**ДОМАШНЕЕ ЗАДАНИЕ № 1**

по курсу «Основы программирования на C++»

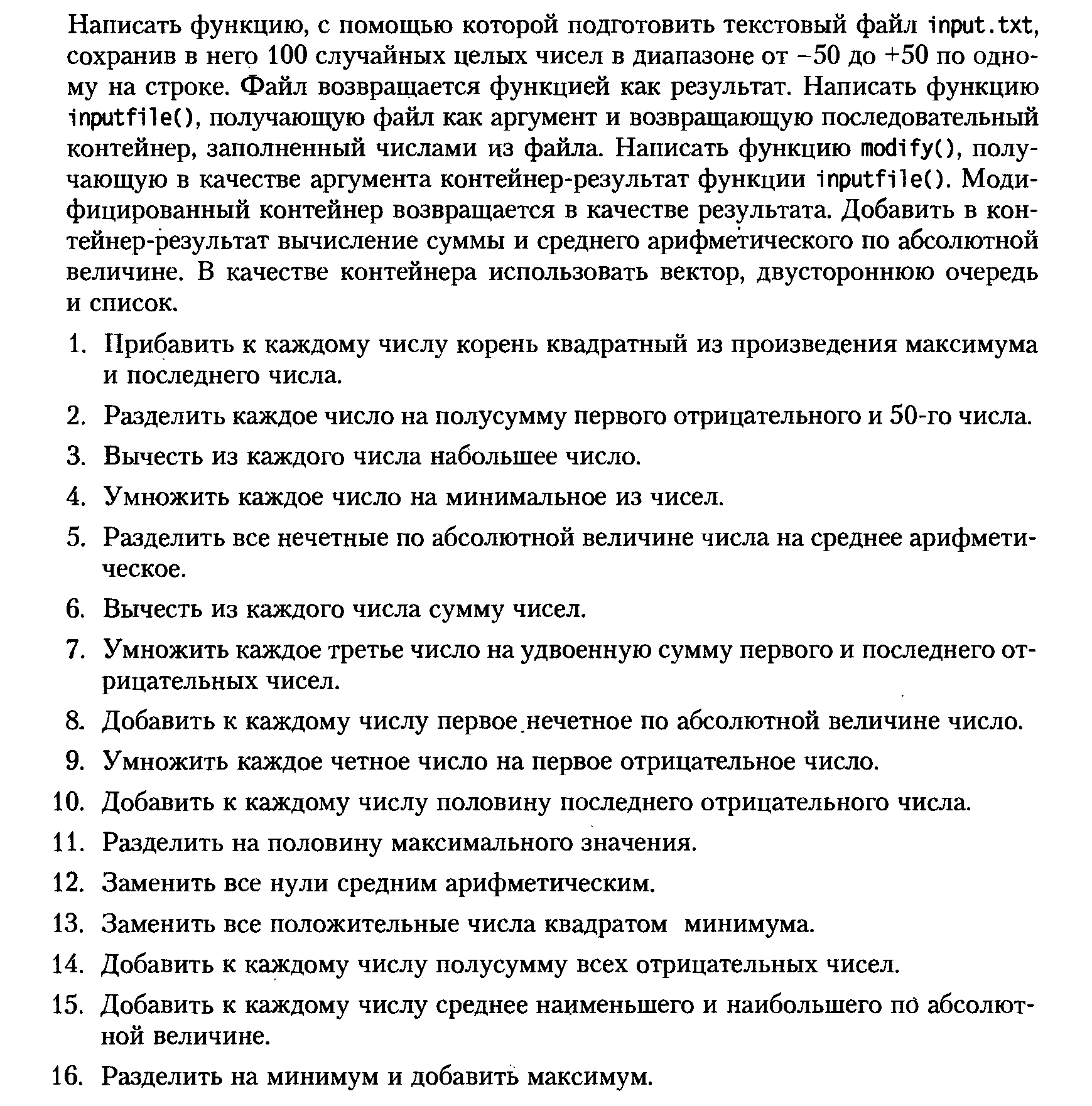
**Потоки ввода-вывода и стандартная библиотека шаблонов (STL)**

