самостоятельной работы по дисциплине «Информационные технологии», а также для выполнения контрольной работы по теме: «Обработка однородного набора структурированных данных».

Редактор _____

Подп. к печ. _____. Формат $60 \times 84^{1/16}$. Бумага газетная. Печать офсетная.
Печ. л. _____. Уч.-изд. л. _____. Тираж _____ экз. Заказ_

Нижегородский государственный технический университет им. Р.Е. Алексеева.
Типография НГТУ. 603950, Н. Новгород, ул. Минина, 24.

© Сидоров С.Б., 2025

1. ОБЩИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ КОНТРОЛЬНОЙ РАБОТЫ

Выполнение контрольной работы состоит в разработке программы на языке программирования С/С++ и формирования отчета. Для выполнения работы студент получает индивидуальное задание в виде номера варианта. Вариант включает формулировку задания. Варианты приведены в данных методических указаниях.

Под разработкой программы понимается анализ поставленной задачи, создание алгоритма обработки, разработка тестов для проверки правильности алгоритма и его программной реализации, реализация алгоритма в виде программы на языке С/С++, компиляция, тестирование и отладка программы. Разработка программы может выполняться с использованием имеющейся в распоряжении студента операционной системы и любой доступной среды разработки.

Выполненная контрольная работа отправляется на первичную проверку преподавателю по e-mail в виде единого архива. Обязательным требованием является указание в качестве имени архива ФИО (фамилия и инициалы) студента, выполнившего работу. Работы, отправленные с нарушением данного требования к проверке не принимаются с уведомлением студента. Архив должен содержать исходный код программы (файл с окончанием .cpp), а также отчет в **PDF**. Присылать исполняемые образы программ и целиком проекты, созданные в использованной среде разработки категорически не допускается. Также не рассматриваются отчеты, представленные в формате, отличном от PDF.

По результатам первичной проверки выставляется допуск работы к сдаче. При отрицательном решении необходимо выполнить доработку с целью устранения указанных замечаний по работе. В то же время положительное решение о допуске работы к сдаче не означает ее правильности в смысле отсутствия ошибок. В связи с этим настоятельно рекомендуется тщательно проверять правильность и полноту решения поставленной задачи даже после получения допуска работы к сдаче.

Сдача работы выполняется в виде индивидуального обсуждения с преподавателем полученных результатов выполнения работы. Обсуждение основывается на представленном отчете и касается полноты и качества выполнения работы, грамотного оформления отчета в соответствии с предъявляемыми требованиями, способности объяснить принятые решения по программной реализации. По результатам беседы выносится оценка. Оценивание успешности и качества выполнения

работы осуществляется по 100-бальной системе. Допускается досрочная сдача работы по индивидуальной договоренности между студентом и преподавателем.

2. ОСНОВНЫЕ ПОНЯТИЯ

2.1. Общая постановка задачи по контрольной работе

Рассматривается модель определенного понятия проблемной области. Описание понятия представлено совокупностью свойств, в общем случае разнотипных. В области программной реализации модель понятия имеет вид структурного типа данных.

Рассматривается набор экземпляров данного понятия с конечным количеством элементов. Требуется получить программную реализацию заданной обработки такого набора структурированных данных.

Все данные, необходимые для обработки, запрашиваются у пользователя, то есть считываются со стандартного устройства ввода. К таковым относятся значения элементов, входящих в набор и, возможно, дополнительная информация, необходимая для выполнения конкретной обработки данных.

Обработка должна быть реализована отдельной функцией. При этом вся необходимая для выполнения обработки информация должна передаваться в функцию через список аргументов. Результат обработки набора структурированных данных должен передаваться из функции также через список аргументов.

Полученные результаты обработки должны быть выданы на монитор, то есть стандартное устройство вывода.

2.2. Рекомендации по архитектуре программы

Указанное в задании понятие в программной реализации необходимо представлять в виде структурного типа данных. Набор полей структуры определяется из постановки задачи либо в явном виде, либо выделяя только существенные свойства, необходимые для выполнения указанной обработки.

Набор экземпляров данного понятия следует представлять в виде массива структур. Учитывая, что, хотя в постановке задачи отмечено конечное количество элементов в наборе, точное их количество не задано. Данный факт влечет необходимость запроса количества элементов набора у пользователя при вводе исходных данных. Учитывая ограничение языка программирования при использовании массива, ограничим количество

элементов в массиве некоторой константой, определяемой из эмпирических соображений.

Диалог с пользователем при задании исходных данных реализуем в основной функции программы, функции **main**.

Аналогично, выдачу результатов обработки данных реализуем непосредственно в основной функции программы, функции **main**.

Непосредственно программную реализацию обработки совокупности однородных структурированных данных следует представить отдельной функцией. Эта функция будет иметь ряд параметров, в которые входят:

- ✓ адрес области памяти, содержащей набор элементов (массив элементов);
- ✓ количество элементов в наборе;
- ✓ дополнительная информация, необходимая для выполнения обработки набора данных;
- ✓ результат обработки данных, который в общем случае может быть представлен более чем одним параметром.

