

Задача А. Сервера

Имя входного файла: a.in
 Имя выходного файла: a.out
 Ограничение по времени: 0.5 с
 Ограничение по памяти: 256 Мб

Примечание

Источник — Всесибирская олимпиада, очный тур (29 ноября 2009).server

Компьютерная сеть в некотором доме строилась по принципу присоединения нового компьютера к последнему из уже подключенных. Никакие два компьютера, будучи подключенными в сеть, между собой дополнительно никак не связывались. Таким образом, в сеть были объединены последовательно N компьютеров. Соседи обменивались информацией между собой, но в какой-то момент поняли, что им нужны прокси-серверы. Компьютерное сообщество дома решило установить прокси-серверы ровно на K компьютеров. Осталось только решить, какие именно компьютеры выбрать для этой цели. Главным критерием является ежемесячная стоимость обслуживания серверами всех компьютеров.

Для каждого компьютера установлен тариф его обслуживания, выраженный в рублях за метр провода. Стоимость обслуживания одного компьютера каким-то сервером равна тарифу компьютера, умноженному на суммарную длину провода от этого компьютера до сервера, которым он обслуживается.

Ваша задача написать программу, которая выберет такие компьютеры, чтобы установить на них прокси-серверы, что общие затраты на обслуживание всех компьютеров были бы минимальными

Формат входного файла

В первой строке входного файла записано два целых числа N и K — количество компьютеров в сети и количество прокси-серверов, которые нужно установить ($1 \leq K \leq N \leq 2000$).

Все компьютеры в сети пронумерованы числами от 1 до N по порядку подключения.

Во второй строке записано одно целое число T_1 — тариф обслуживания первого компьютера.

В следующих $N - 1$ строках записано через пробел по два целых неотрицательных числа L_i , T_i — информация об остальных компьютерах в сети по порядку номеров. L_i — длина провода, соединяющего i — компьютер с соседним с меньшим номером, T_i — тариф обслуживания данного компьютера ($2 \leq i \leq N$). Все L_i и T_i не превышают 10^6 .

Формат выходного файла

В первую строку выходного файла необходимо вывести одно целое число — минимальную стоимость обслуживания всех компьютеров всеми серверами. Во второй строке должны быть записаны через пробел K номеров компьютеров, на которые необходимо установить серверы. При существовании нескольких вариантов размещения разрешается вывести любой.

Пример

a.in	a.out
3 1	19
10	1
2 2	
3 3	
3 2	4
10	1 3
2 2	
3 3	

Задача В. Wordperiod

Имя входного файла: `b.in`
 Имя выходного файла: `b.out`
 Ограничение по времени: 1 с
 Ограничение по памяти: 256 Мб

Примечание

Источник — чемпионат MIPT (май 2011).wordperiod

Назовём периодом слова s такое слово t , длина которого не превосходит длины слова s и для которого существует такое натуральное число k , что слово s является префиксом слова t^k (то есть слова, полученного конкатенацией k копий слова t). Например, периодами слова **хузхузх** являются слова **хуз**, **хузхуз**, **хузхузх**.

Пусть имеется некоторое слово w длины l . Рассмотрим l слов длины $l - 1$, i — из которых получено из слова w вычёркиванием его i — буквы. Для каждого из этих слов найдём период наименьшей длины. Выведите наименьшее из получившихся l чисел.

Формат входного файла

В первой строке входного файла задано целое число d ($1 \leq d \leq 10$) — количество тестовых примеров. В последующих d строках заданы тестовые примеры, по одному на строку. В начале i — тестового примера идёт число n_i ($2 \leq n_i \leq 200\,000$) — длина l слова w . Далее через пробел следует слово w , состоящее из l строчных латинских букв.

Формат выходного файла

Для каждого тестового примера выведите в отдельной строке одно число — минимальную длину периода слова, полученного выбрасыванием из исходного слова одной буквы.

Пример

<code>b.in</code>	<code>b.out</code>
1 8 ababcaba	2

Пояснение

Для слова w из тестового примера имеем следующие слова, наименьшие периоды и их длины:

- **ababcaba** — **babca** — длина 5;
- **aababcaba** — **aabcab** — длина 6;
- **abbcaba** — **abbcab** — длина 6;
- **abacaba** — **abac** — длина 4;
- **abababa** — **ab** — длина 2;
- **ababcba** — **ababcb** — длина 6;
- **ababcaa** — **ababca** — длина 6;
- **ababcab** — **ababc** — длина 5.

