

Данная задача состоит из двух частей: базовой и дополнительной. Базовая часть обязательна к решению. Дополнительная является факультативной, но, будучи решенной, может выгодно подчеркнуть знания Кандидата.

Базовая задача

Японцы бесконечно влюблены в технику, которая их окружает. Они внимательно следят за всеми техническими новинками и стараются пользоваться самыми современными и «умными» из них. У Дена и Сергея есть гениальный план: они хотят создать текстовый редактор, который покори́т японцев. Важнейшей уберинтеллектуальной функцией редактора должна стать функция автодополнения. Если пользователь набрал несколько первых букв слова, редактор должен предложить ему самые правдоподобные окончания.

Ден и Сергей уже собрали огромное количество японских текстов. Для каждого слова японского языка они посчитали число раз, которое оно встречается в текстах. Если пользователь уже ввел несколько букв, то редактор должен показать не более десяти самых часто употребляемых слов, начинающихся со введенных пользователем букв, отсортированных по убыванию частоты упоминания.

Помогите Сергею с Деном перевернуть рынок текстовых редакторов.

Исходные данные

В первой строке находится единственное число N ($1 \leq N \leq 10^5$) — количество слов в найденных текстах. Каждая из следующих N строк содержит слово w_i (непустая последовательность строчных латинских букв длиной не более 15) и целое число n_i ($1 \leq n_i \leq 10^6$) — число раз, которое встречается это слово в текстах. Слово и число разделены единственным пробелом. Ни одно слово не повторяется более одного раза. В $(N + 2)$ -й строке находится число M ($1 \leq M \leq 15000$). В следующих M строках содержатся слова u_i (непустая последовательность строчных латинских букв длиной не более 15) — начала слов, введенных пользователем.

Результат

Для каждой из M строк необходимо вывести наиболее часто употребляемые японские слова, начинающихся с u_i , в порядке убывания частоты. В случае совпадения частот слова необходимо сортировать по алфавиту. Если существует больше десяти возможных вариантов, то вывести нужно лишь первые десять из них. Варианты дополнения для каждого слова необходимо разделять переводами строк.

Пример

исходные данные	результат
5	kanojo
kare 10	kare
kanojo 20	korosu
karetachi 1	karetachi
korosu 7	kanojo
sakura 3	kare
3	karetachi
k	
ka	kare
kar	karetachi

Решение должно отвечать следующим требованиям:

- Программа должна являться консольным приложением.
- Входные данные подаются программе в стандартном потоке ввода (ввод с клавиатуры). Программа должна выводить ответ в стандартный поток вывода (вывод на экран).
- Программа должна выводить только те данные, которые требует условие задачи. Выводить приглашение для ввода («Введите N:») не нужно. Также не нужно ожидать нажатия клавиши в конце работы программы.
- Скрипт для сборки
- Код на Java в стиле, соответствующем рекомендациям <https://google-styleguide.googlecode.com/svn/trunk/javaguide.html>
- Решение должно работать быстро (не больше 1-10 секунд) на тестовом файле test.in, прилагающемся к заданию.

Дополнение

Преобразовать консольное приложение из базовой задачи в сетевой сервис и реализовать клиента для него.

Сервис обрабатывает команды вида "get <prefix>". В ответ на которые отправляет клиенту наиболее часто употребляемые слова из словаря, начинающиеся с <prefix>, в порядке убывания частоты. В случае совпадения частот слова необходимо сортировать по алфавиту. Варианты дополнения для каждого слова необходимо разделять переводами строк.

В серверном приложении реализовать поддержку нескольких одновременных клиентских соединений с эффективной многопоточностью. Эффективность измеряется значением throughput.

Ограничения на использование серверных ресурсов отсутствуют.

Сервер

Сервер — это консольное приложение, которое при старте принимает в качестве параметра путь к текстовому файлу, содержащему словарь, и номер порта.

Клиент

Консольное приложение.

При старте принимает в качестве параметров командной строки IP адрес или hostname, а также порт сервера.

Общие требования к организации сетевого обмена

* TCP/IP v4

* Сервер должен обслуживать все соединения по заданному порту на всех доступных сетевых интерфейсах

* Кодировка протокола ASCII

Напутствие

К тестовому заданию следует относиться как к упрощенной аналогии настоящего технического задания. Проверая тестовое задание, рецензент оценивает не только подход к решению, но и критерии качества самого Кандидата.

То есть результатом выполнения тестового задания должно быть идеально написанное с точки зрения Кандидата решение, готовое для production. Протестированное, удовлетворяющее

исходному техническому заданию, имеющее максимально аккуратный и чистый код (для простой дальнейшей поддержки).

Не следует посылать задание, которое работает дольше 10 секунд или завершается ошибкой при запуске на тестовом файле test.in. Рецензент будет рассматривать это как невнимательность и халатность при подготовке задания к отправке на проверку, что значительно уменьшает шансы Кандидата.