Если дополнительная информация, необходимая для выполнения обработки набора данных, состоит более чем из одной величины, то удобно для нее также определить специальный структурный тип данных. Полями этой структуры будут величины, необходимые для выполнения обработки. При этом для передачи структуры в качестве параметра в функцию следует применять указатель, содержащий адрес структурной переменной.

Если результатом обработки является массив элементов, то кроме самого массива необходим еще один выходной параметр — количество элементов в результирующем массиве. Поскольку он является выходным параметром, то есть вычисляется самой функцией, то передаваться должен через указатель.

Пример программы, разработанной в соответствии с приведенными рекомендациями, приведен в разделе 5.

3. ВАРИАНТЫ КОНТРОЛЬНОЙ РАБОТЫ

Общая постановка задачи приведена в разделе 2.1. Ниже приводятся только варианты заданий, которые **в обязательном порядке** должны быть рассмотрены совместно с общей постановкой.

1. Рассматривается расписание движения пригородных электропоездов и поездов дальнего следования. Каждая запись в расписании должна содержать следующую информацию: тип транспорта, направление движения, время отправления, режим отправления (ежедневно, по четным/нечетным дням, кроме праздничных дней, конкретный день недели), время прибытия. Определить поезд указанного типа, время отправления которого наиболее подходит к заданному времени.

2. Рассматривается информация о сыгранных матчах в чемпионате по футболу. Каждая запись включает названия команд и результат матча (количество голов каждой из команд). Принять, что за победу начисляется 3 очка, за ничью - 1, за поражение - 0. В случае равенства очков преимущество получает команда, имеющая лучшую разность между забитыми и пропущенными мячами. Требуется получить статистику по заданной команде (количество набранных очков, забитых и пропущенных мячей).

3. Рассматривается информация о подписке жителей на газеты и журналы. Каждая запись содержит следующие атрибуты: фамилия подписчика, название издания, начальный и конечный месяцы, на которые сделана подписка. Необходимо подсчитать требуемое количество экземпляров заданного издания в каждый из месяцев в году.

4. Рассматривается набор точек на декартовой плоскости. Каждая точка определяется парой координат (x, y) . Задан набор окружностей с одинаковым центром и различными радиусами. Подсчитать количество точек, попавших в каждое из колец, образуемых окружностями.

5. Рассматривается информация о какой-либо коллекции (марок, монет, и т. д.). Сведения об отдельном предмете коллекции включают: название предмета, стоимость, год создания. Должна быть реализована операция отбора предметов коллекции, год создания которых не превосходит заданное значение и стоимость которых не меньше чем указанное значение.

6. Рассматривается набор точек на плоскости. Каждая точка задается парой координат (x, y) и приписанным к ней весом w . Требуется отобрать

точки, попавшие в заданную прямоугольную область и имеющие вес не меньше определённого значения.

7. Рассматривается информация о программе телевизионных передач на текущую неделю. Описание каждой передачи включает день недели, название телеканала, время передачи (начало и завершение), название передачи. Требуется выдать список названий передач, которые будут показаны в заданный день недели в выбранный интервал времени по всем телеканалам.

8. Рассматривается набор прямоугольных областей на плоскости. Каждая прямоугольная область задается координатами $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ со сторонами, ориентированными вдоль координатных осей. Требуется отобрать такие области, которые полностью лежат внутри некоторой заданной области и имеют площадь, не меньшую определённого значения.

9. Рассматривается информация о ресурсах библиотеки. Сведения о печатном издании включают в себя: название, год выпуска, жанр. Требуется найти все издания указанного жанра, вышедшие в указанный период времени.

10. Имеется таблично заданная функция, у которой каждому значению аргумента из определенного набора поставлено в соответствие значение функции, то есть $(x_i, f(x_i))$. Принять, что $x_i < x_{i+1}$. Требуется определить все локальные максимумы функции, находящиеся в указанной области изменения аргумента (x_{\min}, x_{\max}) .

11. Рассматривается информация из записной книжки. Каждая запись в книжке содержит данные об одном человеке и содержит следующую информацию: ФИО, телефон, дата рождения. Требуется составить список тех, у кого в указанные сроки (интервал по датам) будет день рождения.

12. На плоскости задана ломаная линия. Требуется определить узлы линии, угол излома в которых более указанного значения и оба примыкающих к узлу отрезка не менее определённой длины.

13. Рассматриваются сведения об имеющихся в наличии товарах в магазине. Каждый вид товара описывается следующим образом: название, цена, срок реализации (день, месяц, год). Требуется выдать список товаров, у которых срок реализации истекает ранее указанной даты.

14. Имеется набор комплексных чисел $z = x + i \cdot y$. Из набора выбрать те числа, модуль которых лежит в указанном интервале $(|z|_{\min}, |z|_{\max})$. Модуль комплексного числа производится по формуле $|z| = \sqrt{x^2 + y^2}$.

15. Рассматривается информация из системы «Ассортимент», содержащую сведения об игрушках. Каждый вид игрушки описывается следующим набором характеристик: название игрушки (кукла, конструктор, кубики и т. д.), цена, возрастные границы (например, игрушка предназначена для детей в возрасте от двух до четырех лет). Требуется получить названия игрушек, цена которых не превышает заданную сумму и подходящие детям до указанного возраста.