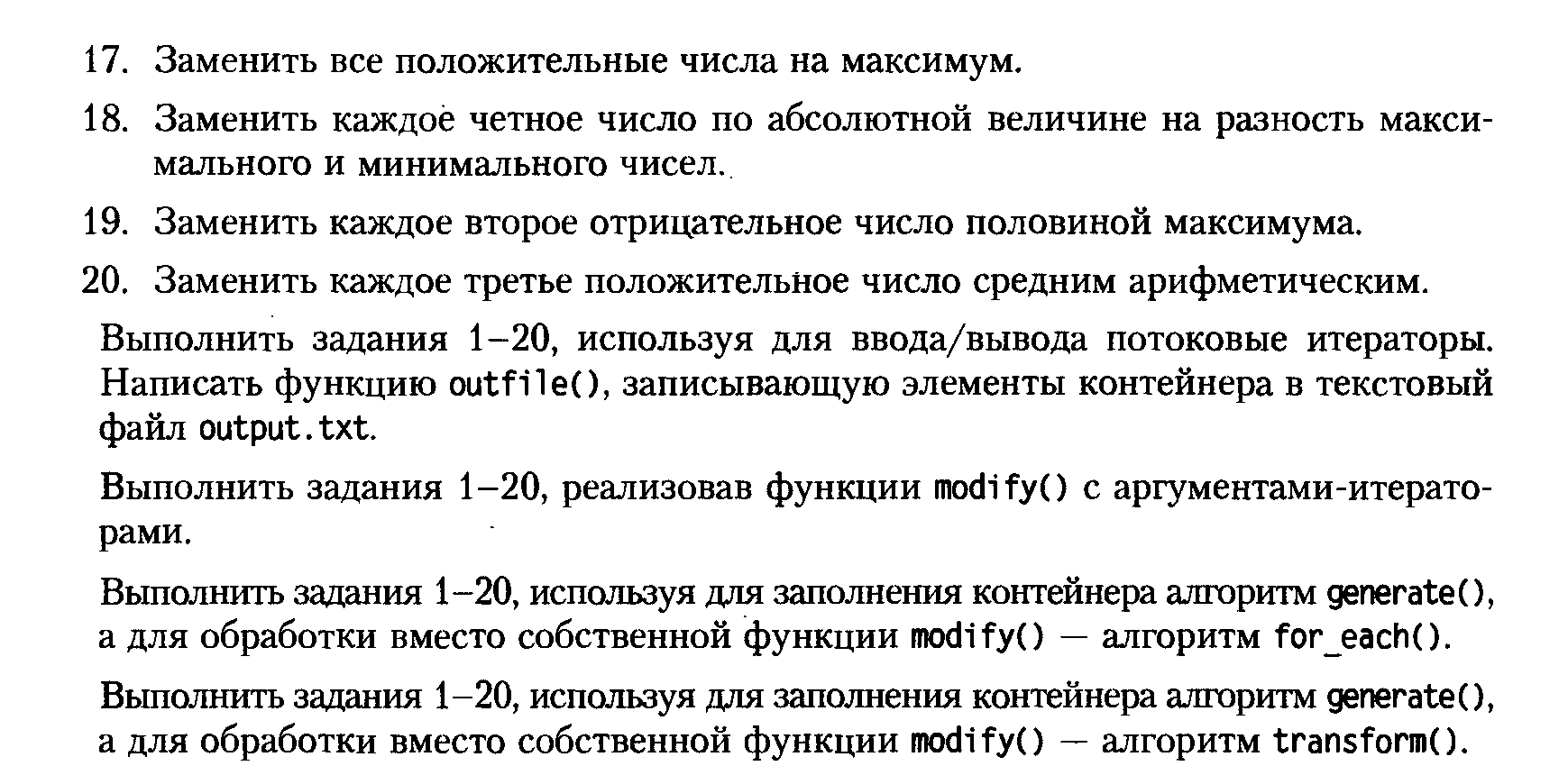
Для каждой части предоставить *меню*, позволяющее продемонстрировать основные возможности программы.

**Часть 1. Последовательные контейнеры**

Разработать следующий набор функций, используя в качестве последовательных контейнеров вектор, двустороннюю очередь и список.

