

Применение различных средств стандартной библиотеки C++

В каждом задании даются имена одного или нескольких текстовых файлов, содержащих исходные последовательности, а также имя текстового файла, в который требуется записать результаты обработки исходных последовательностей (имя результирующего файла указывается последним). Каждая исходная последовательность содержится в отдельном файле. Все исходные файлы содержат текст в кодировке «windows-1251»; эта же кодировка должна использоваться при записи полученных данных в результирующий файл.

Каждый элемент последовательности размещается в отдельной строке файла, в начале и конце строки пробелы отсутствуют, поля элемента не содержат пробелов и разделяются ровно одним пробелом. Все исходные числовые данные являются положительными. В качестве десятичного разделителя используется *точка*.

Если в задание входят дополнительные числовые или строковые исходные данные, то они указываются в начале набора исходных данных (перед именами файлов).

Во всех заданиях данной группы заготовки решения уже содержат определения исходных структур данных (в заданиях STL7Mix1–STL7Mix70 структура имеет имя `data`, в заданиях STL7Mix71–STL7Mix100 имя дополнительно содержит буквенный идентификатор соответствующего набора данных, например, `dataA` или `dataC`). Кроме того, заготовки содержат операторы, обеспечивающие чтение данных из исходных файлов в векторы типа `vector<data>`, где `data` — ранее определенная структура данных, и операторы, создающие результирующий файл.

Например, в задании STL7Mix1 заготовка задания включает определение структуры `data` с целочисленными полями `code`, `year`, `month`, `len`, а функция `Solve` содержит следующие операторы:

```
string name1, name2;
pt >> name1 >> name2;
ifstream f1(name1);
vector<data> V((istream_iterator<data>(f1)),
  istream_iterator<data>());
f1.close();
ShowLine(V.begin(), V.end(), "V: ");

ofstream f2(name2);

f2.close();
```

Оператор отладочной печати отображает в разделе отладки все введенные исходные данные. Выводить полученные данные следует в файловый поток `f2`.

Если компилятор поддерживает стандарт C++11, то для вывода результатов рекомендуется использовать цикл `for(auto& e : C)`, где `C` — контейнер с результирующими данными; в противном случае можно использовать алгоритм `for_each` для этого контейнера.

При решении большинства задач требуется выполнить *группировку* исходных данных, используя вспомогательное отображение подходящего типа. В задачах STL7Mix71–STL7Mix100 может оказаться удобным предварительно выполнить внутреннее или внешнее *объединение* данных по общему ключу. Группировка и объединение данных с использованием ассоциативных контейнеров рассматриваются в группе заданий STL5Assoc.

Если порядок вывода сгруппированных данных отличается от порядка следования их ключей, то следует скопировать полученные данные в последовательный контейнер (обычно век-

тор) и выполнить для него требуемую сортировку (используя алгоритмы `sort` или `stable_sort`).

Обработка отдельных последовательностей

STL7Mix1. Исходная последовательность содержит сведения о клиентах фитнес-центра. Каждый элемент последовательности включает следующие целочисленные поля:

```
<Код клиента> <Год> <Номер месяца>
<Продолжительность занятий (в часах)>
```

Найти элемент последовательности с минимальной продолжительностью занятий. Вывести эту продолжительность, а также соответствующие ей год и номер месяца (в указанном порядке на той же строке). Если имеется несколько элементов с минимальной продолжительностью, то вывести данные того из них, который является последним в исходной последовательности.

Указание. Используйте алгоритм `min_element` с предикатом.

STL7Mix2. Исходная последовательность содержит сведения о клиентах фитнес-центра. Каждый элемент последовательности включает следующие целочисленные поля:

```
<Номер месяца> <Год> <Код клиента>
<Продолжительность занятий (в часах)>
```

Найти элемент последовательности с максимальной продолжительностью занятий. Вывести эту продолжительность, а также соответствующие ей год и номер месяца (в указанном порядке на той же строке). Если имеется несколько элементов с максимальной продолжительностью, то вывести данные, соответствующие самой поздней дате.

Указание. Ср. с STL7Mix1. Для нахождения требуемого элемента используйте предикат, анализирующий поля «длительность», «год», «месяц».

STL7Mix3. Исходная последовательность содержит сведения о клиентах фитнес-центра. Каждый элемент последовательности включает следующие целочисленные поля:

```
<Год> <Номер месяца> <Продолжительность занятий
(в часах)> <Код клиента>
```

Определить год, в котором суммарная продолжительность занятий всех клиентов была наибольшей, и вывести этот год и наибольшую суммарную продолжительность. Если таких годов было несколько, то вывести наименьший из них.

Указание. Ср. с STL7Mix1. Для группировки по полю «год» используйте вспомогательное отображение.

STL7Mix4. Исходная последовательность содержит сведения о клиентах фитнес-центра. Каждый элемент последовательности включает следующие целочисленные поля:

```
<Год> <Номер месяца> <Продолжительность занятий
(в часах)> <Код клиента>
```

Для каждого клиента, присутствующего в исходных данных, определить суммарную продолжительность занятий в течение всех лет (вначале выводить суммарную продолжительность, затем код клиента). Сведения о каждом клиенте выводить на новой строке и упорядочивать по убыванию суммарной продолжительности, а при их равенстве — по возрастанию кода клиента.

Указание. Для группировки по полю «код» используйте вспомогательное отображение.

STL7Mix5. Исходная последовательность содержит сведения о клиентах фитнес-центра. Каждый элемент последовательности включает следующие целочисленные поля:

```
<Код клиента> <Продолжительность занятий (в часах)>
<Год> <Номер месяца>
```

Для каждого клиента, присутствующего в исходных данных, определить общее количество месяцев, в течение которых он

посещал занятия (вначале выводить количество месяцев, затем код клиента). Сведения о каждом клиенте выводить на новой строке и упорядочивать по возрастанию количества месяцев, а при их равенстве — по возрастанию кода клиента. Указание. Ср. с STL7Mix4.

STL7Mix6. Исходная последовательность содержит сведения о клиентах фитнес-центра. Каждый элемент последовательности включает следующие целочисленные поля:

```
<Код клиента> <Продолжительность занятий (в часах)>  
<Год> <Номер месяца>
```

Для каждого месяца определить суммарную продолжительность занятий всех клиентов за все годы (вначале выводить суммарную продолжительность, затем номер месяца). Если данные о некотором месяце отсутствуют, то для этого месяца вывести 0. Сведения о каждом месяце выводить на новой строке и упорядочивать по убыванию суммарной продолжительности, а при равной продолжительности — по возрастанию номера месяца.

Указание. Ср. с STL7Mix4. В данном случае вместо вспомогательного отображения можно использовать вектор с 12 элементами, соответствующими номерам месяцев (при этом элементами вектора должны быть пары «длительность, номер месяца»).

STL7Mix7. Дано целое число K — код одного из клиентов фитнес-центра. Исходная последовательность содержит сведения о клиентах этого фитнес-центра. Каждый элемент последовательности включает следующие целочисленные поля:

```
<Продолжительность занятий (в часах)> <Год>  
<Номер месяца> <Код клиента>
```

Для каждого года, в котором клиент с кодом K посещал центр, определить месяц, в котором продолжительность занятий данного клиента была наибольшей для данного года (если таких месяцев несколько, то выбирать месяц с наименьшим номером). Сведения о каждом годе выводить на новой строке в следующем порядке: год, номер месяца, продолжительность занятий в этом месяце. Упорядочивать сведения по убыванию номера года. Если данные о клиенте с кодом K отсутствуют, то записать в результирующий файл строку «Нет данных».

Указание. Для отбора данных, связанных с клиентом K , и их одновременной группировки по полю «год» используйте вспомогательное отображение M типа `map<int, data*, greater<int>>` (при этом функциональный объект `greater` обеспечит требуемую сортировку данных по убыванию номера года). Используя алгоритм `for_each` или цикл `for` по элементам контейнера, перебирайте элементы исходного набора данных, добавляя (или изменяя) элементы отображения M в соответствии с условиями задачи.

STL7Mix8. Дано целое число K — код одного из клиентов фитнес-центра. Исходная последовательность содержит сведения о клиентах этого фитнес-центра. Каждый элемент последовательности включает следующие целочисленные поля:

```
<Продолжительность занятий (в часах)> <Код клиента>  
<Год> <Номер месяца>
```

Для каждого года, в котором клиент с кодом K посещал центр, определить месяц, в котором продолжительность занятий данного клиента была наименьшей для данного года (если таких месяцев несколько, то выбирать первый из этих месяцев в исходном наборе; месяцы с нулевой продолжительностью занятий не учитывать). Сведения о каждом годе выводить на новой строке в следующем порядке: наименьшая продолжительность занятий, год, номер месяца. Упорядочивать сведения по возрастанию продолжительности занятий, а при равной продолжительности — по возрастанию

номера года. Если данные о клиенте с кодом K отсутствуют, то записать в результирующий файл строку «Нет данных».

Указание. Ср. с STL7Mix7. В данном случае после построения отображения M следует скопировать его во вспомогательный вектор и отсортировать этот вектор в требуемом порядке. Если в отображении M использовать порядок по умолчанию (т. е. по возрастанию ключа «год»), то для сортировки вспомогательного вектора достаточно использовать алгоритм устойчивой сортировки по возрастанию поля «длительность».

STL7Mix9. Дано целое число K — код одного из клиентов фитнес-центра. Исходная последовательность содержит сведения о клиентах этого фитнес-центра. Каждый элемент последовательности включает следующие целочисленные поля:

```
<Код клиента> <Продолжительность занятий (в часах)>  
<Номер месяца> <Год>
```

Для каждого года, в котором клиент с кодом K посещал центр, определить число месяцев, для которых продолжительность занятий данного клиента превосходила 15 часов (вначале выводить число месяцев, затем год). Если для некоторого года требуемые месяцы отсутствуют, то вывести для него 0. Сведения о каждом годе выводить на новой строке; данные упорядочивать по убыванию числа месяцев, а при равном числе месяцев — по возрастанию номера года. Если данные об указанном клиенте отсутствуют, то записать в результирующий файл строку «Нет данных».

Указание. Ср. с STL7Mix8. В данном случае достаточно использовать отображение M типа `map<int, int>`.

STL7Mix10. Исходная последовательность содержит сведения о клиентах фитнес-центра. Каждый элемент последовательности включает следующие целочисленные поля:

```
<Год> <Номер месяца> <Код клиента>  
<Продолжительность занятий (в часах)>
```

Для каждой пары «год–месяц», присутствующей в исходных данных, определить количество клиентов, которые посещали центр в указанное время (вначале выводится год, затем месяц, затем количество клиентов). Сведения о каждой паре «год–месяц» выводить на новой строке и упорядочивать по убыванию номера года, а для одинакового номера года — по возрастанию номера месяца.

Указание. Для группировки по составному ключу «год, номер месяца» используйте вспомогательное отображение M типа `map<pair<int, int>, int, greater_less>`, предварительно описав функциональный объект `greater_less` для реализации варианта сравнения пары ключей, описанного в задаче. Благодаря этому объекту полученное отображение будет содержать данные в требуемом порядке.

STL7Mix11. Исходная последовательность содержит сведения о клиентах фитнес-центра. Каждый элемент последовательности включает следующие целочисленные поля:

```
<Код клиента> <Год> <Номер месяца>  
<Продолжительность занятий (в часах)>
```

Для каждой пары «год–месяц», присутствующей в исходных данных, определить общую продолжительность занятий всех клиентов в указанное время (вначале выводится общая продолжительность, затем год, затем месяц). Сведения о каждой паре «год–месяц» выводить на новой строке и упорядочивать по возрастанию общей продолжительности занятий, для одинаковой продолжительности — по убыванию номера года, а для одинакового номера года — по возрастанию номера месяца.

Указание. Ср. с STL7Mix10. В данном случае дополнительно потребуется сортировка полученных данных по возрастанию общей продолжительности занятий. Необходимо использовать алгоритм устойчивой сортировки, что обеспечит

требуемый порядок пар с одинаковой общей продолжительностью (при условии использования отображения с функциональным объектом `greater_less`, описанным в указании к STL7Mix10).