16. Набор содержит геометрические точки, заданные в полярных координатах (R, α) . Определить прямоугольник в плоскости Oxy , в который попадают все указанные точки $x = R \cdot \cos \alpha$, $y = R \cdot \sin \alpha$. Стороны прямоугольника выбрать ориентированными вдоль координатных осей.

17. Рассматривается информация о странах мира. Каждая страна описывается следующими атрибутами: название, часть света, численность населения (в тыс. человек), площадь (в тыс. кв. км). Реализовать операцию формирования списка стран из заданной части света, площадь которых меньше заданного значения.

18. Рассматривается информация о результатах баскетбольной встречи. Каждый игрок обеих команд описывается следующим набором атрибутов: ФИО, команда, номер игрока, очки. Вывести номер лучшего игрока встречи указанной команды и количество набранных им очков.

19. Рассматривается набор точек на плоскости. Каждая точка задается парой координат (x, y) и приписанным к ней весом w . Требуется отобрать две точки минимального веса, превосходящие указанное граничное значение веса.

20. Рассматривается информация о студентах факультета. Студент характеризуется следующим набором атрибутов: ФИО, номер группы, номер зачетной книжки, текущая успеваемость по дисциплинам (рассмотреть фиксированное число дисциплин, например, 5). Успеваемость оценивается по 5-бальной системе. Требуется выбрать студентов, имеющих задолженность по определенной дисциплине с указанием фамилии и группы каждого из таких студентов.

21. На декартовой плоскости рассматривается совокупность «окрашенных» кругов. Каждый круг определяется значениями следующих атрибутов: центр, радиус, цвет (синий или красный). Если круг целиком лежит внутри другого круга, то он считается «покрытым». Подсчитать количество покрытых кругов каждого цвета, площадь которых превосходит заданное значение.

22. Рассматриваются сведения о выполненных экспериментах. Каждый эксперимент описывается следующим набором характеристик: название, экспериментатор, дата эксперимента, имя файла с результатами. Дата эксперимента записывается в формате гггг/мм/дд-чч.мм.сс Требуется выдать список названий экспериментов, выполненных в указанный интервал времени. Интервал времени задается начальной и конечной датами в формате записи даты в описании эксперимента.

23. Рассматривается набор задач. Каждая задача характеризуется следующим набором атрибутов: название задачи, важность и срок её выполнения. Важность представляется в виде некоторого натурального числа, причем большее значение соответствует большей важности. Срок выполнения задается датой, как совокупности дня, месяца и года. Сформировать список задач, важность которых не менее заданной и срок выполнения которых не позднее указанной даты.

24. Имеется таблично заданная функция, у которой каждому значению аргумента из определенного набора поставлено в соответствие значение функции, то есть $(x_i, f(x_i))$. Принять, что $x_i < x_{i+1}$. Требуется получить набор всех интервалов $[x_i, x_{i+1}]$ из области определения функции, в которых выполняется переход через заданное значение функции y .

25. Имеется набор описаний видов деталей. Каждый вид детали характеризуется следующим набором атрибутов: название, размеры по длине и ширине, материал детали. Получить список названий видов деталей из указанного материала, размеры которых находятся в заданных диапазонах по длине и ширине.

26. В пространстве в декартовой системе координат задан набор сфер. Каждая сфера определяется положением центра и радиусом. Получить список всех пар пересекающихся в пространстве сфер. Описание отдельной пары содержит номера обеих сфер в исходном наборе.

27. Задана цепочка «костей» («камней») домино. Отдельная «кость» определяется парой неотрицательных целых чисел (левое и правое). В правильной цепочке соседние «кости» соприкасаются своими половинками с одинаковыми значениями чисел. Проверить, является ли заданная цепочка правильной. Для неправильной цепочки указать количество нарушений правила формирования цепочки.

28. Задан набор описаний функций вида $A \cdot \sin(\omega \cdot x + \varphi)$, определенных с точностью до значений параметров A, ω, φ . Получить список описаний функций, частоты которых лежат в указанном интервале $(\omega_{min}, \omega_{max})$, а значение амплитуды A превышает указанное нижнее пороговое значение

29. Имеется описание маршрута в виде набора названий остановок и расстояний для каждой из остановок от начальной точки маршрута. По заданным названиям остановок определить возможность использования этого маршрута для перемещения между данными остановками. При положительном решении сформировать список промежуточных остановок, а также подсчитать общее расстояние.

30. Имеется набор прямоугольников, ориентированных вдоль координатных осей декартовой плоскости. Сформировать из них список прямоугольников, площадь которых превышает некоторое заданное значение и полностью лежащих в заданной прямоугольной области декартовой плоскости.

31. Из набора точек на декартовой плоскости сформировать новый набор, в который включаются только те точки, которые находятся вне окружности с центром (x_c, y_c) и радиусом R .

32. Имеется набор точек, заданных на декартовой плоскости. Также задана прямая, которая проходит через указанные две точки. Из исходного набора сформировать два, в каждом из которых находятся точки, лежащие по одну сторону относительно прямой. Точки, которые оказываются на прямой, можно поместить в произвольный набор.