1. Функция createData(), создающая *текстовый* файл с данными в виде 100 случайно сгенерированных целых чисел в диапазоне от –50 до 50. Предусмотреть следующие версии данной функции:
   1. Функция, выгружающая данные с использованием файловых потоков.
   2. Функция, использующая алгоритм generate для инициализации вспомогательного контейнера и алгоритм copy для вывода его содержимого в файл с использованием потоковых итераторов.
2. Шаблонная функция loadData(), загружающая данные в последовательный контейнер данного типа. Предусмотреть следующие версии данной функции:
   1. Функция, загружающая данные с использованием файловых потоков.
   2. Функция, загружающая данные с использованием *потоковых итераторов* и алгоритма copy.
3. Функция modify(), модифицирующая загруженные данные в соответствии с вариантами:





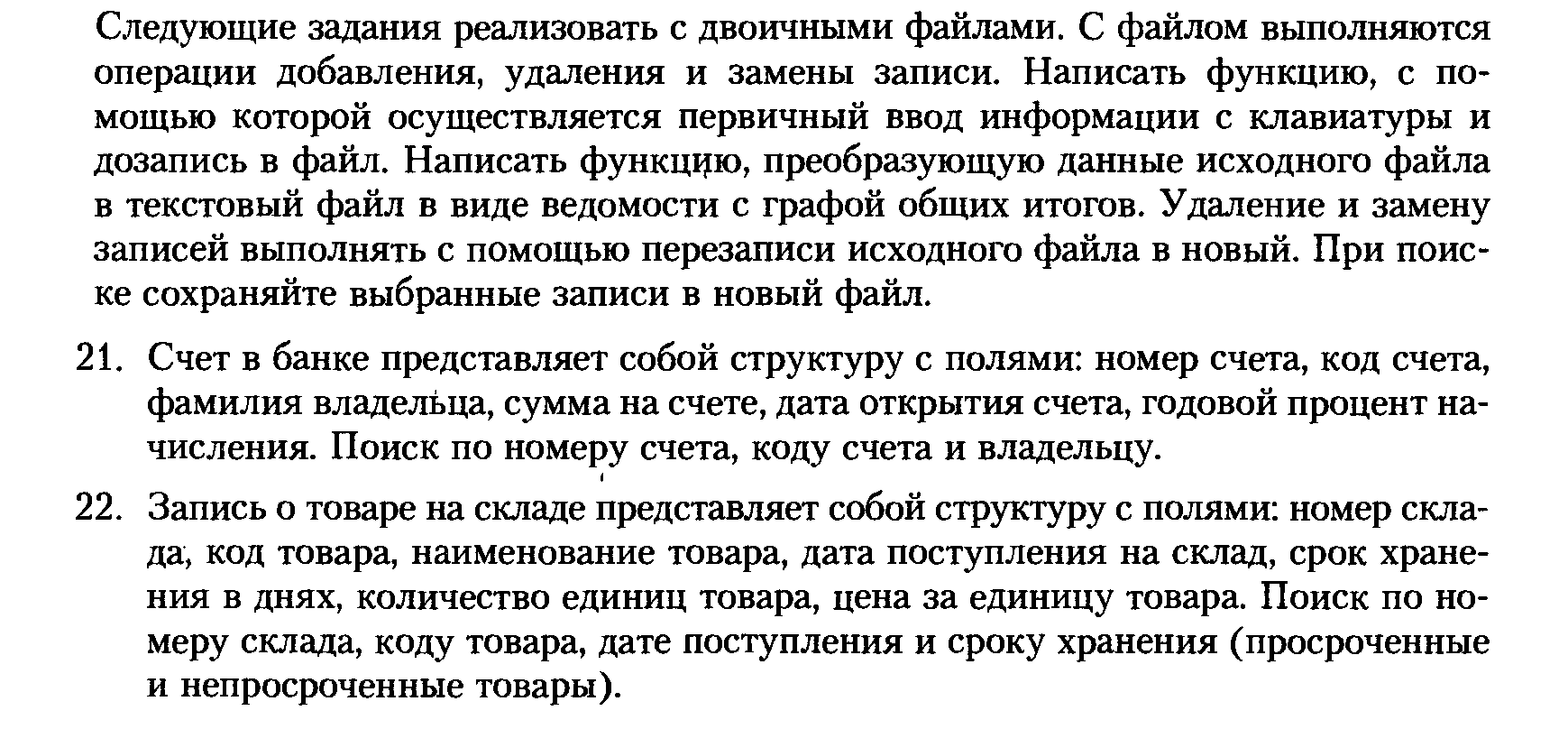
Предусмотреть следующие версии функции Modify():

