Лабораторная работа №5

В рамках данной контрольной работы необходимо решить предложенные задачи на языке программирования высокого уровня из предложенного перечня. Варианты задач находятся в отдельном файле. В качестве результата выполнения практической работы необходимо приложить **архив с файлами решенных задач, а также отчет о выполнении работы**. Файлы решенных задач представляют собой файлы с исходным кодом с установленным расширением.

Отчет о выполнении практической работы должен содержать:

1. Титульный лист

2. Содержание

3. 3 параграфа, в которых раскрыто решение каждой задачи. По каждой задаче необходимо представить следующую информацию:

3.1. *Условие задачи*. Берется из файла.

3.2. *Ход решения задачи*. Студент описывает логику решения данной задачи. Какие алгоритмы и для чего использованы, как построена программа. Данная часть является описательной. Здесь следует говорить именно о построении алгоритма, опуская процессы ввода и вывода данных (если это не является основной сутью алгоритма).

3.3. *Листинг программы с комментариями*. Копируется весь программный код.

3.4. *Тестирование программы*. Составляется таблица, содержащая следующие поля: номер теста, входные данные, результат выполнения программы, корректность, время выполнения (мс), затраченная память (Мб). Составляется не менее 10-ти тестовых наборов данных согласно условию задачи. Тестовые наборы входных данных студент составляет самостоятельно. В обязательном порядке программа тестируется на граничных наборах входных данных (например, если N варьируется от 0 до 10^9, то обязательно рассмотреть решение задачи при N=0 и при N близком к 10^9). Если написанная программа не позволяет решить задачу при граничных входных данных, все равно включить в тест и в качестве результата написать "Не решено". В столбце "входные данные" данные впечатываются вручную, в столбце "результат..." представляется скриншот выполнения программы (если не влезает на одну страницу, делать несколько скриншотов).

Задачи по теме 5. Обобщенный быстрый поиск и хеш-функции

Задача 1. Подстроки

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

Входной файл состоит из одной строки I, содержащей малые буквы английского алфавита.

Назовем подстроковой длиной L с началом S множество непрерывно следующих символов строки.

Например, строка

содержит подстроки:

длины 1: a, b, c, a, b

длины 2: ab, bc, ca, ab

длины 3: abc, bca, cab

длины 4: abca, bcab

длины 5: abcab

В строках длины 1 есть два повторяющихся элемента — a и b. Назовем весом подстрок длины L произведение максимального количества повторяющихся подстрок этой длины на длину L.

В нашем случае вес длины 1 есть 2 (2*1), длины 2 есть 4 (2*2), длины 3 — 3 (1*3), длины 4 — 4 и длины 5 — 5.

Требуется найти наибольший из всех весов различных длин.

Примеры

Стандартный ввод	Стандартный вывод

Замечание

Длина входной строки превышает 10000 символов

Задача 2. Большая книжка

Ограничение по времени: 5 секунды

Ограничение по памяти: 4 мегабайта

Заказчику понравилось решение нашей задачи по созданию записной книжки и он предложил нам более сложную задачу: создать простую базу данных, которая хранит много записей вида ключ: значение. Для работы с книжкой предусмотрены 4 команды:

ADD KEY VALUE — добавить в базу запись с ключом KEY и значением VALUE. Если такая запись уже есть, вывести ERROR.

DDELETE KEY — удалить из базы данных запись с ключом KEY. Если такой записи нет — вывести ERROR.

UPDATE KEY VALUE — заменить в записи с ключом KEY значение на VALUE. Если такой записи нет — вывести ERROR.

PRINT KEY — вывести ключ записи и значение через пробел. Если такой записи нет — вывести ERROR.

Количество входных строк в файле с данными не превышает 300000, количество первоначальных записей равно половине количества строк (первые N/2 команд есть команды ADD).

Длины ключей и данных не превосходят 4096. Ключи и данные содержат только буквы латинского алфавита и цифры и не содержат пробелов.

Особенность задачи: все данные не поместятся в оперативной памяти и поэтому придется использовать внешнюю.

Формат входных данных

См. В примерах.

Формат выходных данных

См. В примерах.

Примеры

Стандартный ввод	Стандартный вывод
ADD ADX LVT ADD LKFLG UWM PRINT ADX UPDATE HNTP JQPVG PRINT QURWB	ADX LVT ERROR QURWB MEGW ERROR
15 ADD RWJSN JFTF ADD ZDH GOON ADD FCDS TCAY ADD FCDS TCAY ADD HMGVI BWK ADD JTDU TLWWN ADD IXRJ ERF ADD IAOD GRDO PRINT IXRJ PRINT JTDU PRINT IXRJ UPDATE ZDH IOX PRINT ZDH ADD GVWU RTA DELETE ZDH ADD FCDS IVFJV	IXRJ ERF JTDU TLWWN IXRJ ERF

Задача 3. Сопоставление по образцу

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

Известно, что при работе с файлами можно указывать метасимволы * и ? для отбора нужной группы файлов, причем знак * соответствует любому множеству, даже пустому, в имени файла, а символ ? Соответствует ровно одному символу в имени.

Первая строка программы содержит имя файла, состоящее только из заглавных букв латинского языка (A-Z), а вторая — образец, содержащий только заглавные буквы латинского алфавита и, возможно, символы * и ?. Строки не превышают по длине 700 символов. Требуется вывести слова YES или NO в зависимости от того, сопоставляется ли имя файла указанному образцу.

Формат входных данных

Формат выходных данных

или

Примеры

Стандартный ввод	Стандартный вывод

Задача 4. Точные квадраты

Ограничение по времени: 1 секунды

Ограничение по памяти: 256 мегабайта

Можете ли вы по десятичному представлению натурального числа определить, является ли это число полным квадратом? А если в числе много десятичных знаков?

Формат входных данных

Первая строка содержит $5 \le N \le 10^6$ — количество тех чисел, которые нужно проверить. В последующих N строках — десятичные представления натуральных чисел количеством десятичных цифр в представлении не более 100.

Формат выходных данных

Для каждого из чисел, являющихся полным квадратом, вывести его номер. Нумерация начинается с единицы.

Примеры

Стандартный ввод	Стандартный вывод

Задача 5. Такси

Ограничение по времени: 2 секунды

Ограничение по памяти: 32 мегабайта

В некотором очень большом городе руководство осознало, что в автономные такси, то есть, такси без водителя — большое благо и решило открыть 10 станций по аренде таких такси. Были получены данные о том, откуда клиенты могут заказывать машины. Было замечено, что если станция находится от клиента на расстоянии, не большем, чем некое число R, то клиент будет арендовать машину именно на этой станции, причем, если таких станций несколько — клиент может выбрать любую. Для экономии, станции решено строить только в местах возможного расположения клиентов. Задача заключается в том, чтобы определить места наилучшей постройки, то есть такие, которые могут обслужить наибольшее количество клиентов.

Формат входных данных

В первой строке входного файла — два числа, количество клиентов и значение параметра R. В каждом из последующих N — два числа, координаты X_i и Y_i i-го клиента (нумерация ведется с нуля).

Формат выходных данных

В выходном файле должно присутствовать не более 10 строк. Каждая строка должна содержать номер клиента, у которого выгоднее всего строить станцию, и количество обслуживаемых этой станцией клиентов, отличное от нуля. Выводимые строки должны быть упорядочены от наибольшего количества обслуживаемых клиентов к наименьшему. Если две и более станции могут обслужить одинаковое количество клиентов, то выше в списке должна находиться станция с меньшим номером.

Примеры

Стандартный ввод	Стандартный вывод