33. Рассматривается информация из записной книжки. Каждая запись в книжке содержит данные об одном человеке и содержит следующую информацию: ФИО и дата рождения. Требуется определить дату ближайшего предстоящего дня рождения относительно некоторой заданной даты и у кого будет день рождения. Учесть, что может быть несколько человек, у которых день рождения совпадает.

34. Задан набор описаний поставок товаров. Описание поставки включает название товара и срок (количество дней). Набор может содержать несколько поставок товара с одним именем. Определить для каждого названия товара минимальный срок поставки.

35. Имеется журнал событий некоторого производственного процесса. Описание события включает название события и его дату. Получить распределение количества событий заданного названия по месяцам.

36. Задан набор многократных измерений некоторой величины несколькими датчиками. Описание отдельного измерения содержит серийный номер датчика и числовое значение величины. Сформировать

набор разброса значений (минимальное и максимальное) по каждому из датчиков.

37. Задан набор измерений некоторой величины двумя датчиками. Отдельный элемент набора включает результат измерения от каждого из датчиков. Задан диапазон допустимых значений величины. Сформировать список ситуаций «подтвержденного» выхода значения величины за пределы диапазона. Выход считать «подтвержденным», если значения от обоих датчиков вышли за пределы диапазона в одну сторону. Элемент результирующего списка кроме значений от датчиков должен включать и «время» события, то есть номер измерения в исходном наборе.

38. Имеется набор описаний поставок различных изделий. Описание поставки включает код изделия, количество экземпляров и дату поставки (день, месяц и год). Для каждого кода изделия рассчитать количество поставленных экземпляров в течение заданного периода времени.

39. Задан набор выпуклых четырехугольников, размещенных на декартовой плоскости. Требуется сформировать набор треугольников, образованных разделением по диагонали каждого четырехугольника на два треугольника. Для разделения следует использовать диагональ меньшей длины. При одинаковых длинах выбор произволен.

40. На декартовой плоскости рассматривается совокупность кругов. Каждый круг определяется значениями следующих атрибутов: центр и радиус. Сформировать набор кругов, которые полностью или частично перекрываются заданным прямоугольником. Принять, что прямоугольник ориентирован вдоль координатных осей декартовой плоскости.

41. Имеется набор описаний пациентов. Описание каждого пациента включает его ФИО и дату последнего посещения. Для указанной даты составить список пациентов, для которых время, прошедшее с последнего посещения, превысило заданное количество месяцев.

42. Задан набор описаний инженерных приборов. Описание прибора состоит из его названия, номера подразделения, за которым он закреплен и год приобретения. Получить список приборов, закрепленных за указанным подразделением и приобретенных ранее указанного года.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

При выполнении контрольной работы следует неукоснительно соблюдать этапы ее выполнения в соответствии с приведенными ниже рекомендациями. Отчетность по завершении каждого этапа не является обязательной, за исключением специально оговоренных случаев. Однако по желанию студента предварительная демонстрация промежуточных результатов на консультациях и в прочие дни (по предварительной договоренности с преподавателем) возможна или даже необходима при возникновении каких-либо затруднений, связанных с выполнением работы.

1. Информационное обеспечение выполнения контрольной работы. На этом этапе студент получает необходимые сведения по организации выполнения работы, доступ к справочным, примерным и прочим материалам, необходимым для выполнения работы.

2. Выбор темы контрольной работы. Контрольная работа выполняется в соответствии с индивидуальным заданием. Задание включает в себя общую постановку задачи и определенный вариант, в котором содержится конкретная формулировка темы из указанной проблемной области. Вариант выбирается для каждого студента преподавателем из предлагаемого в данном пособии перечня тем (раздел 3). Информация о распределении вариантов передается преподавателем одновременно в виде списка тем студентов группы.

3. Изучение материала, необходимого для выполнения контрольной работы и разбор примера выполнения работы. На этом этапе вначале осуществляется анализ общей постановки задачи. Затем прорабатываются рекомендации по архитектуре разрабатываемой программы для решения выбранной задачи в соответствии с разделом 2.2. Завершается выполнение этапа детальным разбором примера разработки программы, приведённого в разделе 5,

4. Поэтапная разработка приложения. На основе общей постановки и выбранного варианта выполняется анализ задачи. По результатам анализа принимается решение о необходимых пользовательских типах данных как для программного описания понятия, используемого в задаче, так и набора параметров обработки. В соответствии с формулировкой задачи разрабатывается алгоритм обработки данных. Он записывается либо в виде словесного описания, в свободной форме, но четко и со всеми подробностями, либо в виде формальной блок-схемы по желанию. Затем должна быть получена программная реализация алгоритма в виде функции с необходимым

набором аргументов. Заключительной частью выполнения этапа является разработка основной управляющей программы в соответствии с требованиями из общей постановки задачи. При этом выполняется тестирование полученной программы и при необходимости ее доработка с целью устранения ошибок и выявленных несоответствий требованиям.