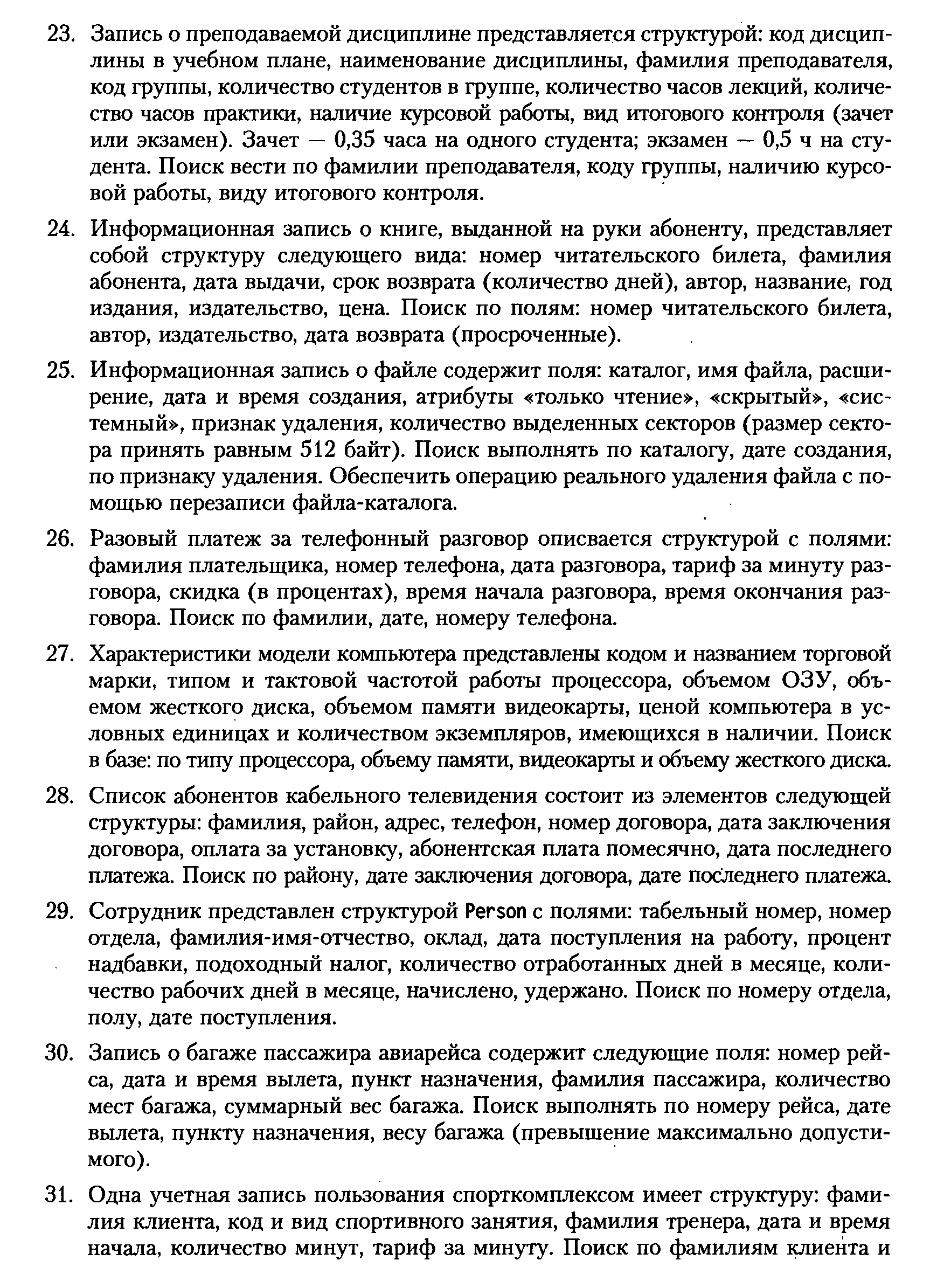
* 1. Как шаблонная функция, принимающая последовательный контейнер данного типа. Должна добавлять в конец контейнера результата сумму и среднее арифметическое элементов контейнера по абсолютной величине.
  2. Как функция, принимающая аргументы-итераторы.
  3. Как функция, использующая алгоритм for\_each(). Конкретную операцию передавать в виде функционального объекта или лямбда-функции.
  4. Как функция, использующая алгоритм transform() и потоковые итераторы для одновременного считывая данных из файла, их преобразования и вывода в файл результата.