STL7Mix12. Дано целое число P ($10 < P < 50$). Исходная последовательность содержит сведения о клиентах фитнес-центра. Каждый элемент последовательности включает следующие целочисленные поля:

<Продолжительность занятий (в часах)> <Код клиента>
<Номер месяца> <Год>

Для каждого года, присутствующего в исходных данных, определить количество месяцев, в которых суммарная длительность занятий всех клиентов составляла более P процентов от суммарной длительности за этот год (вначале выводить количество месяцев, затем год). Если в некотором году ни для одного месяца не выполнялось требуемое условие, то вывести для него 0. Сведения о каждом годе выводить на новой строке и упорядочивать по убыванию количества месяцев, а для одинакового количества — по возрастанию номера года.

Указание. Для группировки данных по полю «год» используйте отображение M типа `map<int, vector<int>>`, включая в вектор значений (длины 13) не только данные о каждом месяце (сохраняя их в элементах вектора с индексами от 1 до 12), но и суммарные данные по всему году (сохраняя их в элементе вектора с индексом 0). После формирования отображения M постройте на основе полученных данных вспомогательный вектор типа `vector<pair<int, int>>` и отсортируйте элементы вектора требуемым образом, используя алгоритм `stable_sort`.

STL7Mix13. Исходная последовательность содержит сведения об абитуриентах. Каждый элемент последовательности включает следующие поля:

<Номер школы> <Год поступления> <Фамилия>

Для каждого года, присутствующего в исходных данных, найти школу с наибольшим номером среди школ, которые окончили абитуриенты, поступившие в этом году, и вывести год и найденный номер школы. Сведения о каждом годе выводить на новой строке и упорядочивать по возрастанию номера года.

Указание. Используйте группировку по полю «год».

STL7Mix14. Исходная последовательность содержит сведения об абитуриентах. Каждый элемент последовательности включает следующие поля:

<Год поступления> <Номер школы> <Фамилия>

Определить, в какие годы общее число абитуриентов для всех школ было наибольшим, и вывести это число, а также количество таких лет. Каждое число выводить на новой строке.

Указание. Выполните группировку по полю «год», сохранив полученную последовательность во вспомогательном отображении. Используя алгоритм `max_element`, найдите наибольшее число абитуриентов, после чего воспользуйтесь алгоритмом `count_if`, чтобы определить количество лет, для которых число абитуриентов было максимальным.

STL7Mix15. Исходная последовательность содержит сведения об абитуриентах. Каждый элемент последовательности включает следующие поля:

<Номер школы> <Год поступления> <Фамилия>

Определить, в какие годы общее число абитуриентов для всех школ было наибольшим, и вывести это число, а также годы, в которые оно было достигнуто (годы упорядочивать по возрастанию, каждое число выводить на новой строке).

Указание. Ср. с STL7Mix14.

STL7Mix16. Исходная последовательность содержит сведения об абитуриентах. Каждый элемент последовательности включает следующие поля:

<Год поступления> <Номер школы> <Фамилия>

Для каждого года, присутствующего в исходных данных, вывести общее число абитуриентов, поступивших в этом году (вначале указывать число абитуриентов, затем год). Сведения о каждом годе выводить на новой строке и упорядочивать по убыванию числа поступивших, а для совпадающих чисел — по возрастанию номера года.

Указание. Ср. с STL7Mix14. Для упорядочивания полученных сгруппированных данных по убыванию числа поступивших используйте вспомогательный вектор и алгоритм `stable_sort`.

STL7Mix17. Исходная последовательность содержит сведения об абитуриентах. Каждый элемент последовательности включает следующие поля:

<Номер школы> <Год поступления> <Фамилия>

Для каждого года, присутствующего в исходных данных, вывести число различных школ, которые окончили абитуриенты, поступившие в этом году (вначале указывать число школ, затем год). Сведения о каждом годе выводить на новой строке и упорядочивать по возрастанию числа школ, а для совпадающих чисел — по возрастанию номера года.

Указание. В данном случае для группировки по полю «год» удобно использовать отображение `map<int, set<int>>`.

STL7Mix18. Исходная последовательность содержит сведения об абитуриентах. Каждый элемент последовательности включает следующие поля:

<Год поступления> <Номер школы> <Фамилия>

Найти годы, для которых число абитуриентов было не меньше среднего значения по всем годам (вначале указывать число абитуриентов для данного года, затем год). Сведения о каждом годе выводить на новой строке и упорядочивать по убыванию числа абитуриентов, а для совпадающих чисел — по возрастанию номера года.

Указание. Ср. с STL7Mix16.

STL7Mix19. Исходная последовательность содержит сведения об абитуриентах. Каждый элемент последовательности включает следующие поля:

<Фамилия> <Год поступления> <Номер школы>

Для каждой школы вывести общее число абитуриентов за все годы и фамилию первого из абитуриентов этой школы, содержащихся в исходном наборе данных (вначале указывать номер школы, затем число абитуриентов, затем фамилию). Сведения о каждой школе выводить на новой строке и упорядочивать по возрастанию номеров школ.

Указание. В данном случае для группировки по полю «номер школы» можно использовать отображение `map<int, pair<int, string>>`.

STL7Mix20. Исходная последовательность содержит сведения об абитуриентах. Каждый элемент последовательности включает следующие поля:

<Фамилия> <Номер школы> <Год поступления>

Определить, для каких школ общее число абитуриентов за все годы было наибольшим, и вывести данные об абитуриентах из этих школ (вначале указывать номер школы, затем фамилию абитуриента). Сведения о каждом абитуриенте выводить на новой строке и упорядочивать по возрастанию номеров школ, а для одинаковых номеров — в порядке следования абитуриентов в исходном наборе данных.

Указание. Ср. с STL7Mix14. После определения максимального количества абитуриентов с помощью первого вспомога-

тельного отображения (типа `map<int, int>`) сформировать второе отображение (типа `map<int, vector<string>>`), содержащее данные только о тех школах, количество абитуриентов для которых было максимальным.

STL7Mix21. Исходная последовательность содержит сведения об абитуриентах. Каждый элемент последовательности включает следующие поля:

<Фамилия> <Год поступления> <Номер школы>

Определить, для каких школ общее число абитуриентов за все годы было наибольшим, и вывести для каждой из этих школ данные о первом абитуриенте в алфавитном порядке (вначале указывать фамилию абитуриента, затем номер школы). Сведения о каждом абитуриенте выводить на новой строке и упорядочивать в алфавитном порядке, а для одинаковых фамилий — по возрастанию номеров школ.

Указание. Ср. с STL7Mix20. В данном случае можно использовать второе вспомогательное отображение типа `map<int, string>`, а также (для сортировки в нужном порядке) вспомогательный вектор типа `vector<pair<int, string>>`.

STL7Mix22. Исходная последовательность содержит сведения об абитуриентах. Каждый элемент последовательности включает следующие поля:

<Фамилия> <Номер школы> <Год поступления>

Для каждой школы найти годы поступления абитуриентов из этой школы и вывести номер школы и найденные для нее годы (годы располагаются на той же строке, что и номер школы, и упорядочиваются по возрастанию). Сведения о каждой школе выводить на новой строке и упорядочивать по возрастанию номеров школ.

Указание. Здесь, как и в задании STL7Mix17, удобно использовать вспомогательное отображение `map<int, set<int>>`, позволяющее сразу получить результирующие данные в требуемом порядке.

STL7Mix23. Исходная последовательность содержит сведения об абитуриентах. Каждый элемент последовательности включает следующие поля:

<Фамилия> <Год поступления> <Номер школы>

Для каждой пары «год–школа», присутствующей в исходных данных, найти число абитуриентов, относящихся к этому году и школе, и вывести год, номер школы и найденное число абитуриентов. Сведения о каждой паре «год–школа» выводить на новой строке и упорядочивать по убыванию года, а для совпадающих годов — по возрастанию номера школы.

Указание. Ср. с STL7Mix10.

STL7Mix24. Исходная последовательность содержит сведения об абитуриентах. Каждый элемент последовательности включает следующие поля:

<Фамилия> <Номер школы> <Год поступления>

Для каждой пары «школа–год», присутствующей в исходных данных, найти трех первых абитуриентов, относящихся к этой школе и году, и вывести номер школы, год и найденные фамилии (в порядке их следования в исходном наборе данных). Если для некоторой пары «школа–год» имеется менее трех абитуриентов, то вывести информацию обо всех абитуриентах, относящихся к этой паре. Сведения о каждой паре «школа–год» выводить на новой строке и упорядочивать по возрастанию номера школы, а для совпадающих номеров — по убыванию года.

Указание. Ср. с STL7Mix10. В данном случае вспомогательный функциональный объект целесообразно назвать `less_greater`, а в качестве типа вспомогательного отображения

использовать `map<pair<int, int>, vector<string>, less_greater>`.

STL7Mix25. Исходная последовательность содержит сведения о задолжниках по оплате коммунальных услуг, живущих в 144-квартирном 9-этажном доме. Каждый элемент последовательности включает следующие поля:

<Фамилия> <Задолженность> <Номер квартиры>

Задолженность указывается в виде дробного числа (целая часть — рубли, дробная часть — копейки). В каждом подъезде на каждом этаже располагаются по 4 квартиры. Найти номер подъезда, жильцы которого имеют наибольшую суммарную задолженность, и вывести этот номер вместе с размером суммарной задолженности (выводится с двумя дробными знаками). Считать, что суммарные задолженности для всех подъездов имеют различные значения.

Указание. Выполните группировку по номеру подъезда, используя следующую формулу нахождения номера подъезда `entrance` по номеру квартиры `num`: `entrance = (num - 1) / 36 + 1` (операция «/» обозначает деление нацело). К полученному в результате группировки отображению примените алгоритм `max_element`. Вещественные результаты выводите с двумя дробными знаками.

STL7Mix26. Исходная последовательность содержит сведения о задолжниках по оплате коммунальных услуг, живущих в 144-квартирном 9-этажном доме. Каждый элемент последовательности включает следующие поля:

<Номер квартиры> <Фамилия> <Задолженность>

Задолженность указывается в виде дробного числа (целая часть — рубли, дробная часть — копейки). В каждом подъезде на каждом этаже располагаются по 4 квартиры. Для каждого из 4 подъездов дома вывести сведения о задолжниках, живущих в этом подъезде: номер подъезда, число задолжников, средняя задолженность для жильцов этого подъезда (выводится с двумя дробными знаками). Жильцы, не имеющие долга, при вычислении средней задолженности не учитываются. Сведения о каждом подъезде выводить на отдельной строке и упорядочивать по возрастанию номера подъезда. Если в каком-либо подъезде задолжники отсутствуют, то данные об этом подъезде не выводить.

Указание. Ср. с STL7Mix25. При выполнении группировки используйте отображение типа `map<int, pair<int, double>>`.

STL7Mix27. Исходная последовательность содержит сведения о задолжниках по оплате коммунальных услуг, живущих в 144-квартирном 9-этажном доме. Каждый элемент последовательности включает следующие поля:

<Фамилия> <Номер квартиры> <Задолженность>

Задолженность указывается в виде дробного числа (целая часть — рубли, дробная часть — копейки). В каждом подъезде на каждом этаже располагаются по 4 квартиры. Для каждого из 9 этажей дома вывести сведения о задолжниках, живущих на этом этаже: число задолжников, номер этажа, суммарная задолженность для жильцов этого этажа (выводится с двумя дробными знаками). Сведения о каждом этаже выводить на отдельной строке и упорядочивать по возрастанию числа задолжников, а для совпадающих чисел — по возрастанию этажа. Если на каком-либо этаже задолжники отсутствуют, то данные об этом этаже не выводить.

Указание. Ср. с STL7Mix26. Используйте формулу нахождения номера этажа `floor` по номеру квартиры `num`: `floor = (num - 1) % 36 / 4 + 1` (операция «/» обозначает деление нацело, операция «%» — взятие остатка от деления нацело).

STL7Mix28. Исходная последовательность содержит сведения о задолжниках по оплате коммунальных услуг, живущих в

144-квартирном 9-этажном доме. Каждый элемент последовательности включает следующие поля:

<Задолженность> <Фамилия> <Номер квартиры>

Задолженность указывается в виде дробного числа (целая часть — рубли, дробная часть — копейки). В каждом подъезде на каждом этаже располагаются по 4 квартиры. Для каждого из 9 этажей дома вывести сведения о задолжниках, живущих на этом этаже: номер этажа, суммарная задолженность для жильцов этого этажа (выводится с двумя дробными знаками), число задолжников. Сведения о каждом этаже выводить на отдельной строке и упорядочивать по убыванию номера этажа. Если на каком-либо этаже задолжники отсутствуют, то вывести для этого этажа нулевые данные.