5. Составление отчета. На данном этапе в соответствии с требованиями, приведёнными в разделе 6, составляется отчет о выполненной контрольной работе. При этом настоятельно рекомендуется создавать отчет в соответствии с примером, приведенным там же. Для подготовки отчета могут быть использованы любые программные средства, имеющиеся в наличии у студента

6. Получение допуска к сдаче контрольной работы. На этапе проводится первичная проверка работы в соответствии с правилами, изложенными в разделе 1 до получения положительного решения о допуске работы к сдаче.

7. Подготовка к сдаче контрольной работы. На этапе студент проверяет полученные результаты выполнения работы на соответствие требованиям. Средствами самоконтроля повторяет как новый, так и ранее известный теоретический материал, использованный при выполнении задания.

8. Индивидуальная сдача работы с получением оценки. Сдача работы выполняется в соответствии с правилами, изложенными в разделе 1

5. ПРИМЕР ВЫПОЛНЕНИЯ КОНТРОЛЬНОЙ РАБОТЫ

В данной главе приведен пример выполнения контрольной работы «Поиск прямых линий, пересекающих ось абсцисс в заданном интервале изменения абсциссы». Рассматривается полный цикл разработки программы, начиная с постановки задачи и заканчивая составлением отчета о выполненной работе. Приводятся общие рекомендации по структуре разрабатываемых программ и их оформлению.

5.1. Постановка задачи

Рассматривается модель определенного понятия проблемной области. Описание понятия представлено совокупностью свойств, в общем случае разнотипных. В области программной реализации модель понятия имеет вид структурного типа данных.

Рассматривается набор экземпляров данного понятия с конечным количеством элементов. Требуется получить программную реализацию заданной обработки такого набора структурированных данных.

Все данные, необходимые для обработки, запрашиваются у пользователя, то есть считываются со стандартного устройства ввода. К таковым относятся значения элементов, входящих в набор и, возможно, дополнительная информация, необходимая для выполнения конкретной обработки данных.

Обработка должна быть реализована отдельной функцией. При этом вся необходимая для выполнения обработки информация должна передаваться в функцию через список параметров. Результат обработки набора структурированных данных должен передаваться из функции также через список параметров.

Полученные результаты обработки должны быть выданы на монитор, то есть стандартное устройство вывода.

Задан набор прямых линий. Прямая однозначно определяется парой параметров a, b и задается уравнением вида $y = a \cdot x + b$. Требуется отобрать те прямые из набора, корень которых лежит в пределах указанного интервала $[xleft, xright]$.

5.2. Определение пользовательских типов данных

В соответствии с требованиями к программе для описания прямой определяем структурный тип данных. Дадим ему содержательное имя, поясняющее суть описываемого понятия, **Line**. Из постановки задачи явно следует, что прямая однозначно задается значениями двух свойств (величин): a и b . Каждому из этих свойств соответствует свое поле в структурном типе данных. Таким образом получаем следующее определение структурного типа данных для описания понятия «Прямая»:

```
typedef struct {  
    double a;  
    double b;  
} Line;
```

Для представления набора прямых используем массив из элементов типа **Line**. Максимально допустимое количество элементов в массиве ограничим константой **MaxLinesCount** со значением **20**.

Для выполнения заданной обработки необходима дополнительная информация, а именно координаты интервала $[xleft, xright]$. Поскольку в данном случае количество величин более одного, для представления в программе этой информации определим специальный структурный тип данных **ProcessingParameters**. Полями этой структуры будут величины, необходимые для выполнения обработки, а именно **xleft** и **xright**:

```
typedef struct {  
    double xleft;  
    double xright;  
} ProcessingParameters;
```

5.3. Представление функции обработки

Непосредственно программную реализацию обработки совокупности однородных структурированных данных представим отдельной функцией с содержательным именем, поясняющим ее назначение **findLinesWithRoot**.

Прежде всего эта функция должна получать в качестве двух первых параметров адрес области памяти (**source**), содержащей набор элементов (массив элементов) и количество элементов в наборе (**sourceCount**) соответственно.

Кроме этого, поскольку для обработки необходима дополнительная информация, то третьим параметром функция будет получать именно ее. При этом для передачи структуры в качестве параметра в функцию будем применять указатель **procParam**, содержащий адрес структурной переменной.

Из постановки задачи следует, что результатом обработки является массив элементов. Поэтому последними параметрами функции обработки будут два указателя: один на область памяти (**result**), содержащей результирующий массив, а второй (**resultCount**) на переменную, в которой будет сохраняться количество найденных прямых, удовлетворяющих условию задачи.

Окончательно, для функции обработки получаем следующее описание:

```
void findLinesWithRoot(Line* source,  
                        unsigned sourceCount,  
                        ProcessingParameters* procParam,  
                        Line* result,  
                        unsigned* resultCount);
```

5.4. Управляющая программа

Управляющая программа представлена функцией **main**. Схема работы управляющей программы сводится к последовательному выполнению следующих этапов:

1. Ввод исходных данных
 - 1.1. Ввод количества прямых линий
 - 1.2. Ввод значений прямых линий
 - 1.3. Ввод параметров обработки

- 2. Обработка набора прямых
 - 2.1. Выполнение функции обработки
- 3. Выдача результатов
 - 3.1. Выдача количества найденных прямых
 - 3.2. Выдача значений прямых линий

5.5. Определение нахождения корня в пределах заданного отрезка

Определение нахождения корня в пределах заданного отрезка математически сводится к вычислению значения функции $y=a \cdot x+b$ на границах отрезка и сравнению знаков получившихся значений.