Соответственно, наименьшая из длин равна 2, что и является ответом на тестовый пример.

Задача C. Arithmetic

Имя входного файла: c.in
Имя выходного файла: c.out
Ограничение по времени: 0.5 с
Ограничение по памяти: 256 Мб

Примечание

Источник — XV чемпионат СПбГУ (22 октября 2006).arithmetic

Вы только что получили задание от Kitten Computing, известного разработчика программного обеспечения. Вы не можете поверить в ваше счастье — множество ваших друзей подавали туда заявку раньше, но они либо провалились на интервью, либо не смогли выполнить их первое задание и были уволены в первый месяц их работы в Kitten Computing.

Сейчас вы и сами получили ваше первое задание, в команде которая разрабатывает программное обеспечение для нового арифметического кодирования. Напомним, что во время арифметического кодирования, строка, которую нужно закодировать, переходит в интервал (α, β) реальной строки, а затем любая дробь p/q на этом интервале и есть результат кодирования.

Ваше задание — найти эту дробь p/q , так чтобы числитель и знаменатель были как можно меньше, для заданных рациональных чисел α и β . Вам не нужно беспокоиться о происхождении α и β : программное обеспечение для создания этих чисел написал другой человек из вашей команды.

Формат входного файла

Каждая строка входного файла содержит четыре целых числа P_1, Q_1, P_2, Q_2 , таких, что $\alpha = P_1/Q_1$ и $\beta = P_2/Q_2$. Все числа неотрицательные и не превышают 10^{18} , знаменатели Q_1 и Q_2 отличны от нуля. Гарантируется, что $\alpha < \beta$. Во входном файле содержится не более 10^4 строк.

Формат выходного файла

Для каждой строки ввода вы должны выдать два натуральных числа — P и Q , разделенные одним пробелом, такие, что $\alpha < P/Q < \beta$. При этом Q должно быть минимально возможным. Если есть несколько оптимальных Q , выводите дробь с минимальным P .

Пример

c.in	c.out
3 10 7 20	1 3
0 1 1 1	1 2

Задача D. Таможенные правила

Имя входного файла: d.in
 Имя выходного файла: d.out
 Ограничение по времени: 0.5 с
 Ограничение по памяти: 256 Мб

Деньги на бочку!

Из заявки Диогена на грант

Примечание

Источник — XIX чемпионат СПбГУ (9 декабря 2007).tax

Author: Vitaly Valtman

Description Author: Vitaly Valtman

Text Author: Oleg Hristenko

Наконец, приз, состоящий из N золотых монет, был получен, и команда Поссилтума отбыла назад, в измерение Пент. По дороге к Аазу, Скиву, Гвидо и Нунцио присоединились попутчики из других команд, прибывших из Пента. Так что возвращавшаяся делегация составляла K человек.

Однако по пути им встретилось достаточно неприятное королевство, таможенное законодательство которого было не только очень строгим, но и очень запутанным.

В случае, если у прибывшего путешественника оказывалось с собой менее A золотых монет, путешественник отправлялся под арест за нищенство.

В случае, если путешественники прибыли группой, и у каких-то двух путешественников в группе количество золотых монет равно $t \cdot a$ и $t \cdot b$ соответственно, где t — целое число, большее единицы, а a и b — натуральные числа, то считается, что имел место сговор с целью подрыва местной валюты (установления её курса в t), и оба путешественника отправляются под арест за экономическую диверсию. Естественно, что с конфискацией орудия преступления (то есть золота).

Так что перед Аазом и Скивом встала задача — можно ли распределить сумму приза между всеми участниками делегации так, чтобы никто не был арестован при прохождении таможенного досмотра?

Формат входного файла

Входной файл содержит три натуральных числа. $1 \leq N \leq 1\,000\,000\,000$, $2 \leq K \leq 10$ и $1 \leq A \leq N$.