Указание. Ср. с STL7Mix27. В данном случае отображение, полученное в результате группировки по этажам, должно включать элементы, соответствующие всем этажам дома (при этом некоторые элементы могут содержать нулевые значения).

STL7Mix29. Исходная последовательность содержит сведения о задолжниках по оплате коммунальных услуг, живущих в 144-квартирном 9-этажном доме. Каждый элемент последовательности включает следующие поля:

<Номер квартиры> <Фамилия> <Задолженность>

Задолженность указывается в виде дробного числа (целая часть — рубли, дробная часть — копейки). В каждом подъезде на каждом этаже располагаются по 4 квартиры. Для каждого из 4 подъездов дома найти жильца с наибольшей задолженностью и вывести сведения о нем: номер подъезда, номер квартиры, фамилия жильца, задолженность (выводится с двумя дробными знаками). Считать, что в наборе исходных данных все задолженности имеют различные значения. Сведения о каждом задолжнике выводить на отдельной строке и упорядочивать по возрастанию номера подъезда. Если в каком-либо подъезде задолжники отсутствуют, то данные об этом подъезде не выводить.

Указание. Ср. с STL7Mix25. При группировке можно использовать отображение типа `map<int, vector<data*>>`, где `data` — тип элемента из исходного набора данных.

STL7Mix30. Исходная последовательность содержит сведения о задолжниках по оплате коммунальных услуг, живущих в 144-квартирном 9-этажном доме. Каждый элемент последовательности включает следующие поля:

<Фамилия> <Задолженность> <Номер квартиры>

Задолженность указывается в виде дробного числа (целая часть — рубли, дробная часть — копейки). В каждом подъезде на каждом этаже располагаются по 4 квартиры. Для каждого из 9 этажей дома найти жильца с наименьшей задолженностью и вывести сведения о нем: номер квартиры, номер этажа, фамилия жильца, задолженность (выводится с двумя дробными знаками). Считать, что в наборе исходных данных все задолженности имеют различные значения. Сведения о каждом задолжнике выводить на отдельной строке и упорядочивать по возрастанию номера квартиры. Если на каком-либо этаже задолжники отсутствуют, то данные об этом этаже не выводить.

Указание. Ср. с STL7Mix27 и STL7Mix29.

STL7Mix31. Исходная последовательность содержит сведения о задолжниках по оплате коммунальных услуг, живущих в 144-квартирном 9-этажном доме. Каждый элемент последовательности включает следующие поля:

<Задолженность> <Фамилия> <Номер квартиры>

Задолженность указывается в виде дробного числа (целая часть — рубли, дробная часть — копейки). В каждом подъезде на каждом этаже располагаются по 4 квартиры. Для ка-

ждого из 4 подъездов дома найти трех жильцов с наибольшей задолженностью и вывести сведения о них: задолженность (выводится с двумя дробными знаками), номер подъезда, номер квартиры, фамилия жильца. Считать, что в наборе исходных данных все задолженности имеют различные значения. Сведения о каждом задолжнике выводить на отдельной строке и упорядочивать по убыванию размера задолженности (номер подъезда при сортировке не учитывать). Если в каком-либо подъезде число задолжников меньше трех, то включить в полученный набор всех задолжников этого подъезда.

Указание. Ср. с STL7Mix29. Для отбора трех задолжников используйте алгоритм `partial_sort`.

STL7Mix32. Исходная последовательность содержит сведения о задолжниках по оплате коммунальных услуг, живущих в 144-квартирном 9-этажном доме. Каждый элемент последовательности включает следующие поля:

<Фамилия> <Номер квартиры> <Задолженность>

Задолженность указывается в виде дробного числа (целая часть — рубли, дробная часть — копейки). В каждом подъезде на каждом этаже располагаются по 4 квартиры. Для каждого из 9 этажей дома найти жильца с наименьшей задолженностью и вывести сведения о нем: номер этажа и задолженность (выводится с двумя дробными знаками). Считать, что в наборе исходных данных все задолженности имеют различные значения. Сведения о каждом этаже выводить на отдельной строке и упорядочивать по возрастанию номера этажа. Если на каком-либо этаже задолжники отсутствуют, то для этого этажа вывести задолженность, равную 0.00.

Указание. Ср. с STL7Mix28. В данном случае достаточно использовать вспомогательный массив вещественных чисел из 9 элементов (соответствующих всем этажам дома).

STL7Mix33. Исходная последовательность содержит сведения о задолжниках по оплате коммунальных услуг, живущих в 144-квартирном 9-этажном доме. Каждый элемент последовательности включает следующие поля:

<Номер квартиры> <Задолженность> <Фамилия>

Задолженность указывается в виде дробного числа (целая часть — рубли, дробная часть — копейки). В каждом подъезде на каждом этаже располагаются по 4 квартиры. Найти жильцов, долг которых не меньше величины средней задолженности по дому, и вывести сведения о них: номер квартиры, фамилия, задолженность (выводится с двумя дробными знаками). Жильцы, не имеющие долга, при вычислении средней задолженности не учитываются. Сведения о каждом задолжнике выводить на отдельной строке и упорядочивать по возрастанию номеров квартир.

Указание. Используйте алгоритм `remove_copy_if` для копирования требуемых элементов во вспомогательный массив. Для того чтобы при этом избежать многократного вычисления средней задолженности по дому, предварительно сохраните ее значение (найденное, например, с помощью алгоритма `accumulate`) во вспомогательной переменной.

STL7Mix34. Исходная последовательность содержит сведения о задолжниках по оплате коммунальных услуг, живущих в 144-квартирном 9-этажном доме. Каждый элемент последовательности включает следующие поля:

<Задолженность> <Номер квартиры> <Фамилия>

Задолженность указывается в виде дробного числа (целая часть — рубли, дробная часть — копейки). В каждом подъезде на каждом этаже располагаются по 4 квартиры. Найти жильцов, долг которых не больше величины средней задолженности по дому, и вывести сведения о них: номер этажа, номер квартиры, фамилия, задолженность (выводится с двумя дробными знаками). Жильцы, не имеющие долга, при вы-

числении средней задолженности не учитываются. Сведения о каждом задолжнике выводятся на отдельной строке и упорядочиваются по убыванию номеров этажей, а для одинаковых этажей — по возрастанию номеров квартир.

Указание. Ср. с STL7Mix33.

STL7Mix35. Исходная последовательность содержит сведения о задолжниках по оплате коммунальных услуг, живущих в 144-квартирном 9-этажном доме. Каждый элемент последовательности включает следующие поля:

<Задолженность> <Фамилия> <Номер квартиры>

Задолженность указывается в виде дробного числа (целая часть — рубли, дробная часть — копейки). В каждом подъезде на каждом этаже располагаются по 4 квартиры. Для каждого из 4 подъездов дома найти задолжников, долг которых не меньше величины средней задолженности по данному подъезду, и вывести сведения о них: номер подъезда, задолженность (выводится с двумя дробными знаками), фамилия, номер квартиры. Жильцы, не имеющие долга, при вычислении средней задолженности не учитываются. Сведения о каждом задолжнике выводятся на отдельной строке и упорядочиваются по возрастанию номеров подъездов, а для одинаковых подъездов — по убыванию размера задолженности. Считать, что в наборе исходных данных все задолженности имеют различные значения.

Указание. См. указания к STL7Mix25 и STL7Mix33. В данном случае для хранения средних значений удобно использовать вспомогательный вектор размера 4.

STL7Mix36. Исходная последовательность содержит сведения о задолжниках по оплате коммунальных услуг, живущих в 144-квартирном 9-этажном доме. Каждый элемент последовательности включает следующие поля:

<Номер квартиры> <Фамилия> <Задолженность>

Задолженность указывается в виде дробного числа (целая часть — рубли, дробная часть — копейки). В каждом подъезде на каждом этаже располагаются по 4 квартиры. Для каждого из 9 этажей дома найти задолжников, долг которых не больше величины средней задолженности по данному этажу, и вывести сведения о них: номер этажа, задолженность (выводится с двумя дробными знаками), фамилия, номер квартиры. Жильцы, не имеющие долга, при вычислении средней задолженности не учитываются. Сведения о каждом задолжнике выводятся на отдельной строке и упорядочиваются по возрастанию номеров этажей, а для одинаковых этажей — по возрастанию размера задолженности. Считать, что в наборе исходных данных все задолженности имеют различные значения.

Указание. Ср. с STL7Mix35; см. также указание к STL7Mix27.

STL7Mix37. Исходная последовательность содержит сведения об автозаправочных станциях (АЗС). Каждый элемент последовательности включает следующие поля:

<Компания> <Марка бензина> <Цена 1 литра
(в копейках)> <Улица>

Названия компаний и улиц не содержат пробелов. В качестве марки бензина указываются числа 92, 95 или 98. Каждая компания имеет не более одной АЗС на каждой улице; цены на разных АЗС одной и той же компании могут различаться. Для каждой марки бензина, присутствующей в исходных данных, определить минимальную и максимальную цену литра бензина этой марки (вначале выводить марку, затем цены в указанном порядке). Сведения о каждой марке выводятся на новой строке и упорядочиваются по убыванию значения марки.

STL7Mix38. Исходная последовательность содержит сведения об автозаправочных станциях (АЗС). Каждый элемент последовательности включает следующие поля:

<Цена 1 литра (в копейках)> <Марка бензина>
<Компания> <Улица>

Названия компаний и улиц не содержат пробелов. В качестве марки бензина указываются числа 92, 95 или 98. Каждая компания имеет не более одной АЗС на каждой улице; цены на разных АЗС одной и той же компании могут различаться. Для каждой марки бензина, присутствующей в исходных данных, определить количество станций, предлагавших эту марку (вначале выводить количество станций, затем номер марки). Сведения о каждой марке выводятся на новой строке и упорядочиваются по возрастанию количества станций, а для одинакового количества — по возрастанию значения марки.

STL7Mix39. Дано целое число M — значение одной из марок бензина. Исходная последовательность содержит сведения об автозаправочных станциях (АЗС). Каждый элемент последовательности включает следующие поля:

<Улица> <Компания> <Марка бензина> <Цена 1 литра
(в копейках)>

Названия компаний и улиц не содержат пробелов. В качестве марки бензина указываются числа 92, 95 или 98. Каждая компания имеет не более одной АЗС на каждой улице; цены на разных АЗС одной и той же компании могут различаться. Для каждой улицы определить количество АЗС, предлагавших марку бензина M (вначале выводить количество АЗС на данной улице, затем название улицы; количество АЗС может быть равно 0). Сведения о каждой улице выводятся на новой строке и упорядочиваются по возрастанию количества АЗС, а для одинакового количества — по названиям улиц в алфавитном порядке.

Указание. При выполнении группировки по полю «улица» используйте отображение типа `map<string, int>`.

STL7Mix40. Исходная последовательность содержит сведения об автозаправочных станциях (АЗС). Каждый элемент последовательности включает следующие поля:

<Компания> <Улица> <Марка бензина> <Цена 1 литра
(в копейках)>

Названия компаний и улиц не содержат пробелов. В качестве марки бензина указываются числа 92, 95 или 98. Каждая компания имеет не более одной АЗС на каждой улице; цены на разных АЗС одной и той же компании могут различаться. Для каждой улицы определить количество АЗС, предлагавших определенную марку бензина (вначале выводить название улицы, затем три числа — количество АЗС для бензина марки 92, 95 и 98; некоторые из этих чисел могут быть равны 0). Сведения о каждой улице выводятся на новой строке и упорядочиваются по названиям улиц в алфавитном порядке.

Указание. При выполнении группировки по полю «улица» используйте отображение типа `map<string, vector<int>>` (вектор должен иметь длину 3 и хранить информацию о количестве АЗС, предлагавших определенную марку бензина). Заметим, что индекс элемента вектора можно определить по значению марки m следующим образом: $(m - 92)/3$.