Если знаки различаются или хотя бы одно из значений равно нулю, то это устанавливает факт наличия корня. В противном случае корня в пределах отрезка для прямой нет.

В программной реализации проверка указанного условия сводится к определению знака произведения значений функции на границах отрезка.

6. СОСТАВЛЕНИЕ ОТЧЕТА

Цель составления студентами отчета о выполненной контрольной работе заключается в том, чтобы овладеть навыками документирования программного обеспечения и закрепить полученные при изучении базового курса знания и умения использования стандартных программных средств для создания документов. В данном разделе приводятся требования к отчету.

6.1. Общая структура отчета

Отчет о выполнении контрольной работы должен содержать **титульный лист** с названием работы, фамилиями студента и преподавателя, а также названием университета, города и года выполнения работы. На следующей после титульного листа странице нужно привести содержание отчета, содержащее перечень его разделов с указанием соответствующих страниц.

Первым разделом является **введение**, в котором рекомендуется дать краткое обоснование важности темы, возможные области применения, а также охарактеризовать структуру работы и привести список принятых условных обозначений.

За введением следует **основная часть**, которая должна включать в себя следующие разделы:

- **Постановка задачи.** Содержит общую постановку задачи и полную формулировку варианта, перечень исходных данных и требуемых результатов, а также контрольные примеры для уточнения постановки задачи.
- **Руководство пользователя.** Включает общее описание программы, решающей поставленную задачу (назначение, возможности, выполняемые операции) и правила ее использования.
- **Руководство программиста.** Должно содержать укрупненную схему программы, описание наиболее важных или интересных алгоритмов решения поставленной задачи, описание структур данных и структуры их хранения.

За основной частью следует **заключение**. В нем необходимо перечислить основные результаты выполненной работы, привести выводы и рекомендации.

В конце отчета приводится **список литературы и приложение**. В приложение выносятся тексты программ, диаграммы и прочие материалы, которые не желательно включать в основную часть отчета вследствие его обширности или меньшей значимости.

6.2. Пример оформления отчёта по контрольной работе

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМ. Р.Е. АЛЕКСЕЕВА

ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ И ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ

Кафедра «Информационные радиосистемы»

КОНТРОЛЬНАЯ РАБОТА

***«Поиск прямых линий, пересекающих ось абсцисс
в заданном интервале изменения абсциссы»***

Выполнил:
студент группы 25-Рз И. И. Иванов

Проверил:
доцент кафедры ИРС С. Б. Сидоров

Нижний Новгород 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1.ПОСТАНОВКА ЗАДАЧИ.....	4
2.РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ.....	5
3.РУКОВОДСТВО ПРОГРАММИСТА.....	7
3.1. Структура программы.....	7
3.2. Структуры данных.....	7
3.3. Алгоритм определения наличия корня.....	8
ЗАКЛЮЧЕНИЕ.....	9
СПИСОК ЛИТЕРАТУРЫ.....	10
ПРИЛОЖЕНИЕ А. ОСНОВНАЯ ПРОГРАММА	11

Введение

В работе рассматривается решение задачи разработки прикладной программы поиска прямых линий, пересекающих ось абсцисс в заданном интервале изменения абсциссы.

Одним из способов преодоления сложности данной задачи является использование отдельной функции для выполнения обработки. Этот подход позволяет свести задачу к последовательному решению более простых задач. Его использование позволяет уменьшить затраты на отладку и повысить надёжность программы. Также существенным моментом является использование структурных типов данных для адекватного отображения сущностей проблемной области в область программной реализации.

В начале основной части отчёта приводится точная формализованная постановка задачи с указанием полного набора операций, выполнение которых должна обеспечивать прикладная программа.

В руководстве пользователя раскрывается назначение программы, её возможности и выполняемые операции. Подробно объясняются правила пользования программой и приводятся конкретные примеры диалога с пользователем.

В руководстве программиста рассматриваются вопросы внутренней организации программы, в том числе перечень функций и их взаимодействие. Кроме этого описываются используемые структуры данных и наиболее важные, и интересные с точки зрения их реализации алгоритмы.

В заключении делаются выводы о полноте решения поставленной задачи, а также приводится ряд соображений по возможным направлениям доработки полученной прикладной программы.

Также приведена техническая информация, включающая листинги программы.

1. Постановка задачи

Рассматривается модель определенного понятия проблемной области. Описание понятия представлено совокупностью свойств, в общем случае разнотипных. В области программной реализации модель понятия имеет вид структурного типа данных.

Рассматривается набор экземпляров данного понятия с конечным количеством элементов. Требуется получить программную реализацию заданной обработки такого набора структурированных данных.

Все данные, необходимые для обработки, запрашиваются у пользователя, то есть считываются со стандартного устройства ввода. К таковым относятся значения элементов, входящих в набор и, возможно, дополнительная информация, необходимая для выполнения конкретной обработки данных.