Формат выходного файла

Если пройти таможеню без ареста можно, то в первой строке выведите слово «YES», а во второй — K чисел a_i — выданное каждому участнику количество золотых монет.

В противном случае выведите единственную строку «NO».

Примеры

d.in	d.out
6 3 1	YES 1 2 3
6 3 2	NO
6 2 1	YES 1 5

Задача Е. Covering Points

Имя входного файла: e.in
Имя выходного файла: e.out
Ограничение по времени: 3 с
Ограничение по памяти: 256 Мб

Примечание

Источник — XX чемпионат СПбГУ (23 март 2008).covers

Author: Andrew Lopatin

Text Author: Andrew Lopatin

Description: DP

Сегодня, Вася отправляется в научно-исследовательский институт сфер и кругов (RISC). Этот институт изучает все задачи, связанные со сферами и кругами, и теперь Васю просят что-то сделать с кругами.

Даны N точек на плоскости, необходимо покрыть их все K кругами минимально возможного радиуса. Кругам разрешено касаться и пересекаться друг с другом. Не могли бы вы помочь Васе справиться с заданием?

Формат входного файла

Входной файл состоит из одного или нескольких тестов. Каждый тест начинается со строки, содержащей два целых числа N и K ($1 \leq N \leq 15$, $1 \leq K \leq N$), а затем N строк, каждая из которых описывает одну точку её целыми координатами x_i и y_i . Точки могут совпадать. Все координаты не превышают 1000 по абсолютной величине. Ввод завершается тестом $N = K = 0$, который учитывать не нужно. Общая сумма чисел N по всем тестам в одном входном файле не превышает 150.

Формат выходного файла

Для каждого теста сначала выведете минимальный радиус, затем приведите пример покрытия. Если есть несколько решений — выводите любое. Радиусы нужно вывести с точностью не менее 6 знаков после запятой. Чекер будет осуществлять проверку “попала ли точка в круг” с точностью 10^{-6} . Мы рекомендуем выводить числа с максимальной точностью.

Заметьте, что круг с нулевым радиусом — точка.

Примеры

e.in
3 2 0 0 0 1 0 2 3 1 0 0 0 1 1 0 0 0
e.out
Case 1: The minimal possible radius is 0.5 circle 1 at (0.0, 0.5) circle 2 at (0.0, 2.0) Case 2: The minimal possible radius is 0.7071067811865476 circle 1 at (0.5, 0.5)

Задача F. Гамильтонов цикл в полном графе

Имя входного файла: `f.in`
Имя выходного файла: `f.out`
Ограничение по времени: 0.5 с
Ограничение по памяти: 256 Мб

Примечание

Источник — из задач спецкурса 3-го курса СПбГУ (весна 2011).fullham

Original idea : Folklore

Text : Sergey Kopeliovich

Tests : Sergey Kopeliovich

Дан граф из N вершин, в котором степень любой вершины не меньше $\frac{N}{2}$. Ваша задача — найти гамильтонов цикл.

Формат входного файла

На первой строке входного файла записано целое число N ($3 \leq N \leq 4000$) — количество вершин в графе. На следующих N строках записана матрица смежности. Т.к. матрица смежности симметрична, а на диагонали всегда стоят нули, на i -й строке записаны $i - 1$ символ — нули и единицы. Если j -й символ i -й строки равен единице, значит есть ребро между вершинами i и j .

Гарантируется, что в графе есть гамильтонов цикл и, что степень каждой вершины не меньше $\frac{N}{2}$.

Формат выходного файла

Выведите перестановку из N чисел — номера вершин в порядке гамильтонова цикла.

Пример

<code>f.in</code>	<code>f.out</code>
4	1 2 3 4
1	
11	
101	

Задача G. PreQueL

Имя входного файла: g.in
Имя выходного файла: g.out
Ограничение по времени: 2 с
Ограничение по памяти: 256 Мб

Примечание

Источник — Дальневосточные областные соревнования NEERC
Far Eastern NEERC Subregional Contests (год = ?).prequel

В некоторых упрощенных СУБД называемых PreQueL, единственный допустимый тип названия - CHAR (1) (один символ), а кроме того, его название должно состоять только из английских заглавных букв (от A до Z). Таблица может содержать до 9 столбцов, пронумерованных от 1 до 9. Таблицы называются строчными английскими буквами (от a до z).