STL7Mix41. Дано целое число M — значение одной из марок бензина. Исходная последовательность содержит сведения об автозаправочных станциях (АЗС). Каждый элемент последовательности включает следующие поля:

<Марка бензина> <Улица> <Компания> <Цена 1 литра
(в копейках)>

Названия компаний и улиц не содержат пробелов. В качестве марки бензина указываются числа 92, 95 или 98. Каждая компания имеет не более одной АЗС на каждой улице; цены

на разных АЗС одной и той же компании могут различаться. Для каждой улицы, на которой имеются АЗС с бензином марки M , определить максимальную цену бензина этой марки (вначале выводить максимальную цену, затем название улицы). Сведения о каждой улице выводить на новой строке и упорядочивать по возрастанию максимальной цены, а для одинаковой цены — по названиям улиц в алфавитном порядке. Если ни одной АЗС с бензином марки M не найдено, то записать в результирующий файл строку «Нет».

STL7Mix42. Исходная последовательность содержит сведения об автозаправочных станциях (АЗС). Каждый элемент последовательности включает следующие поля:

<Марка бензина> <Компания> <Улица> <Цена 1 литра (в копейках)>

Названия компаний и улиц не содержат пробелов. В качестве марки бензина указываются числа 92, 95 или 98. Каждая компания имеет не более одной АЗС на каждой улице; цены на разных АЗС одной и той же компании могут различаться. Для каждой улицы определить минимальную цену бензина каждой марки (вначале выводить название улицы, затем три числа — минимальную цену для бензина марки 92, 95 и 98). При отсутствии бензина нужной марки выводить число 0. Сведения о каждой улице выводить на новой строке и упорядочивать по названиям улиц в алфавитном порядке.

Указание. Ср. с STL7Mix40.

STL7Mix43. Дано целое число M — значение одной из марок бензина. Исходная последовательность содержит сведения об автозаправочных станциях (АЗС). Каждый элемент последовательности включает следующие поля:

<Цена 1 литра (в копейках)> <Марка бензина> <Улица> <Компания>

Названия компаний и улиц не содержат пробелов. В качестве марки бензина указываются числа 92, 95 или 98. Каждая компания имеет не более одной АЗС на каждой улице; цены на разных АЗС одной и той же компании могут различаться. Для каждой компании определить разброс цен на бензин указанной марки M (вначале выводить разность максимальной и минимальной цены бензина марки M для АЗС данной компании, затем — название компании). Если бензин марки M не предлагался данной компанией, то разброс положить равным -1 . Сведения о каждой компании выводить на новой строке, данные упорядочивать по убыванию значений разброса, а для равных значений разброса — по названиям компаний в алфавитном порядке.

Указание. Ср. с STL7Mix41. Отличие от STL7Mix41 состоит в том, что в данном случае полученный набор должен содержать сведения обо всех компаниях (а не только тех, которые имеют АЗС, предлагающие бензин марки M). Можно, например, использовать для группировки по полю «компания» отображение типа `map<string, vector<int>>`, в котором для каждой компании хранятся все ее цены на бензин марки M , а затем на основе данного отображения сформировать вектор пар `pair<string, int>`, содержащий для каждой компании разброс цен, и отсортировать его в требуемом порядке. Для нахождения разброса цен можно использовать алгоритмы `max_element` и `min_element`.

STL7Mix44. Исходная последовательность содержит сведения об автозаправочных станциях (АЗС). Каждый элемент последовательности включает следующие поля:

<Марка бензина> <Цена 1 литра (в копейках)> <Компания> <Улица>

Названия компаний и улиц не содержат пробелов. В качестве марки бензина указываются числа 92, 95 или 98. Каждая компания имеет не более одной АЗС на каждой улице; цены на разных АЗС одной и той же компании могут различаться.

Для каждой компании определить разброс цен для всех марок бензина (вначале выводить название компании, затем три числа — разброс цен для бензина марки 92, 95 и 98). При отсутствии бензина нужной марки выводить число -1 . Сведения о каждой компании выводить на новой строке и упорядочивать по названиям компаний в алфавитном порядке.

Указание. Ср. с STL7Mix43. В данном случае при группировке по полю «компания» можно использовать отображение типа `map<string, multimap<int, int>>`, в котором для каждой компании хранятся все ее цены, сгруппированные по маркам бензина.

STL7Mix45. Исходная последовательность содержит сведения об автозаправочных станциях (АЗС). Каждый элемент последовательности включает следующие поля:

<Компания> <Цена 1 литра (в копейках)> <Марка бензина> <Улица>

Названия компаний и улиц не содержат пробелов. В качестве марки бензина указываются числа 92, 95 или 98. Каждая компания имеет не более одной АЗС на каждой улице; цены на разных АЗС одной и той же компании могут различаться. Для каждой улицы определить количество АЗС (вначале выводить название улицы, затем количество АЗС). Сведения о каждой улице выводить на новой строке и упорядочивать по названиям улиц в алфавитном порядке.

Указание. При группировке по полю «улица» используйте отображение типа `map<string, set<string>>`, в котором для каждой улицы хранится множество компаний, имеющих АЗС на этой улице.

STL7Mix46. Исходная последовательность содержит сведения об автозаправочных станциях (АЗС). Каждый элемент последовательности включает следующие поля:

<Улица> <Марка бензина> <Цена 1 литра (в копейках)> <Компания>

Названия компаний и улиц не содержат пробелов. В качестве марки бензина указываются числа 92, 95 или 98. Каждая компания имеет не более одной АЗС на каждой улице; цены на разных АЗС одной и той же компании могут различаться. Для каждой компании определить количество АЗС, предлагавших все три марки бензина (вначале выводить количество АЗС, затем — название компании; количество может быть равно 0). Сведения о каждой компании выводить на новой строке и упорядочивать по убыванию количества АЗС, а при равных количествах — по названиям компаний в алфавитном порядке.

Указание. Ср. с STL7Mix45. В данном случае при группировке по полю «компания» можно использовать отображение типа `map<string, map<string, int>>`, в котором с каждой компанией связывается вспомогательное отображение, содержащее в качестве ключей названия улиц, а в качестве значений — количество марок бензина, предлагавшихся на АЗС данной компании, расположенной на данной улице.

STL7Mix47. Исходная последовательность содержит сведения об автозаправочных станциях (АЗС). Каждый элемент последовательности включает следующие поля:

<Цена 1 литра (в копейках)> <Компания> <Улица> <Марка бензина>

Названия компаний и улиц не содержат пробелов. В качестве марки бензина указываются числа 92, 95 или 98. Каждая компания имеет не более одной АЗС на каждой улице; цены на разных АЗС одной и той же компании могут различаться. Вывести данные обо всех АЗС, предлагавших не менее двух марок бензина (вначале выводится название компании, затем название улицы, затем количество предлагавшихся марок бензина). Сведения о каждой АЗС выводить на новой строке и упорядочивать по названиям компаний в алфавитном по-

рядке, а для одинаковых компаний — по названиям улиц (также в алфавитном порядке). Если ни одной требуемой АЗС не найдено, то записать в результирующий файл строку «Нет».

Указание. Выполните группировку по ключу, полученному в результате сцепления полей «компания» и «улица» (с пробелом между ними).

STL7Mix48. Исходная последовательность содержит сведения об автозаправочных станциях (АЗС). Каждый элемент последовательности включает следующие поля:

<Цена 1 литра (в копейках)> <Улица> <Марка бензина>
<Компания>

Названия компаний и улиц не содержат пробелов. В качестве марки бензина указываются числа 92, 95 или 98. Каждая компания имеет не более одной АЗС на каждой улице; цены на разных АЗС одной и той же компании могут различаться. Перебрать все возможные комбинации улиц и компаний, содержащихся в исходном наборе данных, и для каждой пары «улица–компания» вывести название улицы, название компании и количество марок бензина, которое предлагает АЗС данной компании, расположенная на данной улице (если АЗС отсутствует, то количество полагается равным 0). Сведения о каждой паре выводить на новой строке и упорядочивать по названиям улиц в алфавитном порядке, а для одинаковых названий улиц — по названиям компаний (также в алфавитном порядке).

Указание. Предварительно сформируйте вспомогательное отображение M типа `map<string, int>`, содержащее в качестве ключей названия всех компаний, входящих в исходный набор данных (значения оставьте нулевыми). После этого выполните группировку по полю «улица», используя отображение типа `map<string, map<string, int>>`. При добавлении в отображение новой улицы сразу связывайте с ней построенное ранее вспомогательное отображение M и используйте его значения в качестве счетчика марок бензина, предлагавшихся на АЗС данной компании, расположенной на данной улице.

STL7Mix49. Исходная последовательность содержит сведения о результатах сдачи учащимися ЕГЭ по математике, русскому языку и информатике (в указанном порядке). Каждый элемент последовательности включает следующие поля:

<Фамилия> <Инициалы> <Номер школы> <Баллы ЕГЭ>

Баллы ЕГЭ представляют собой три целых числа в диапазоне от 0 до 100, которые отделяются друг от друга одним пробелом. Определить наименьший суммарный балл и вывести его. Вывести также сведения обо всех учащихся, получивших наименьший суммарный балл (для каждого учащегося указывать фамилию, инициалы и номер школы). Сведения о каждом учащемся выводить на отдельной строке и располагать в порядке их следования в исходном наборе.

Указание. Для нахождения минимального суммарного балла используйте алгоритм `min_element`.

STL7Mix50. Исходная последовательность содержит сведения о результатах сдачи учащимися ЕГЭ по математике, русскому языку и информатике (в указанном порядке). Каждый элемент последовательности включает следующие поля:

<Фамилия> <Инициалы> <Баллы ЕГЭ> <Номер школы>

Баллы ЕГЭ представляют собой три целых числа в диапазоне от 0 до 100, которые отделяются друг от друга одним пробелом. Определить два наибольших суммарных балла и вывести эти баллы на одной строке в порядке убывания (считать, что в исходных данных всегда присутствуют учащиеся с различными суммарными баллами). Также вывести сведения обо всех учащихся, получивших один из двух наибольших суммарных баллов (для каждого учащегося указывать фами-

лию, инициалы и суммарный балл). Сведения о каждом учащемся выводить на отдельной строке и располагать в порядке их следования в исходном наборе.

Указание. Ср. с STL7Mix49. Для нахождения двух наибольших суммарных баллов достаточно сформировать вспомогательное множество суммарных баллов типа `set<int, greater<int>>` и извлечь из него два первых элемента.

STL7Mix51. Исходная последовательность содержит сведения о результатах сдачи учащимися ЕГЭ по математике, русскому языку и информатике (в указанном порядке). Каждый элемент последовательности включает следующие поля:

<Баллы ЕГЭ> <Фамилия> <Инициалы> <Номер школы>

Баллы ЕГЭ представляют собой три целых числа в диапазоне от 0 до 100, которые отделяются друг от друга одним пробелом. Для каждой школы вывести сведения об учащемся, набравшем наибольший балл ЕГЭ по информатике среди учащихся этой школы. Если таких учащихся несколько, то вывести сведения о первом учащемся в порядке их следования в исходном наборе. Сведения о каждом учащемся выводить на отдельной строке, указывая номер школы, фамилию учащегося, его инициалы и балл ЕГЭ по информатике. Данные упорядочивать по возрастанию номера школы.

Указание. Используйте вспомогательное отображение типа `map<int, data*>`.

STL7Mix52. Исходная последовательность содержит сведения о результатах сдачи учащимися ЕГЭ по математике, русскому языку и информатике (в указанном порядке). Каждый элемент последовательности включает следующие поля:

<Номер школы> <Фамилия> <Инициалы> <Баллы ЕГЭ>

Баллы ЕГЭ представляют собой три целых числа в диапазоне от 0 до 100, которые отделяются друг от друга одним пробелом. Для каждой школы вывести сведения об учащемся, набравшем наименьший суммарный балл ЕГЭ среди учащихся этой школы. Если таких учащихся несколько, то вывести сведения о первом учащемся в алфавитном порядке их фамилий и инициалов. Сведения о каждом учащемся выводить на отдельной строке, указывая номер школы, суммарный балл ЕГЭ, фамилию учащегося и его инициалы. Данные упорядочивать по убыванию номера школы.

Указание. Ср. с STL7Mix51.