Обработка должна быть реализована отдельной функцией. При этом вся необходимая для выполнения обработки информация должна передаваться в функцию через список аргументов. Результат обработки набора структурированных данных должен передаваться из функции также через список аргументов.

Полученные результаты обработки должны быть выданы на монитор, то есть стандартное устройство вывода.

Задан набор прямых линий. Прямая однозначно определяется парой параметров a, b и задается уравнением вида $y = a \cdot x + b$. Требуется отобрать те прямые из набора, корень которых лежит в пределах указанного интервала $[x_{left}, x_{right}]$.

2. Руководство пользователя

Программа предназначена для поиска прямых линий, имеющих корень в заданном интервале изменения аргумента в режиме диалога с пользователем. Программа позволяет пользователю задать набор прямых и указать критерии поиска. Результаты обработки будут выданы на монитор.

Запуск программы на выполнение осуществляется либо набором в командной строке имени исполняемого образа программы **roots**, полученного в результате компиляции, с последующим нажатием на клавишу **Enter**, либо каким-то другим способом, зависящем от конкретной операционной системы.

После запуска программы на экране монитора появляется краткое информационное сообщение о программе следующего вида:

Программа «отбор прямых»
Автор: Сидоров С.Б.

Программа является интерактивным консольным приложением. Весь диалог с пользователем осуществляется в текстовом режиме.

После информационного сообщения пользователю предлагается указать количество прямых, среди которых будет выполняться отбор.

Введите количество элементов в наборе:

После ввода количества элементов пользователю последовательно предлагается ввести параметры прямых. При этом на экране монитора появляются запросы на ввод в следующем виде:

укажите параметры 0-ой линии:

Пользователю необходимо ввести сначала параметр a , а затем через пробел значение параметра b прямой. После указания пользователем параметров всех прямых осуществляется запрос значений критерия отбора в следующем виде:

Введите интервал для нахождения корня:

Пользователь обязан указать сначала левую границу отрезка, а затем через пробел правую границу отрезка поиска.

По завершении ввода значений критерия ввод всех исходных данных завершен. По завершении обработки на экран будет выдан результат проведенного отбора в следующем виде:

Всего найдено 2 прямых
3 0
1 -2

Здесь вначале отображается общее количество отобранных прямых, а затем описания самих прямых. Описание прямой состоит из двух значений: первое соответствует значению параметра a , а второе - b . При этом описание каждой прямой приводится на отдельной строке.

По завершении выдачи результатов программа завершает свою работу.

3. Руководство программиста

3.1 Структура программы

Прикладная программа разработана с использованием принципов императивного программирования. Она является совокупностью взаимодействующих функций. Структура программы представлена на рис. 1.

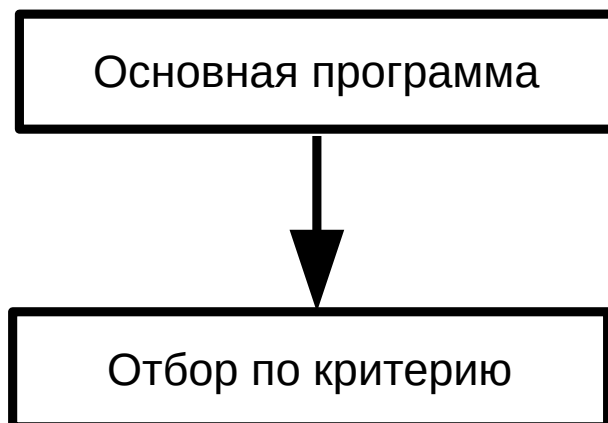


Рисунок 1: Структура программы

Программа состоит из 2 функций, назначение каждой из которых приведено ниже:

1. **main** – основная программа приложения;
2. **findLinesWithRoot** – функция отбора прямых по заданному критерию.

3.2 Структуры данных

Для описания прямой определяем структурный тип данных **Line**. Из постановки задачи явно следует, что прямая однозначно задается значениями двух свойств (величин): a и b . Каждому из этих свойств соответствует свое поле в структурном типе данных. Таким образом получаем следующее определение структурного типа данных для описания понятия «Прямая»:

```
typedef struct {  
    double a;  
    double b;  
} Line;
```

Для представления набора прямых используем массив из элементов типа **Line**. Максимально допустимое количество элементов в массиве ограничим константой **MaxLinesCount** со значением **20**.

Для выполнения заданной обработки необходима дополнительная информация, а именно координаты интервала $[x_{left}, x_{right}]$. Поскольку в данном случае количество величин более одного, для представления в программе этой информации определим специальный структурный тип данных **ProcessingParameters**. Полями этой структуры будут величины,

необходимые для выполнения обработки, а именно **xleft** и **xright**:

```
typedef struct {  
    double xleft;  
    double xright;  
} ProcessingParameters;
```

3.3 Алгоритм определения наличия корня

Определение нахождения корня в пределах заданного отрезка математически сводится к вычислению значения функции $y=a \cdot x+b$ на границах отрезка и сравнению знаков получившихся значений.

Если знаки различаются или хотя бы одно из значений равно нулю, то это устанавливает факт наличия корня. В противном случае корня в пределах отрезка для прямой нет.

