

Лабораторная работа №5

В рамках данной лабораторной 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. При решении задач необходимо писать свои собственные структуры данных и методы сортировки, использование встроенных не допускается (кроме массивов).

Отчет о выполнении практической работы должен содержать:

1. Титульный лист

2. Содержание

3. 4 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*. Если для задачи предусмотрены автотесты, приложить скриншот их прохождения. Если нет: 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).

Каждая лабораторная работа защищается на занятии преподавателю.

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 подстроки длины L произведение максимального количества повторяющихся подстрок этой длины на длину L .

В нашем случае вес длины 1 есть 2 ($2 \cdot 1$), длины 2 есть 4 ($2 \cdot 2$), длины 3 — 3 ($1 \cdot 3$), 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:

— добавить в базу запись с ключом KEY и значением $VALUE$. Если такая запись уже есть, вывести **ERROR**.
— 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 $5 \leq N \leq 10^6$ — 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 некоторое 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

Задача 6. Поисковая система

Ограничение по времени: 4 секунды (Python – 6,5 секунд).

Ограничение по памяти: 128 мегабайт

Тимофей пишет свою поисковую систему.

Имеется n документов, каждый из которых представляет собой текст из слов. По этим документам требуется построить поисковый индекс. На вход системе будут подаваться запросы. Запрос — некоторый набор слов. По запросу надо вывести 5 самых релевантных документов.

Релевантность документа оценивается следующим образом: для каждого уникального слова из запроса берётся число его вхождений в документ, полученные числа для всех слов из запроса суммируются. Итоговая сумма и является релевантностью документа. Чем больше сумма, тем больше документ подходит под запрос.

Сортировка документов на выдаче производится по убыванию релевантности. Если релевантности документов совпадают — то по возрастанию их порядковых номеров в базе (то есть во входных данных).

Формат ввода

В первой строке дано натуральное число n - количество документов в базе ($1 \leq n \leq 10^4$). Далее в n строках даны документы по одному в строке. Каждый документ состоит из нескольких слов, слова отделяются друг от друга одним пробелом и состоят из маленьких латинских букв. Длина одного текста не превосходит 1000 символов. Текст не бывает пустым.

В следующей строке дано число запросов - натуральное число m ($1 \leq m \leq 10^4$). В следующих m строках даны запросы по одному в строке. Каждый запрос состоит из одного или нескольких слов. Запрос не бывает пустым. Слова отделяются друг от друга одним

пробелом и состоят из маленьких латинских букв. Число символов в запросе не превосходит 100.

Формат вывода

Для каждого запроса выведите на одной строке номера пяти самых релевантных документов. Если нашлось менее пяти документов, то выведите столько, сколько нашлось. Документы с релевантностью 0 выдавать не нужно.

Примеры

Ввод	Вывод
3 i love coffee coffee with milk and sugar free tea for everyone	1 2 3 2 1
3 i like black coffee without milk everyone loves new year mary likes black coffee without milk	
6 buy flat in moscow rent flat in moscow sell flat in moscow want flat in moscow like crazy clean flat in moscow on weekends renovate flat in moscow	4 5 1 2 3
1 flat in moscow for crazy weekends	

Задача 7. Хеш-таблица

Ограничение по времени: 5 секунд (Python – 15 секунд).

Ограничение по памяти: 64 мегабайта

Тимофей, как хороший руководитель, хранит информацию о зарплатах своих сотрудников в базе данных и постоянно её обновляет. Он поручил вам написать реализацию хеш-таблицы, чтобы хранить в ней базу данных с зарплатами сотрудников.

Хеш-таблица должна поддерживать следующие операции:

- put key value – добавление пары ключ-значение. Если заданный ключ уже есть в таблице, то соответствующее ему значение обновляется.
- get key – получение значения по ключу. Если ключа нет в таблице, то вывести «None». Иначе вывести найденное значение.
- delete key – удаление ключа из таблицы. Если такого ключа нет, то вывести «None», иначе вывести хранимое по данному ключу значение и удалить ключ.

В таблице хранятся уникальные ключи.

Требования к реализации:

- Нельзя использовать имеющиеся в языках программирования реализации хеш-таблиц (std::unordered_map в C++, dict в Python, HashMap в Java, и т. д.)
- Число хранимых в таблице ключей не превосходит 10^5 .

- Разрешать коллизии следует с помощью метода цепочек или с помощью открытой адресации.
- Все операции должны выполняться за $O(1)$ в среднем.
- Поддерживать рехеширование и масштабирование хеш-таблицы не требуется.
- Ключи и значения, id сотрудников и их зарплата, – целые числа. Поддерживать произвольные хешируемые типы не требуется.

Формат ввода

В первой строке задано общее число запросов к таблице n ($1 \leq n \leq 10^6$).

В следующих n строках записаны запросы, которые бывают трех видов – get, put, delete – как описано в условии.

Все ключи и значения – целые неотрицательные числа, не превосходящие 10^9 .

Формат вывода

На каждый запрос вида get и delete выведите ответ на него в отдельной строке.