STL7Mix53. Исходная последовательность содержит сведения о результатах сдачи учащимися ЕГЭ по математике, русскому языку и информатике (в указанном порядке). Каждый элемент последовательности включает следующие поля:

<Фамилия> <Инициалы> <Номер школы> <Баллы ЕГЭ>

Баллы ЕГЭ представляют собой три целых числа в диапазоне от 0 до 100, которые отделяются друг от друга одним пробелом. Для каждой школы определить количество учащихся, суммарный балл которых превышает 150 баллов (вначале выводится количество учащихся, набравших в сумме более 150 баллов, затем номер школы; количество учащихся может быть равно 0). Сведения о каждой школе выводить на новой строке и упорядочивать по убыванию количества учащихся, а для одинакового количества — по возрастанию номера школы.

Указание. Используйте вспомогательное отображение типа `map<int, int>` для группировки исходных данных и вектор типа `vector<pair<int, int>>` совместно с алгоритмом `stable_sort` для упорядочивания полученных результатов.

STL7Mix54. Исходная последовательность содержит сведения о результатах сдачи учащимися ЕГЭ по математике, русскому языку и информатике (в указанном порядке). Каждый элемент последовательности включает следующие поля:

<Баллы ЕГЭ> <Номер школы> <Фамилия> <Инициалы>

Баллы ЕГЭ представляют собой три целых числа в диапазоне от 0 до 100, которые отделяются друг от друга одним пробелом. Для каждой школы найти среднее значение суммарного балла ЕГЭ, набранного учащимися этой школы (среднее значение является целым числом — результатом *деления нацело* суммы баллов всех учащихся на количество учащихся). Сведения о каждой школе выводить на отдельной строке, указывая средний суммарный балл ЕГЭ и номер школы. Данные упорядочивать по убыванию среднего балла, а при равных значениях среднего балла — по возрастанию номера школы.

Указание. Ср. с STL7Mix53. В данном случае в качестве типа отображения для группировки можно использовать `map<int, pair<int, int>>`.

STL7Mix55. Исходная последовательность содержит сведения о результатах сдачи учащимися ЕГЭ по математике, русскому языку и информатике (в указанном порядке). Каждый элемент последовательности включает следующие поля:

<Баллы ЕГЭ> <Фамилия> <Инициалы> <Номер школы>

Баллы ЕГЭ представляют собой три целых числа в диапазоне от 0 до 100, которые отделяются друг от друга одним пробелом. Вывести сведения об учащихся, набравших не менее 50 баллов по каждому предмету (вначале выводится фамилия и инициалы, затем номер школы и суммарный балл ЕГЭ по всем предметам). Сведения о каждом учащемся выводить на отдельной строке в алфавитном порядке фамилий и инициалов, а при их совпадении — в порядке следования учащихся в наборе исходных данных. Если ни один из учащихся не удовлетворяет указанным условиям, то записать в результирующий файл текст «Требуемые учащиеся не найдены».

Указание. Используйте алгоритм `remove_if`.

STL7Mix56. Исходная последовательность содержит сведения о результатах сдачи учащимися ЕГЭ по математике, русскому языку и информатике (в указанном порядке). Каждый элемент последовательности включает следующие поля:

<Фамилия> <Инициалы> <Баллы ЕГЭ> <Номер школы>

Баллы ЕГЭ представляют собой три целых числа в диапазоне от 0 до 100, которые отделяются друг от друга одним пробелом. Вывести сведения об учащихся, набравших более 90 баллов хотя бы по одному из предметов (вначале выводится фамилия и инициалы, затем номер школы). Сведения о каждом учащемся выводить на отдельной строке и располагать в алфавитном порядке фамилий и инициалов, а при их совпадении — по возрастанию номера школы. Если ни один из учащихся не удовлетворяет указанным условиям, то записать в результирующий файл текст «Требуемые учащиеся не найдены».

Указание. Ср. с STL7Mix55.

STL7Mix57. Исходная последовательность содержит сведения о результатах сдачи учащимися ЕГЭ по математике, русскому языку и информатике (в указанном порядке). Каждый элемент последовательности включает следующие поля:

<Номер школы> <Фамилия> <Инициалы> <Баллы ЕГЭ>

Баллы ЕГЭ представляют собой три целых числа в диапазоне от 0 до 100, которые отделяются друг от друга одним пробелом. Для каждой школы вывести фамилии и инициалы трех первых учащихся (в алфавитном порядке), набравших менее 50 баллов по каждому предмету (вначале выводится номер школы, затем фамилия и инициалы). Сведения о каждом учащемся выводить на отдельной строке и упорядочивать по возрастанию номера школы, а для совпадающих номеров — в алфавитном порядке фамилий и инициалов. Если для некоторой школы имеется менее трех учащихся, удовлетворяющих указанным условиям, то вывести сведения обо всех таких учащихся. Если в исходном наборе нет ни одного учащегося, удовлетворяющего указанным условиям, то записать в

результирующий файл текст «Требуемые учащиеся не найдены».

Указание. Используйте вспомогательное отображение типа `map<int, multiset<string>>` для группировки исходных данных.

STL7Mix58. Исходная последовательность содержит сведения о результатах сдачи учащимися ЕГЭ по математике, русскому языку и информатике (в указанном порядке). Каждый элемент последовательности включает следующие поля:

<Фамилия> <Инициалы> <Номер школы> <Баллы ЕГЭ>

Баллы ЕГЭ представляют собой три целых числа в диапазоне от 0 до 100, которые отделяются друг от друга одним пробелом. Для каждой школы найти трех первых учащихся (в алфавитном порядке), набравших менее 50 баллов хотя бы по одному из предметов, и вывести их фамилию, инициалы и номер школы. Сведения о каждом учащемся выводить на отдельной строке и упорядочивать в алфавитном порядке фамилий и инициалов, а при их совпадении — по возрастанию номера школы. Если для некоторой школы имеется менее трех учащихся, удовлетворяющих указанным условиям, то вывести сведения обо всех таких учащихся. Если в исходном наборе нет ни одного учащегося, удовлетворяющего указанным условиям, то записать в результирующий файл текст «Требуемые учащиеся не найдены».

Указание. Ср. с STL7Mix57.

STL7Mix59. Исходная последовательность содержит сведения о результатах сдачи учащимися ЕГЭ по математике, русскому языку и информатике (в указанном порядке). Каждый элемент последовательности включает следующие поля:

<Номер школы> <Баллы ЕГЭ> <Фамилия> <Инициалы>

Баллы ЕГЭ представляют собой три целых числа в диапазоне от 0 до 100, которые отделяются друг от друга одним пробелом. Для каждой школы и каждого предмета определить количество учащихся, набравших не менее 50 баллов по этому предмету (вначале выводится номер школы, затем три числа — количество учащихся этой школы, набравших требуемое число баллов по математике, русскому языку и информатике; некоторые из чисел могут быть равны 0). Сведения о каждой школе выводить на новой строке и упорядочивать по возрастанию номера школы.

Указание. Для группировки исходных данных используйте вспомогательное отображение типа `map<int, vector<int>>`.

STL7Mix60. Исходная последовательность содержит сведения о результатах сдачи учащимися ЕГЭ по математике, русскому языку и информатике (в указанном порядке). Каждый элемент последовательности включает следующие поля:

<Фамилия> <Инициалы> <Баллы ЕГЭ> <Номер школы>

Баллы ЕГЭ представляют собой три целых числа в диапазоне от 0 до 100, которые отделяются друг от друга одним пробелом. Для каждой школы и каждого предмета найти среднее значение балла ЕГЭ, набранного учащимися этой школы (среднее значение является целым числом — результатом *деления нацело* суммы баллов всех учащихся на количество учащихся). Сведения о каждой школе выводить на отдельной строке, указывая номер школы и средние баллы по математике, русскому языку и информатике. Данные упорядочивать по убыванию номера школы.

Указание. Ср. с STL7Mix59.

STL7Mix61. Исходная последовательность содержит сведения об оценках учащихся по трем предметам: алгебре, геометрии и информатике. Каждый элемент последовательности содержит данные об одной оценке и включает следующие поля:

<Фамилия> <Инициалы> <Класс> <Название предмета>
<Оценка>

Полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс задается целым числом, оценка — целое число в диапазоне 2–5. Название предмета указывается с заглавной буквы. Для каждого учащегося определить среднюю оценку по каждому предмету и вывести ее с двумя дробными знаками (если по какому-либо предмету учащийся не получил ни одной оценки, то вывести для этого предмета 0.00). Сведения о каждом учащемся выводить на отдельной строке, указывая фамилию, инициалы и средние оценки по алгебре, геометрии и информатике. Данные располагать в алфавитном порядке фамилий и инициалов.

Указание. Для группировки исходных данных используйте вспомогательное отображение типа `map<string, map<string, pair<double, int>>>`. В паре `pair<double, int>` накапливайте суммарную оценку (в первом элементе) и количество оценок (во втором элементе); после обработки всех исходных данных достаточно поделить первый элемент на второй и сохранить полученное среднее значение в первом элементе пары (если второй элемент равен 0, то деление выполнять не следует).

STL7Mix62. Исходная последовательность содержит сведения об оценках учащихся по трем предметам: алгебре, геометрии и информатике. Каждый элемент последовательности содержит данные об одной оценке и включает следующие поля:

```
<Класс> <Фамилия> <Инициалы> <Оценка> <Название предмета>
```

Полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс задается целым числом, оценка — целое число в диапазоне 2–5. Название предмета указывается с заглавной буквы. Для каждого учащегося определить количество оценок по каждому предмету (если по какому-либо предмету учащийся не получил ни одной оценки, то вывести для этого предмета число 0). Сведения о каждом учащемся выводить на отдельной строке, указывая класс, фамилию, инициалы и количество оценок по алгебре, геометрии и информатике. Данные располагать в порядке возрастания номера класса, а для одинаковых классов — в алфавитном порядке фамилий и инициалов.

Указание. Для группировки исходных данных используйте вспомогательное отображение типа `map<int, map<string, map<string, int>>>`.

STL7Mix63. Исходная последовательность содержит сведения об оценках учащихся по трем предметам: алгебре, геометрии и информатике. Каждый элемент последовательности содержит данные об одной оценке и включает следующие поля:

```
<Название предмета> <Фамилия> <Инициалы> <Класс> <Оценка>
```

Полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс задается целым числом, оценка — целое число в диапазоне 2–5. Название предмета указывается с заглавной буквы. Вывести сведения об учащихся, имеющих по алгебре среднюю оценку не более 4: фамилию, инициалы, номер класса и среднюю оценку по алгебре (выводится с двумя дробными знаками). Для учащихся, не имеющих ни одной оценки по алгебре, считать среднюю оценку равной 0.00. Сведения о каждом учащемся выводить на отдельной строке и располагать в алфавитном порядке их фамилий и инициалов. Если ни один из учащихся не удовлетворяет указанным условиям, то записать в результирующий файл текст «Требуемые учащиеся не найдены».

Указание. Для группировки исходных данных используйте вспомогательное отображение типа `map<pair<string, int>, pair<double, int>>`. По поводу пары `pair<double, int>` см. указание к STL7Mix61.

STL7Mix64. Исходная последовательность содержит сведения об оценках учащихся по трем предметам: алгебре, геометрии и информатике. Каждый элемент последовательности содержит данные об одной оценке и включает следующие поля:

```
<Класс> <Фамилия> <Инициалы> <Название предмета> <Оценка>
```

Полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс задается целым числом, оценка — целое число в диапазоне 2–5. Название предмета указывается с заглавной буквы. Вывести сведения об учащихся, имеющих по информатике среднюю оценку не менее 4: номер класса, фамилию, инициалы и среднюю оценку по информатике (выводится с двумя дробными знаками). Сведения о каждом учащемся выводить на отдельной строке и располагать в порядке возрастания классов, а для одинаковых классов — в алфавитном порядке фамилий и инициалов. Если ни один из учащихся не удовлетворяет указанным условиям, то записать в результирующий файл текст «Требуемые учащиеся не найдены».

Указание. Ср. с STL7Mix63. В данном случае для группировки исходных данных можно использовать вспомогательное отображение типа `map<pair<int, string>, pair<double, int>>`.