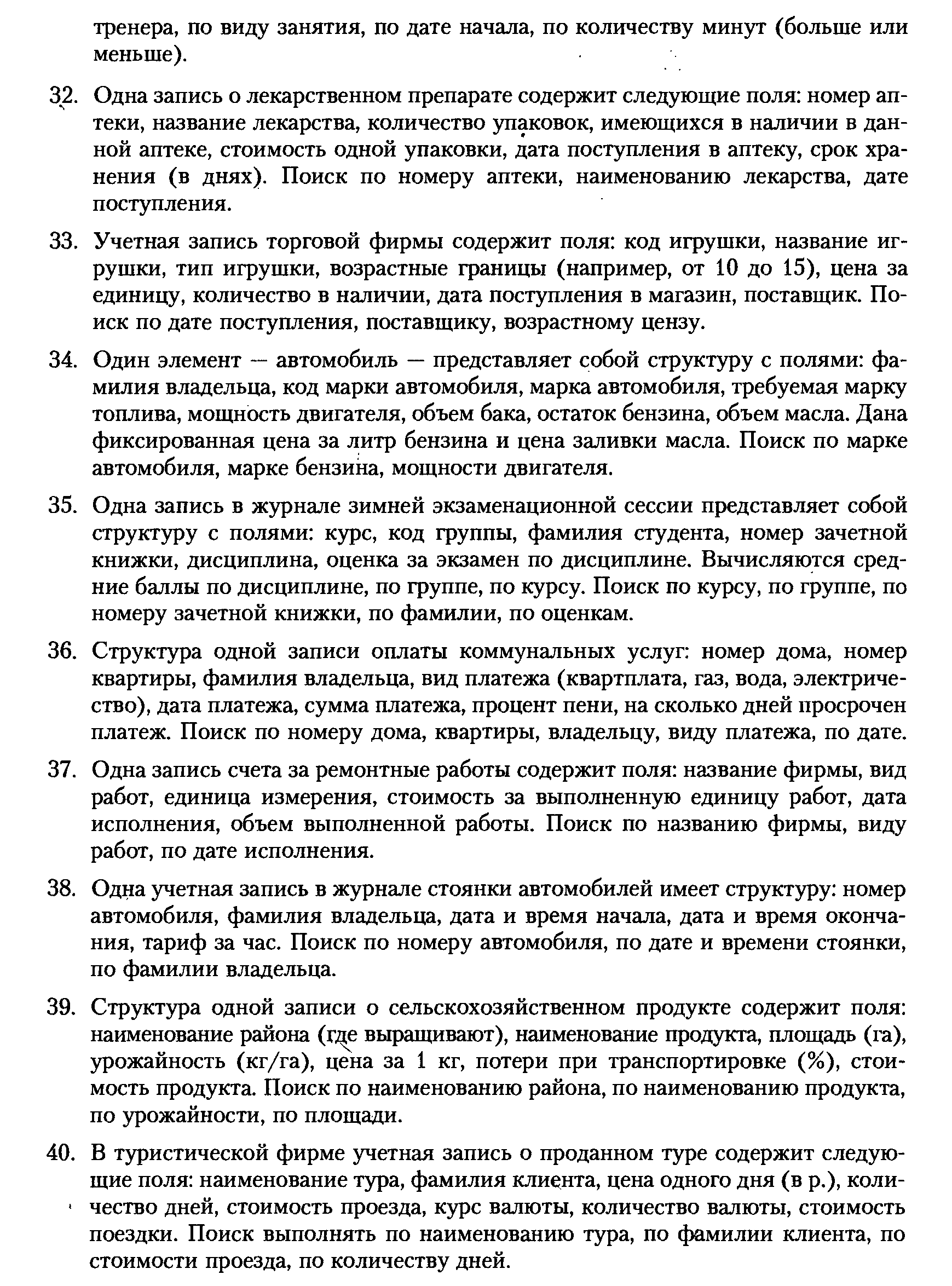
1. Функция outputResult() для вывода контейнера-результата. Предусмотреть следующие версии данной функции:
   1. Функция, выгружающая данные с использованием файловых потоков.
   2. Функция, использующая алгоритм copy для вывода контейнера в файл с использованием потоковых итераторов.

**Часть 2. Ассоциативные контейнеры**

Выполнить следующие задания, реализовав индексный массив в виде *ассоциативного контейнера-отображения*.







**Основные требования**

1. Реализация шаблонных функций не должна зависеть от конкретных типов данных, т.е. не допускается явно указывать конкретный тип (например, тип элементов контейнеров). Нужно использовать типы контейнеров (valuе\_type и др.).
2. Работа с файлами осуществляется на основе потоковых классов C++.