Единственно возможный запрос к базе данных сначала **соединяет** все таблицы (см. пояснение к задаче в конце условия); потом выбирает некоторые строки в соответствии с условиями. Бывают условия двух видов: $\langle \text{column} \rangle = \langle \text{value} \rangle$ или $\langle \text{column1} \rangle = \langle \text{column2} \rangle$, например, $a2=A$ или $b1=c4$. Все условия должны выполняться одновременно, как если бы они были соединены с помощью оператора AND.

Вы должны написать обработчик запросов к PreQueL, который, с учетом данных ему таблиц и набора условий, будет выдавать результат запроса, то есть те строки которые удовлетворяют всем условиям. Строки в ответе должны быть упорядочены лексикографически по алфавиту.

Формат входного файла

Первая строка входного файла содержит два целых числа — количество таблиц T и количество условий, D .

Начиная со второй строки даются T таблиц. В первой строке описания таблицы даются размеры таблицы — RN и CN . Далее идёт RN строк, состоящих ровно из CN символов каждая. После таблицы идут D строк с условиями. Ограничения: $1 \leq T \leq 26$, $1 \leq D \leq 50$, $1 \leq C \leq 9$, $1 \leq R \leq 1000$.

Формат выходного файла

Выходной файл содержит строки результата, каждую строку таблицы выведете на отдельной строке файла. Гарантируется, что ни один входной запрос не выдаст более 1000 строк.

Пример

g.in	g.out
2 2	AXACD
3 2	BXBCC
AX	
BX	
BY	
2 3	
ACD	
BCC	
a1=b1	
a2=x	

Пояснение

- Строки в выведенном файле должны быть упорядочены по возрастанию.
- Если мы **соединяем** таблицы A и B без применения каких-либо условий, число строк в результате будет $|A| \cdot |B|$.
- Пусть у нас есть три таблицы (каждая таблица представлена в виде списка строк): {A, B}, {C, D}, {XX, YY, ZZ}. Их **соединением** будет таблица из 12 строк {ACXX, ACYY, ACZZ, ADXX, ADYY, ADZZ, BCXX, BCYY, BCZZ, BDXX, BDYY, BDZZ}.

Задача Н. Sigma-функция на отрезке

Имя входного файла: `h.in`
Имя выходного файла: `h.out`
Ограничение по времени: 1.5 с
Ограничение по памяти: 256 Мб

Примечание

Источник из задач к экзамену 1-го курса СПбГУ (осень 2010).lrsigma

Нужно научиться считать $\sum_{i=L}^R \sigma(i)$. Где $\sigma(i)$ — сумма натуральных делителей числа i .

Формат входного файла

Последовательность из не более чем 10^5 запросов. Каждый запрос записан на отдельной строке. Формат запроса прост: числа L, R ($1 \leq L \leq R \leq 5 \cdot 10^6$).

Формат выходного файла

Для каждого запроса нужно вывести одно число — $\sum_{i=L}^R \sigma(i)$.

Пример

<code>h.in</code>	<code>h.out</code>
3 10	83

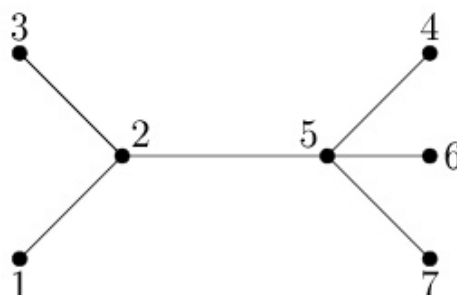
Задача I. Autotourism

Имя входного файла: i.in
Имя выходного файла: i.out
Ограничение по времени: 2 с
Ограничение по памяти: 256 Мб

Примечание

Источник — чемпионат MIPT (май 2011).autotourism

В Бейтландии существуют n городов, соединённых $n - 1$ дорогами с двусторонним движением таким образом, что из каждого города можно проехать в любой другой по сети дорог. Длина каждой дороги равна 1 километру.