STL7Mix65. Дана строка S — название одного из трех предметов: алгебры, геометрии или информатики. Исходная последовательность содержит сведения об оценках учащихся по этим трем предметам. Каждый элемент последовательности содержит данные об одной оценке и включает следующие поля:

```
<Фамилия> <Инициалы> <Название предмета> <Оценка> <Класс>
```

Полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс задается целым числом, оценка — целое число в диапазоне 2–5. Название предмета указывается с заглавной буквы. Для каждого класса, присутствующего в наборе исходных данных, определить число учащихся, имеющих по предмету S среднюю оценку не более 3.5 или не имеющих ни одной оценки по этому предмету. Сведения о каждом классе выводить на отдельной строке, указывая число найденных учащихся (число может быть равно 0) и номер класса. Данные упорядочивать по возрастанию числа учащихся, а для совпадающих чисел — по убыванию номера класса.

Указание. Для группировки исходных данных используйте вспомогательное отображение типа `map<int, map<string, pair<double, int>>, greater<int>>`, после чего постройте на его основе вектор типа `vector<pair<int, int>>` и отсортируйте его требуемым образом. По поводу пары `pair<double, int>` см. указание к STL7Mix61.

STL7Mix66. Дана строка S — название одного из трех предметов: алгебры, геометрии или информатики. Исходная последовательность содержит сведения об оценках учащихся по этим трем предметам. Каждый элемент последовательности содержит данные об одной оценке и включает следующие поля:

```
<Название предмета> <Фамилия> <Инициалы> <Оценка> <Класс>
```

Полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс задается целым числом, оценка — целое число в диапазоне 2–5. Название предмета указывается с заглавной буквы. Для каждого класса, присутствующего в наборе исходных данных, определить число учащихся, имеющих по предмету S среднюю оценку не менее 3.5 и при этом не получивших ни одной двойки по этому

предмету. Сведения о каждом классе выводить на отдельной строке, указывая номер класса и число найденных учащихся (число может быть равно 0). Данные упорядочивать по возрастанию номера класса.

Указание. Ср. с STL7Mix65. Если вспомогательное отображение описать как `map<int, map<string, pair<double, int>>>`, то для построенного на его основе вспомогательного вектора не потребуется выполнять дополнительной сортировки.

STL7Mix67. Исходная последовательность содержит сведения об оценках учащихся по трем предметам: алгебре, геометрии и информатике. Каждый элемент последовательности содержит данные об одной оценке и включает следующие поля:

```
<Класс> <Название предмета> <Фамилия> <Инициалы>
<Оценка>
```

Полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс задается целым числом, оценка — целое число в диапазоне 2–5. Название предмета указывается с заглавной буквы. Найти всех *двоечников* — учащихся, получивших хотя бы одну двойку по какому-либо предмету. Вывести сведения о каждом из двоечников: номер класса, фамилию, инициалы и полученное число двоек. Сведения о каждом двоечнике выводить на отдельной строке и располагать по убыванию классов, а для одинаковых классов — в алфавитном порядке фамилий и инициалов. Если в наборе исходных данных нет ни одной двойки, то записать в результирующий файл текст «Требуемые учащиеся не найдены».

Указание. Организуйте сохранение данных о двоечниках во вспомогательном отображении типа `map<pair<int, string>, int, greater_less>`, предварительно описав функциональный объект `greater_less` для реализации варианта сравнения пары ключей, описанного в задаче (номера классов должны упорядочиваться по убыванию, а фамилии в пределах класса — в алфавитном порядке).

STL7Mix68. Исходная последовательность содержит сведения об оценках учащихся по трем предметам: алгебре, геометрии и информатике. Каждый элемент последовательности содержит данные об одной оценке и включает следующие поля:

```
<Класс> <Оценка> <Фамилия> <Инициалы> <Название
предмета>
```

Полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс задается целым числом, оценка — целое число в диапазоне 2–5. Название предмета указывается с заглавной буквы. Найти всех *хорошистов* — учащихся, не получивших ни одной двойки и тройки, но имеющих хотя бы одну четверку по какому-либо предмету. Вывести сведения о каждом хорошисте: полученное число четверок, фамилию, инициалы и номер класса. Сведения о каждом учащемся выводить на отдельной строке и располагать по возрастанию количества четверок, а при их равенстве — в алфавитном порядке фамилий и инициалов. Если в наборе исходных данных нет ни одного учащегося, удовлетворяющего указанному условию, то записать в результирующий файл текст «Требуемые учащиеся не найдены».

Указание. Организуйте сохранение данных о двоечниках во вспомогательном отображении типа `map<pair<string, int>, pair<int, int>>`, сохраняя в первом компоненте пары ключей фамилию, а во втором — номер класса, и накапливая в первом компоненте пары значений количество четверок, полученных учеником, а во втором — количество двоек и троек. После завершения обработки исходных данных останется преобразовать полученное отображение в вектор типа `vector<pair<pair<string, int>, pair<int, int>>>`, удалить в нем сведения об учениках, имеющих ненулевое

число двоек и троек, и отсортировать его алгоритмом `stable_sort` по возрастанию числа полученных четверок.

STL7Mix69. Исходная последовательность содержит сведения об оценках учащихся по трем предметам: алгебре, геометрии и информатике. Каждый элемент последовательности содержит данные об одной оценке и включает следующие поля:

```
<Класс> <Фамилия> <Инициалы> <Название предмета>
<Оценка>
```

Полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс задается целым числом, оценка — целое число в диапазоне 2–5. Название предмета указывается с заглавной буквы. Для каждого класса найти *злых двоечников* — учащихся, получивших в данном классе максимальное суммарное число двоек по всем предметам (число не должно быть нулевым). Вывести сведения о каждом из злых двоечников: фамилию, инициалы, номер класса и полученное число двоек. Сведения о каждом двоечнике выводить на отдельной строке и располагать в алфавитном порядке их фамилий и инициалов (сортировку по классам не проводить). Если в наборе исходных данных нет ни одной двойки, то записать в результирующий файл текст «Требуемые учащиеся не найдены».

Указание. Для группировки данных используйте отображение типа `map<int, map<string, int>>` (данные группируются по классам, в каждый класс добавляются только двоечники, причем для каждого двоечника сохраняется количество его двоек). Затем, используя алгоритм `max_element` для каждого класса, постройте отображение, в котором для каждого класса хранится максимальное число двоек, полученных его учениками, и с помощью этого отображения постройте итоговое отображение, содержащее данные обо всех учащихся, получивших в каждом классе максимальное число двоек (если ключами в этом отображении будут фамилии, то данные будут автоматически отсортированы требуемым в задаче способом).

STL7Mix70. Исходная последовательность содержит сведения об оценках учащихся по трем предметам: алгебре, геометрии и информатике. Каждый элемент последовательности содержит данные об одной оценке и включает следующие поля:

```
<Оценка> <Класс> <Фамилия> <Инициалы> <Название
предмета>
```

Полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс задается целым числом, оценка — целое число в диапазоне 2–5. Название предмета указывается с заглавной буквы. Для каждого класса найти *лучших учеников* — учащихся, получивших в данном классе максимальное суммарное число пятерок по всем предметам (число не должно быть нулевым). При поиске лучших учеников (в частности, при определении максимального суммарного числа пятерок) не следует учитывать учащихся, получивших хотя бы одну двойку или тройку. Вывести сведения о каждом из лучших учеников: номер класса, фамилию, инициалы и полученное число пятерок. Сведения о каждом учащемся выводить на отдельной строке и располагать по возрастанию классов, а для одинаковых классов — в алфавитном порядке фамилий и инициалов. Если в наборе исходных данных нет ни одного учащегося, удовлетворяющего указанным условиям, то записать в результирующий файл текст «Требуемые учащиеся не найдены».

Указание. Ср. с STL7Mix69. В данном случае в ходе начальной группировки необходимо сохранять не только количество пятерок, но также количество полученных двоек и троек. Затем надо удалить учащихся, получивших двойки и тройки, после чего действовать как при решении STL7Mix69 с учетом того, что сортировать итоговые данные требуется в другом порядке.

Обработка нескольких взаимосвязанных последовательностей

В каждом задании данной подгруппы требуется обработать несколько (от двух до четырех) последовательностей из следующего набора:

A: сведения о потребителях, содержащие поля «Код потребителя», «Год рождения», «Улица проживания»;

B: сведения о товарах, содержащие поля «Артикул товара», «Категория», «Страна-производитель»;

C: скидки для потребителей в различных магазинах, содержащие поля «Код потребителя», «Название магазина», «Скидка (в процентах)»;

D: цены товаров в различных магазинах, содержащие поля «Артикул товара», «Название магазина», «Цена (в рублях)»;

E: сведения о покупках потребителей в различных магазинах, содержащие поля «Код потребителя», «Артикул товара», «Название магазина».

Порядок следования полей для элементов каждой последовательности определяется в формулировке задания.

В последовательности *A* все элементы имеют различные значения поля «Код потребителя». В последовательности *B* все элементы имеют различные значения поля «Артикул товара». В последовательности *C* все элементы имеют различные комбинации полей «Код потребителя» и «Название магазина». В последовательности *D* все элементы имеют различные комбинации полей «Артикул товара» и «Название магазина». Последовательность *E* может содержать одинаковые элементы (это соответствует ситуации, при которой один и тот же потребитель приобрел в одном и том же магазине несколько одинаковых товаров).

Все значения кодов потребителей и артикулов товаров, присутствующие в последовательностях *C*, *D* и *E*, обязательно содержатся в последовательностях *A* и *B*. Некоторые значения кодов потребителей и артикулов товаров, присутствующие в последовательностях *A* и *B*, могут отсутствовать в остальных последовательностях. Любая комбинация «магазин–товар», присутствующая в последовательности *E*, обязательно присутствует и в последовательности *D*. Комбинация «потребитель–магазин», присутствующая в последовательности *E*, может отсутствовать в последовательности *C*; это означает, что при покупке указанного товара в данном магазине потребитель не имел скидки (т. е. скидка была равна 0).

Коды потребителей, годы рождения, скидки и цены задаются целыми числами; значения скидок лежат в диапазоне от 5 до 50. Прочие поля являются строковыми и не содержат пробелов. Артикулы товаров имеют формат «AAddd-dddd», где на позициях, помеченных символом «A», располагается какая-либо заглавная латинская буква, а на позициях, помеченных символом «d», — какая-либо цифра.

Если потребитель приобрел товар, имеющий цену p , со скидкой d процентов, то размер скидки на данный товар должен вычисляться по формуле $p \cdot d / 100$, где символ «/» обозначает операцию целочисленного деления (иными словами, при вычислении размера скидки копейки отбрасываются).

STL7Mix71. Даны последовательности *A* и *C*, включающие следующие поля:

A: <Код потребителя> <Год рождения> <Улица проживания>

C: <Код потребителя> <Скидка (в процентах)> <Название магазина>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждого магазина определить потребителей, имеющих максимальную скидку в этом магазине (вначале выводится название магазина, затем код потребителя, его год рождения и значение максимальной скидки). Если для некоторого магазина имеется несколько потре-

бителей с максимальной скидкой, то вывести данные о потребителе с минимальным кодом. Сведения о каждом магазине выводить на новой строке и упорядочивать по названиям магазинов в алфавитном порядке.

STL7Mix72. Даны последовательности *A* и *C*, включающие следующие поля:

A: <Код потребителя> <Улица проживания> <Год рождения>

C: <Скидка (в процентах)> <Код потребителя> <Название магазина>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждого потребителя, указанного в *A*, определить количество магазинов, в которых ему предоставляется скидка (вначале выводится количество магазинов, затем код потребителя, затем его улица проживания). Если у некоторого потребителя нет скидки ни в одном магазине, то для него выводится количество магазинов, равное 0. Сведения о каждом потребителе выводить на новой строке и упорядочивать по возрастанию количества магазинов, а при равном количестве — по возрастанию кодов потребителей.

STL7Mix73. Даны последовательности *A* и *C*, включающие следующие поля:

A: <Год рождения> <Код потребителя> <Улица проживания>

C: <Код потребителя> <Название магазина> <Скидка (в процентах)>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждого магазина и каждой улицы определить количество потребителей, живущих на этой улице и имеющих скидку в этом магазине (вначале выводится название магазина, затем название улицы, затем количество потребителей со скидкой). Если для некоторой пары «магазин–улица» потребители со скидкой не найдены, то данные об этой паре не выводятся. Сведения о каждой паре «магазин–улица» выводить на новой строке и упорядочивать по названиям магазинов в алфавитном порядке, а для одинаковых названий магазинов — по названиям улиц (также в алфавитном порядке).