Пример 1

Ввод	Вывод
10 get 1 put 1 10 put 2 4 get 1 get 2 delete 2 get 2 put 1 5 get 1 delete 2	None 10 4 4 None 5 None 5 None
8 get 9 delete 9 put 9 1 get 9 put 9 2 get 9 put 9 3 get 9	None None 1 2 3

Задача 8. Лемма о накачке

Ограничение по времени: 2 секунды.

Ограничение по памяти: 1024 мегабайта

Вам даны две строки s , t длины n , m , соответственно. Обе строки состоят из строчных букв латинского алфавита.

Подсчитайте тройки (x, y, z) строк, для которых справедливы следующие условия:

- $s = x + y + z$ (символ $+$ обозначает конкатенацию);
- $t = x + \underbrace{y + \dots + y}_{k \text{ раз}} + z$ для некоторого целого числа k .

Входные данные

Первая строка содержит два целых числа n и m ($1 \leq n < m \leq 10^7$) — длины строк s и t , соответственно.

Вторая строка содержит строку s длины n , состоящую из строчных латинских букв.

Третья строка содержит строку t длины m , состоящую из строчных латинских букв.

Выходные данные

Выведите одно целое число: количество допустимых троек (x, y, z) .

Примеры

Ввод	Вывод
4 8 abcd abcbcbcd	1
3 5 aaa aaaaa	5
12 16 abbababacaab abbababababacaab	8

Примечание

В первом наборе входных данных единственной подходящей тройкой является $(x, y, z) = ("a", "bc", "d")$. Действительно,

- $"abcd" = "a" + "bc" + "d"$
- $"abcbcbcd" = "a" + "bc" + "bc" + "bc" + "d"$

Во втором наборе входных данных существует 55 подходящих троек:

- $(x, y, z) = ("", "a", "aa")$
- $(x, y, z) = ("", "aa", "a")$
- $(x, y, z) = ("a", "a", "a")$
- $(x, y, z) = ("a", "aa", "")$
- $(x, y, z) = ("aa", "a", "")$

В третьем наборе входных данных существует 88 подходящих троек:

- $(x, y, z) = ("ab", "ba", "babacaab")$
- $(x, y, z) = ("abb", "ab", "abacaab")$
- $(x, y, z) = ("abba", "ba", "bacaab")$
- $(x, y, z) = ("ab", "baba", "bacaab")$
- $(x, y, z) = ("abbab", "ab", "acaab")$
- $(x, y, z) = ("abb", "abab", "acaab")$
- $(x, y, z) = ("abbaba", "ba", "caab")$
- $(x, y, z) = ("abba", "baba", "caab")$

Задача 9. Совпадения

Ограничение по времени: 2 с.

Ограничение по памяти: 64 мегабайта

Дана непустая строка S . Нужно найти такое наибольшее число k и строку t , что s совпадает со строкой t , выписанной k раз подряд.

Описание входных данных

Одна строка длины N , ($1 \leq N \leq 6 \cdot 10^5$), состоящая только из маленьких латинских букв.

Описание выходных данных

Вывести k и строку t .

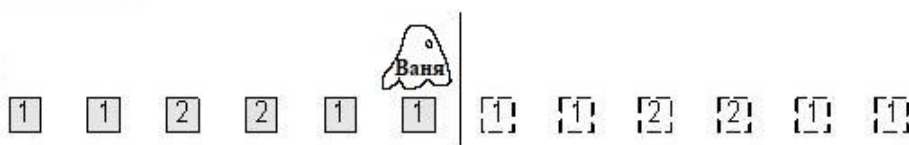
Задача 10. Привидение Ваня

Ограничение по времени: 2 секунды.

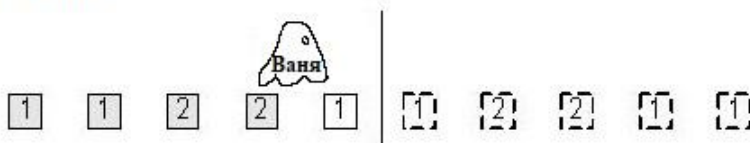
Ограничение по памяти: 64 мегабайта

Привидение Ваня любит играть со своими плитками. Он любит выкладывать их в ряд и разглядывать свое творение. Однако недавно друзья решили подшутить над Ваней и поставили в его игровой комнате зеркало. Ведь всем известно, что привидения не отражаются в зеркале! А плитки отражаются. Теперь Ваня видит перед собой N цветных плиток, но не знает, какие из этих плиток настоящие, а какие — всего лишь отражение в зеркале. Помогите Ване! Выясните, сколько плиток может быть у Вани. Ваня видит отражение всех плиток в зеркале и часть плиток, которая находится перед ним. Часть плиток может быть позади Вани, их он не видит.

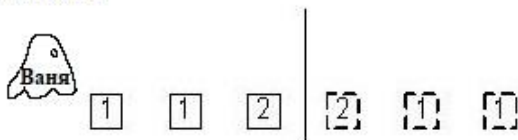
6 плиток



5 плиток



3 плитки



Описание входных данных

Первая строка входного файла содержит число N ($1 \leq N \leq 10^6$) и количество различных цветов, в которые могут быть раскрашены плитки — M ($1 \leq M \leq 10^6$). Следующая строка содержит N целых чисел от 1 до M — цвета плиток

Описание выходных данных

Выведите в выходной файл все такие K , что у Вани может быть K плиток в порядке убывания.