В программной реализации проверка указанного условия сводится к определению знака произведения значений функции на границах отрезка.

Заключение

В данной работе задача разработки прикладной программы поиска прямых линий, пересекающих ось абсцисс в заданном интервале изменения абсциссы была решена с использованием императивного программирования. На основании проведённой отладки и испытаний с помощью контрольных примеров можно сделать вывод, что полученная прикладная программа решает поставленную задачу правильно и в полном объёме.

Список литературы

1. Керниган, Брайан У., Ритчи, Деннис М. Язык программирования С, 2-е издание. [пер. с англ.] / Б.У. Керниган, Д.М. Ритчи – М.: Вильямс, 2007.
2. Павловская, Т.А. С/С++. Программирование на языке высокого уровня: учебник для ВУЗов / Т.А. Павловская. – СПб.: Питер, 2009.
3. Орлов, С.А. Технологии разработки программного обеспечения. учеб. пособие. 2-е изд./ С.А. Орлов, – СПб.: Питер, 2003. – 480 с.: ил.
4. Борисенко, В.В. Основы программирования / В.В.Борисенко, – Интернет-университет информационных технологий – ИНТУИТ.ру, 328 стр. – 2005 г.
5. Шилдт, Г. Полный справочник по С: учеб. пособие / Г. Шилдт. – 4-е изд. – М.: Изд. дом "Вильямс", 2008.
6. Костюкова, Н.И. Язык Си и особенности работы с ним / Н.И. Костюкова, Н.А. Калинина – Интернет-университет информационных технологий – ИНТУИТ.ру, 208 стр. – 2006 г.

Приложение А

Основная программа

```
//roots.c
#include <stdio.h>

//описание прямой  $a \cdot x + b$ 
typedef struct {
    double a;
    double b;
} Line;

//описание параметров обработки
//интервал для нахождения корня [xleft, xright]
typedef struct {
    double xleft;
    double xright;
} ProcessingParameters;

//максимальное допустимое количество линий в наборе
const unsigned MaxLinesCount = 20;

//прототип функции обработки данных
void findLinesWithRoot(Line* source,
                        unsigned sourceCount,
                        ProcessingParameters* procParam,
                        Line* result,
                        unsigned* resultCount);

int main() {

    printf("Программа: отбор прямых\n");
    printf("Автор: Сидоров С.Б.\n");

    Line array[MaxLinesCount];

    //ввод набора прямых
    printf("Введите количество элементов в наборе:");
    unsigned arrayCount;
    scanf("%d",&arrayCount);

    for(unsigned i=0; i<arrayCount; ++i) {
        printf("укажите параметры %d-ой линии:", i);
        scanf("%lf%lf", &array[i].a, &array[i].b);
    }

    //ввод параметров обработки
    printf("Введите интервал для нахождения корня:");
    ProcessingParameters params;
    scanf("%lf%lf",&params.xleft, &params.xright);

    //результаты обработки
    Line result[MaxLinesCount];
    unsigned resultCount;
```

```

//выполнение обработки
findLinesWithRoot( array, arrayCount, &params,
                    result, &resultCount );

//выдача результатов обработки
printf("Всего найдено %d прямых\n", resultCount);
for(unsigned i=0; i<resultCount; ++i) {
    printf("%lf %lf\n", result[i].a, result[i].b);
}

return 0;
}

void findLinesWithRoot(Line* source,
                        unsigned sourceCount,
                        ProcessingParameters* procParam,
                        Line* result,
                        unsigned* resultCount) {

    unsigned found = 0; //количество найденных прямых
    for(unsigned i=0; i<sourceCount; ++i) {
        double leftValue = source[i].a * procParam->xleft+source[i].b;
        double rightValue = source[i].a*procParam->xright+source[i].b;
        if ( leftValue * rightValue <= 0 ) {
            //найдена прямая с корнем внутри интервала
            //занести ее в выходной набор
            result[found] = source[i];
            ++found; //изменить количество найденных
        }
    }

    //сохранить количество найденных в выходном параметре
    *resultCount = found;

    return;
}

```

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Керниган, Брайан У., Ритчи, Деннис М. Язык программирования С, 2-е издание. [пер. с англ.] / Б.У. Керниган, Д.М. Ритчи – М.: Вильямс, 2007.
2. Павловская, Т.А. С/С++. Программирование на языке высокого уровня: учебник для ВУЗов / Т.А. Павловская. – СПб.: Питер, 2009.
3. Орлов, С.А. Технологии разработки программного обеспечения. учеб. пособие. 2-е изд./ С.А. Орлов, – СПб.: Питер, 2003. – 480 с.: ил.
4. Борисенко, В.В. Основы программирования / В.В.Борисенко, – Интернет-университет информационных технологий – ИНТУИТ.ру, 328 стр. – 2005 г.
5. Шилдт, Г. Полный справочник по С: учеб. пособие / Г. Шилдт. – 4-е изд. – М.: Изд. дом "Вильямс", 2008.
6. Костюкова, Н.И. Язык Си и особенности работы с ним / Н.И. Костюкова, Н.А. Калинина – Интернет-университет информационных технологий – ИНТУИТ.ру, 208 стр. – 2006 г.