STL7Mix74. Даны последовательности *A* и *C*, включающие следующие поля:

A: <Улица проживания> <Код потребителя> <Год рождения>

C: <Название магазина> <Скидка (в процентах)> <Код потребителя>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждого магазина и каждого года рождения из *A* определить среднюю скидку в данном магазине (в процентах) для потребителей с этим годом рождения (вначале выводится название магазина, затем год рождения, затем значение средней скидки в процентах). Дробная часть найденного среднего значения отбрасывается, скидка выводится как целое число. При вычислении средней скидки учитываются только потребители данного года рождения, имеющие скидку в данном магазине. Если таких потребителей для данного магазина нет, то для этой пары «магазин–год» в качестве средней скидки выводится 0. Сведения о каждой паре «магазин–год» выводить на новой строке и упорядочивать по названиям магазинов в алфавитном порядке, а для одинаковых названий магазинов — по возрастанию номеров года.

STL7Mix75. Даны последовательности *B* и *D*, включающие следующие поля:

B: <Артикул товара> <Категория> <Страна-производитель>

D: <Название магазина> <Артикул товара> <Цена (в рублях)>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждого магазина и каждой категории товаров определить количество различных артикулов товаров данной категории, имеющихся в данном магазине (вначале выводится название магазина, затем категория, затем количество различных артикулов товаров). Если для некоторого магазина товары данной категории отсутствуют, то информация о соответствующей паре «магазин–категория» не выводится. Сведения о каждой паре «магазин–категория» выводить на новой строке и упорядочивать по названиям магазинов в алфавитном порядке, а для одинаковых названий магазинов — по названиям категорий (также в алфавитном порядке).

STL7Mix76. Даны последовательности *B* и *D*, включающие следующие поля:

B: <Страна-производитель> <Категория> <Артикул товара>

D: <Артикул товара> <Название магазина> <Цена (в рублях)>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждой страны-производителя определить количество магазинов, предлагающих товары, произведенные в этой стране, а также минимальную цену для товаров из данной страны по всем магазинам (вначале выводится количество магазинов, затем страна, затем минимальная цена). Если для некоторой страны не найдено ни одного товара, представленного в каком-либо магазине, то количество магазинов и минимальная цена полагаются равными 0. Сведения о каждой стране выводить на новой строке и упорядочивать по возрастанию количества магазинов, а в случае одинакового количества — по названиям стран в алфавитном порядке.

STL7Mix77. Даны последовательности *B* и *D*, включающие следующие поля:

B: <Категория> <Артикул товара> <Страна-производитель>

D: <Артикул товара> <Цена (в рублях)> <Название магазина>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждой категории товаров определить количество магазинов, предлагающих товары данной категории, а также количество стран, в которых произведены товары данной категории, представленные в магазинах (вначале выводится количество магазинов, затем название категории, затем количество стран). Если для некоторой категории не найдено ни одного товара, представленного в каком-либо магазине, то информация о данной категории не выводится. Сведения о каждой категории выводить на новой строке и упорядочивать по убыванию количества магазинов, а в случае одинакового количества — по названиям категорий в алфавитном порядке.

STL7Mix78. Даны последовательности *D* и *E*, включающие следующие поля:

D: <Цена (в рублях)> <Артикул товара> <Название магазина>

E: <Код потребителя> <Название магазина> <Артикул товара>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждого товара определить количество покупок данного товара и максимальную цену покупки (вначале выводится количество покупок, затем артикул товара, затем максимальная цена покупки). Если некото-

рый товар ни разу не был продан, то информация об этом товаре не выводится. Сведения о каждом товаре выводить на новой строке и упорядочивать по возрастанию количества покупок, а в случае одинакового количества — по артикулам товаров в алфавитном порядке.

STL7Mix79. Даны последовательности *D* и *E*, включающие следующие поля:

D: <Название магазина> <Цена (в рублях)> <Артикул товара>

E: <Код потребителя> <Артикул товара> <Название магазина>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждого магазина и каждого потребителя определить суммарную стоимость покупок, сделанных этим потребителем в данном магазине (вначале выводится название магазина, затем код потребителя, затем суммарная стоимость покупок). Если потребитель не приобрел ни одного товара в некотором магазине, то информация о соответствующей паре «магазин–потребитель» не выводится. Сведения о каждой паре «магазин–потребитель» выводить на новой строке и упорядочивать по названиям магазинов в алфавитном порядке, а в случае одинаковых названий — по возрастанию кодов потребителей.

STL7Mix80. Даны последовательности *D* и *E*, включающие следующие поля:

D: <Цена (в рублях)> <Название магазина> <Артикул товара>

E: <Артикул товара> <Название магазина> <Код потребителя>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждой пары «магазин–товар», указанной в *D*, определить суммарную стоимость продаж этого товара в данном магазине (вначале выводится название магазина, затем артикул товара, затем суммарная стоимость его продаж). Если товар ни разу не был продан в некотором магазине, то для соответствующей пары «магазин–товар» в качестве суммарной стоимости выводится 0. Сведения о каждой паре «магазин–товар» выводить на новой строке и упорядочивать по названиям магазинов в алфавитном порядке, а в случае одинаковых названий — по артикулам товаров (также в алфавитном порядке).

STL7Mix81. Даны последовательности *B*, *D* и *E*, включающие следующие поля:

B: <Артикул товара> <Страна-производитель> <Категория>

D: <Название магазина> <Артикул товара> <Цена (в рублях)>

E: <Название магазина> <Код потребителя> <Артикул товара>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждой страны-производителя определить количество приобретенных товаров из данной страны и суммарную стоимость приобретенных товаров (вначале выводится название страны, затем количество товаров, затем суммарная стоимость). Если сведения о проданных товарах для некоторой страны-производителя отсутствуют, то информация об этой стране не выводится. Сведения о каждой стране выводить на новой строке и упорядочивать по названиям стран в алфавитном порядке.

STL7Mix82. Даны последовательности *B*, *D* и *E*, включающие следующие поля:

B: <Категория> <Страна-производитель> <Артикул товара>

D: <Цена (в рублях)> <Артикул товара> <Название магазина>

E: <Артикул товара> <Код потребителя> <Название магазина>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждого потребителя определить количество категорий приобретенных им товаров и максимальную цену одной его покупки (вначале выводится количество категорий товаров, затем код потребителя, затем максимальная цена покупки). Сведения о каждом потребителе выводить на новой строке и упорядочивать по убыванию количества категорий, а при совпадении количества категорий — по возрастанию кодов потребителей.

STL7Mix83. Даны последовательности *B*, *D* и *E*, включающие следующие поля:

B: <Страна-производитель> <Артикул товара>
<Категория>

D: <Цена (в рублях)> <Название магазина> <Артикул товара>

E: <Название магазина> <Артикул товара> <Код потребителя>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждого магазина, указанного в *E*, и каждой страны-производителя определить суммарную стоимость товаров из данной страны, проданных в данном магазине (вначале выводится название магазина, затем название страны, затем суммарная стоимость). Если для некоторой пары «магазин–страна» отсутствует информация о проданных товарах, то в качестве суммарной стоимости выводится 0. Сведения о каждой паре «магазин–страна» выводить на новой строке и упорядочивать по названиям магазинов в алфавитном порядке, а для одинаковых названий магазинов — по названиям стран (также в алфавитном порядке).

STL7Mix84. Даны последовательности *C*, *D* и *E*, включающие следующие поля:

C: <Скидка (в процентах)> <Название магазина> <Код потребителя>

D: <Артикул товара> <Название магазина> <Цена (в рублях)>

E: <Артикул товара> <Название магазина> <Код потребителя>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждого магазина и каждого товара определить количество покупок этого товара со скидкой в данном магазине и суммарную стоимость этих покупок с учетом скидки (вначале выводится название магазина, затем артикул товара, затем количество покупок со скидкой и их суммарная стоимость). При вычислении размера скидки на товар копейки отбрасываются. Если для некоторой пары «магазин–товар» не найдено ни одной покупки со скидкой, то информация о данной паре не выводится. Если не найдено ни одной подходящей пары «магазин–товар», то записать в результирующий файл текст «Требуемые данные не найдены». Сведения о каждой паре «магазин–товар» выводить на новой строке и упорядочивать по названиям магазинов в алфавитном порядке, а для одинаковых названий — по артикулам товаров (также в алфавитном порядке).

STL7Mix85. Даны последовательности *C*, *D* и *E*, включающие следующие поля:

C: <Название магазина> <Код потребителя> <Скидка (в процентах)>

D: <Артикул товара> <Цена (в рублях)> <Название магазина>

E: <Артикул товара> <Код потребителя> <Название магазина>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждой пары «потребитель–магазин», указанной в *E*, определить суммарный размер скидок на все товары, приобретенные этим потребителем в данном магазине (вначале выводится код потребителя, затем название магазина, затем суммарный размер скидки). При вычислении размера скидки на каждый приобретенный товар копейки отбрасываются. Если потребитель приобретал товары в некотором магазине без скидки, то информация о соответствующей паре «потребитель–магазин» не выводится. Если не найдено ни одной подходящей пары «потребитель–магазин», то записать в результирующий файл текст «Требуемые данные не найдены». Сведения о каждой паре «потребитель–магазин» выводить на новой строке и упорядочивать по возрастанию кодов потребителей, а для одинаковых кодов — по названиям магазинов в алфавитном порядке.

STL7Mix86. Даны последовательности *C*, *D* и *E*, включающие следующие поля:

C: <Скидка (в процентах)> <Код потребителя>
<Название магазина>

D: <Название магазина> <Цена (в рублях)> <Артикул товара>

E: <Код потребителя> <Название магазина> <Артикул товара>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждой пары «товар–магазин», указанной в *E*, определить максимальный размер скидки на этот товар при его приобретении в данном магазине (вначале выводится артикул товара, затем название магазина, затем максимальный размер скидки). При вычислении размера скидки на товар копейки отбрасываются. Если все продажи товара в некотором магазине проводились без скидки, то в качестве максимального размера скидки для данной пары выводится 0. Сведения о каждой паре «товар–магазин» выводить на новой строке и упорядочивать по артикулам товаров в алфавитном порядке, а для одинаковых артикулов — по названиям магазинов (также в алфавитном порядке).

STL7Mix87. Даны последовательности *A*, *D* и *E*, включающие следующие поля:

A: <Год рождения> <Улица проживания> <Код потребителя>

D: <Артикул товара> <Название магазина> <Цена (в рублях)>

E: <Название магазина> <Код потребителя> <Артикул товара>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждой улицы проживания и каждого магазина, указанного в *E*, определить суммарную стоимость покупок в данном магазине, сделанных всеми потребителями, живущими на данной улице (вначале выводится название улицы, затем название магазина, затем суммарная стоимость покупок). Если для некоторой пары «улица–магазин» отсутствует информация о проданных товарах, то в качестве суммарной стоимости выводится 0. Сведения о каждой паре «улица–магазин» выводить на новой строке и упорядочивать по названиям улиц в алфавитном порядке, а для одинаковых названий улиц — по названиям магазинов (также в алфавитном порядке).

STL7Mix88. Даны последовательности *A*, *D* и *E*, включающие следующие поля:

A: <Улица проживания> <Год рождения> <Код потребителя>

D: <Артикул товара> <Цена (в рублях)> <Название магазина>

E: <Название магазина> <Артикул товара> <Код потребителя>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждого года рождения, указанного в *A*, и каждого товара, указанного в *E*, определить суммарную стоимость покупок данного товара, сделанных всеми потребителями с данным годом рождения (вначале выводится год рождения, затем артикул товара, затем суммарная стоимость покупок). Если для некоторой пары «год–товар» отсутствуют сведения о проданных товарах, то информация об этой паре не выводится. Сведения о каждой паре «год–товар» выводить на новой строке и упорядочивать по убыванию номеров года, а для одинаковых номеров — по артикулам товаров в алфавитном порядке.