Бензобак автомобиля позволяет проехать без заправки m километров. Требуется выбрать маршрут, позволяющий посетить наибольшее количество различных городов без дозаправки. При этом начинать и заканчивать маршрут можно в произвольных городах.

Формат входного файла

В первой строке входного файла заданы два целых числа n и m ($2 \leq n \leq 500\,000$, $1 \leq m \leq 200\,000\,000$) — количество городов в стране и количество километров, которое автомобиль может проехать без дозаправки. В последующих $n - 1$ строках описаны дороги. Каждая дорога задаётся двумя целыми числами a и b ($1 \leq a, b \leq n$) — номерами городов, которые она соединяет. Длина каждой дороги равна 1 км.

Формат выходного файла

Выведите одно число — максимальное количество городов, которое можно посетить без дозаправки.

Пример

i.in	i.out
7 6 1 2 2 3 2 5 5 6 5 7 5 4	5

Пояснение

5 городов можно посетить, например, по схеме $4 \rightarrow 5 \rightarrow 7 \rightarrow 5 \rightarrow 6 \rightarrow 5 \rightarrow 2$ или по схеме $3 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 5 \rightarrow 6 \rightarrow 5$.

Задача J. Perfect Powers

Имя входного файла: j.in
Имя выходного файла: j.out
Ограничение по времени: 0.5 с
Ограничение по памяти: 256 Мб

Примечание

Источник — IX чемпионат СПбГУ (17 октября 2004).powers

Число x называется *perfect power*, если существуют такие целые числа a и b , что $x = a^b$, $b > 1$. Ваше задание — вычислить количество таких чисел внутри отрезка $A \leq x \leq B$.

Формат входного файла

Два целых числа A и B , по абсолютной величине не превосходящие 10^{14} .

Формат выходного файла

Одно целое число.

Пример

j.in	j.out
3 14	3
25 50	5

Задача К. Spell Checker

Имя входного файла: k.in
 Имя выходного файла: k.out
 Ограничение по времени: 0.3 с
 Ограничение по памяти: 256 Мб

Примечание

Источник — IX чемпионат СПбГУ (17 октября 2004).spell

Университет магии Kurufinwe в Alkarinqwe широко известен как один из самых сильных университетов магии в стране эльфов Laureling. Сейчас Архимаги этого университета вплотную подошли к разгадке тайны древнего языка магических элементарей Qwen'Iaur. У них уже есть список всех слов этого языка, но комбинации этих слов не всегда дают работающее заклинание.

К счастью, Архимаги нашли глубоко в подземельях их замка древний артефакт Сарандол, также известный как Оракл. Легенды прошлого гласят, что этот артефакт способен создавать работающие заклинания! Но он был создан одним из величайших магов древности Белигуром Великим и у него есть особая механизм защиты. Оракл выдает заклинание в виде строки без каких либо пробелов между словами, так что оно может быть прочитано только как алфавитная строка S длины L . Но это только первая проблема. Вторая заключается в том, что артефакт иногда генерирует последовательности, которые не могут быть составлены из слов языка Qwen'Iaur, т.е. не существует такой последовательности слов a_1, a_2, \dots, a_k , что $S = a_1a_2 \dots a_k$, где все слова a_i присутствуют в языке Qwen'Iaur (одно и то же слово может быть использовано более одного раза). Но если хотя бы одно такое представление существует, то можно заключить, что Оракл сгенерировал строку, которая является работающим заклинанием.

Ваше задание — написать магическую программу для вашего магического компьютера, которая проверит существование хотя бы одного такого представления.

Формат входного файла

В первой строке ввода содержится одно целое число $1 \leq N \leq 1000$ — количество слов в словаре. N следующих строк содержат слова языка Qwen'Iaur. Последняя строка входного файла — строка S , сгенерированная Сарандолом. Все слова в словаре имеют длину не более 100 символов. $L \leq 10000$. Слова и строки не чувствительны к регистру и могут состоять только из английских букв и двух символов ' и ' (эти символы имеют коды ASCII 39 и 96).

Формат выходного файла

Если строка может быть прочитана как рабочее заклинание, выведите YES, иначе выведите NO.

Пример

k.in	k.out
4 all bar Akh kar bakarkarbar	NO
4 all bar Akh kar AllakhAkhbar	YES