STL7Mix89. Даны последовательности *A*, *D* и *E*, включающие следующие поля:

A: <Код потребителя> <Год рождения> <Улица проживания>

D: <Название магазина> <Цена (в рублях)> <Артикул товара>

E: <Код потребителя> <Артикул товара> <Название магазина>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждого магазина, указанного в *E*, определить потребителя с наименьшим годом рождения, купившего один или более товаров в данном магазине (вначале выводится название магазина, затем код потребителя, год его рождения и суммарная стоимость товаров, приобретенных потребителем в данном магазине). Если для некоторого магазина имеется несколько покупателей с наименьшим годом рождения, то выводятся данные обо всех таких покупателях. Сведения о каждой паре «магазин–потребитель» выводить на новой строке и упорядочивать по названиям магазинов в алфавитном порядке, а для одинаковых названий магазинов — по возрастанию кодов потребителей.

STL7Mix90. Даны последовательности *A*, *B* и *E*, включающие следующие поля:

A: <Улица проживания> <Код потребителя> <Год рождения>

B: <Страна-производитель> <Категория> <Артикул товара>

E: <Артикул товара> <Код потребителя> <Название магазина>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждого года рождения из *A* определить страну, в которой было произведено максимальное количество товаров, приобретенных потребителями этого года рождения (вначале выводится год, затем название страны, затем максимальное количество покупок). Если для некоторой пары «год–страна» отсутствует информация о проданных товарах, то эта пара не обрабатывается (в частности, если потребители некоторого года рождения не сделали ни одной покупки, то информация об этом годе не выводится). Если для какого-либо года рождения имеется несколько стран с наибольшим числом приобретенных товаров, то выводятся данные о первой из таких стран (в алфавитном порядке). Сведения о каждом годе выводить на новой строке и упорядочивать по убыванию номера года.

STL7Mix91. Даны последовательности *A*, *B* и *E*, включающие следующие поля:

A: <Улица проживания> <Год рождения> <Код потребителя>

B: <Артикул товара> <Страна-производитель> <Категория>

E: <Название магазина> <Артикул товара> <Код потребителя>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждой улицы и каждой категории товаров определить количество стран, в которых были произведены товары данной категории, купленные потребителями, живущими на этой улице (вначале выводится название улицы, затем название категории, затем количество стран). Если для какой-либо категории отсутствует информация о товарах, проданных жителям некоторой улицы, то информация о соответствующей паре «улица–категория» не выводится. Сведения о каждой паре «улица–категория» выводить на новой строке и упорядочивать по названиям улиц в алфавитном порядке, а для одинаковых улиц — по названиям категорий (также в алфавитном порядке).

STL7Mix92. Даны последовательности *A*, *B* и *E*, включающие следующие поля:

A: <Год рождения> <Улица проживания> <Код потребителя>

B: <Страна-производитель> <Артикул товара> <Категория>

E: <Артикул товара> <Название магазина> <Код потребителя>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждой категории товаров определить улицу с максимальным суммарным количеством товаров данной категории, приобретенных жителями этой улицы (вначале выводится название категории, затем название улицы, затем максимальное суммарное количество покупок). Если для какой-либо категории отсутствует информация о товарах, проданных жителям некоторой улицы, то суммарное количество покупок для соответствующей пары «категория–улица» считается равным 0 (при этом возможна ситуация, когда наибольшее количество покупок для какой-либо категории равно 0). Если для некоторой категории имеется несколько улиц с наибольшим количеством покупок, то выводятся данные обо всех таких улицах. Сведения о каждой паре «категория–улица» выводить на новой строке и упорядочивать по названиям категорий в алфавитном порядке, а для одинаковых категорий — по названиям улиц (также в алфавитном порядке).

STL7Mix93. Даны последовательности *A*, *B*, *C* и *E*, включающие следующие поля:

A: <Код потребителя> <Улица проживания> <Год рождения>

B: <Категория> <Страна-производитель> <Артикул товара>

C: <Название магазина> <Код потребителя> <Скидка (в процентах)>

E: <Код потребителя> <Артикул товара> <Название магазина>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждой страны-производителя и улицы проживания определить максимальный размер скидки (в процентах) для изделий, произведенных в данной стране и приобретенных потребителями, живущими на данной улице (вначале выводится название страны, затем название улицы, затем максимальный размер скидки). Если для некоторой пары «страна–улица» все товары были приобретены без скидки, то в качестве максимального размера скидки выводится 0. Если для некоторой пары «страна–улица» отсутствует информация о приобретенных товарах, то информация о дан-

ной паре не выводится. Сведения о каждой паре «страна–улица» выводить на новой строке и упорядочивать по названиям стран в алфавитном порядке, а для одинаковых названий — по названиям улиц (также в алфавитном порядке).

STL7Mix94. Даны последовательности *A*, *B*, *C* и *E*, включающие следующие поля:

A: <Год рождения> <Код потребителя> <Улица проживания>

B: <Артикул товара> <Категория> <Страна-производитель>

C: <Код потребителя> <Название магазина> <Скидка (в процентах)>

E: <Название магазина> <Код потребителя> <Артикул товара>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждого магазина, указанного в *E*, и каждой категории товаров определить минимальное значение года рождения среди тех потребителей, которые приобрели в данном магазине один или более товаров данной категории, и количество товаров данной категории, приобретенных со скидкой в данном магазине потребителями этого года рождения (вначале выводится название магазина, затем название категории, затем номер минимального года рождения и количество товаров, приобретенных со скидкой). Если для некоторой пары «магазин–категория» информация о проданных товарах отсутствует, то данные об этой паре не выводятся. Если для некоторой пары «магазин–категория» покупатель с минимальным годом рождения приобрел все товары без скидки, то в качестве значения количества товаров, приобретенных со скидкой, выводится 0. Сведения о каждой паре «магазин–категория» выводить на новой строке и упорядочивать по названиям магазинов в алфавитном порядке, а для одинаковых названий магазинов — по названиям категорий (также в алфавитном порядке).

STL7Mix95. Даны последовательности *A*, *C*, *D* и *E*, включающие следующие поля:

A: <Код потребителя> <Улица проживания> <Год рождения>

C: <Название магазина> <Скидка (в процентах)> <Код потребителя>

D: <Название магазина> <Артикул товара> <Цена (в рублях)>

E: <Код потребителя> <Название магазина> <Артикул товара>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждого артикула товара, указанного в *E*, и каждой улицы проживания определить суммарный размер скидки на изделия данного артикула, приобретенные потребителями, живущими на данной улице (вначале выводится артикул товара, затем название улицы, затем суммарный размер скидки). При вычислении размера скидки на товар копейки отбрасываются. Если на проданный товар скидка отсутствует, то ее размер полагается равным 0. Если для некоторой пары «товар–улица» отсутствует информация о приобретенных товарах, то данные об этой паре не выводятся. Если для некоторой пары «товар–улица» все изделия были приобретены без скидок, то в качестве суммарной скидки для этой пары выводится 0. Сведения о каждой паре «товар–улица» выводить на новой строке и упорядочивать по артикулам товаров в алфавитном порядке, а для одинаковых артикулов — по названиям улиц (также в алфавитном порядке).

STL7Mix96. Даны последовательности *A*, *C*, *D* и *E*, включающие следующие поля:

A: <Улица проживания> <Год рождения> <Код потребителя>

C: <Код потребителя> <Скидка (в процентах)> <Название магазина>

D: <Цена (в рублях)> <Артикул товара> <Название магазина>

E: <Название магазина> <Артикул товара> <Код потребителя>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждого года рождения, указанного в *A*, и каждого магазина, указанного в *E*, определить суммарную стоимость всех товаров (с учетом скидки), проданных в данном магазине потребителям данного года рождения (вначале выводится номер года, затем название магазина, затем суммарная стоимость проданных товаров с учетом скидки). При вычислении размера скидки на товар копейки отбрасываются. Если на проданный товар скидка отсутствует, то ее размер полагается равным 0. Если для некоторой пары «год–магазин» отсутствует информация о проданных товарах, то данные об этой паре не выводятся. Сведения о каждой паре «год–магазин» выводить на новой строке и упорядочивать по возрастанию номеров года, а для одинаковых номеров — по названиям магазинов в алфавитном порядке.

STL7Mix97. Даны последовательности *B*, *C*, *D* и *E*, включающие следующие поля:

B: <Категория> <Артикул товара> <Страна-производитель>

C: <Скидка (в процентах)> <Название магазина> <Код потребителя>

D: <Цена (в рублях)> <Название магазина> <Артикул товара>

E: <Название магазина> <Код потребителя> <Артикул товара>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждой страны-производителя и каждого потребителя определить суммарную стоимость (с учетом скидки) всех товаров, произведенных в данной стране и приобретенных этим потребителем (вначале выводится название страны, затем код потребителя, затем суммарная стоимость с учетом скидки). При вычислении размера скидки на товар копейки отбрасываются. Если на проданный товар скидка отсутствует, то ее размер полагается равным 0. Если для некоторой пары «страна–потребитель» данные о покупках отсутствуют, то информация об этой паре не выводится. Сведения о каждой паре «страна–потребитель» выводить на новой строке и упорядочивать по названиям стран в алфавитном порядке, а для одинаковых названий стран — по возрастанию кодов потребителей.

STL7Mix98. Даны последовательности *B*, *C*, *D* и *E*, включающие следующие поля:

B: <Страна-производитель> <Артикул товара> <Категория>

C: <Скидка (в процентах)> <Код потребителя> <Название магазина>

D: <Артикул товара> <Цена (в рублях)> <Название магазина>

E: <Код потребителя> <Артикул товара> <Название магазина>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждой категории товаров и каждого магазина, указанного в *E*, определить суммарный размер скидки на все товары данной категории, проданные в данном магазине (вначале выводится название категории, за-

тем название магазина, затем суммарная скидка). При вычислении размера скидки, затем суммарная скидка). При вычислении размера скидки на товар копейки отбрасываются. Если на проданный товар скидка отсутствует, то ее размер полагается равным 0. Если для некоторой категории товаров в каком-либо магазине не было продаж, то суммарная скидка для этой пары «категория–магазин» полагается равной –1. Сведения о каждой паре «категория–магазин» выводить на новой строке и упорядочивать по названиям категорий в алфавитном порядке, а для одинаковых названий категорий — по названиям магазинов (также в алфавитном порядке).

STL7Mix99. Даны последовательности *A*, *B*, *D* и *E*, включающие следующие поля:

A: <Код потребителя> <Улица проживания> <Год рождения>

B: <Категория> <Страна-производитель> <Артикул товара>

D: <Артикул товара> <Название магазина> <Цена (в рублях)>

E: <Артикул товара> <Название магазина> <Код потребителя>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждой категории товаров и каждой улицы проживания определить магазин, продавший товар данной категории по минимальной цене потребителю, живущему на данной улице (вначале выводится название категории, затем название улицы, затем название магазина и минимальная цена товара). Если для некоторой пары «категория–улица» отсутствует информация о проданных товарах, то данные об этой паре не выводятся. Если для некоторой пары «категория–улица» имеется несколько магазинов, продавших товар по минимальной цене, то выводятся данные обо всех таких магазинах. Сведения о каждой тройке «категория–улица–магазин» выводить на новой строке и упорядочивать по названиям категорий в алфавитном порядке, для одинаковых названий категорий — по названиям улиц, а для одинаковых названий улиц — по названиям магазинов (также в алфавитном порядке).

STL7Mix100. Даны последовательности *A*, *B*, *D* и *E*, включающие следующие поля:

A: <Улица проживания> <Код потребителя> <Год рождения>

B: <Артикул товара> <Страна-производитель> <Категория>

D: <Название магазина> <Цена (в рублях)> <Артикул товара>

E: <Артикул товара> <Код потребителя> <Название магазина>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждой страны-производителя и каждого магазина определить потребителя с наибольшим годом рождения, купившего в данном магазине один или более товаров, произведенных в данной стране (вначале выводится название страны, затем название магазина, затем год рождения потребителя, его код, а также суммарная стоимость товаров из данной страны, купленных в данном магазине). Если для некоторой пары «страна–магазин» отсутствует информация о проданных товарах, то данные об этой паре не выводятся. Если для некоторой пары «страна–магазин» имеется несколько потребителей с наибольшим годом рождения, то выводятся данные обо всех таких потребителях. Сведения о каждой тройке «страна–магазин–потребитель» выводить на новой строке и упорядочивать по названиям стран в алфавитном порядке, для одинаковых названий стран — по названиям магазинов (также в алфавитном порядке), а для одинаковых магазинов — по возрастанию кодов потребителей.