

Задача A. Easy Problems

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

У жюри отборочного турнира, проводимого по системе ТСМ, есть P простых задач. В каждый отборочный раунд жюри, если есть возможность, ставит K таких задач, а если такой возможности нет (осталось меньше K задач), то ставит все оставшиеся задачи. По исчерпанию простых задач отборочный турнир заканчивается.

Ваша задача — по заданным P и K вычислить количество отборочных раундов серии.

Формат входного файла

Во входном файле заданы 2 целых числа: P и K — соответственно общее количество простых задач и количество простых задач, которое жюри при возможности ставит в проблемсет каждого раунда ($0 < P \leq 10^9$, $0 < K \leq 5000$).

Формат выходного файла

Выведите продолжительность серии в днях.

Пример

<code>standard input</code>	<code>standard output</code>
11 3	4

Задача В. Checkers

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

При подготовке любого конкурса рано или поздно возникает необходимость «to set up checkers». В этой задаче чем-то подобным предстоит заняться Вам.

На доске 7×7 расставлены несколько цифр от 0 до 8, при этом существует единственная такая расстановка 10 шашек на остальных клетках, что рядом с каждой из клеток, занятых цифрой K , находятся ровно K шашек (две клетки находятся рядом, если они не совпадают и при этом имеют хотя бы одну общую точку границы). Восстановите эту расстановку.

Формат входного файла

Во входном файле заданы 7 строк, каждая из которых имеет длину 7 — конфигурация доски. Поля, занятые цифрами, обозначены соответствующими цифрами. Остальные поля обозначены точками.

Формат выходного файла

Выведите требуемую расстановку шашек, заменив на тех полях, на которых должны стоять шашки, точку на звездочку (см. пример).

Пример

standard input	standard output
1...0.	1...0.
.1.....	*1.*...
1..3...	1..3*..
..23.2.	..23*2.
..3.....	.*3*...
.3....2	*3*...*2
1.1..2.	1.1..2*

Задача C. Book

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Книга «Школьная информатика во мне» школьного тренера, в карьере которого — победы на Всероссийских олимпиадах по программированию по версиям NEERC и Воронежа в одном сезоне, состоит из нескольких разделов, в каждом из которых, в свою очередь, имеются подразделы. Подраздел также может иметь вложенные подразделы и так далее. Назовём разделы «подразделами нулевого уровня», подразделы, непосредственно вложенные в разделы — «подразделами первого уровня»... подразделы, непосредственно вложенные в разделы N -го уровня — «подразделами $N + 1$ -го уровня».

Вам заданы пары «`text1/text2`», где `text1` — название подраздела $K + 1$ -го уровня, а `text2` — название подраздела K -го уровня, непосредственно в который вложен подраздел `text1`. С учётом того, что различные подразделы имеют различные названия, постройте оглавление книги, в котором за названием каждого раздела или подраздела следовали бы названия его подразделов (в случае, если таковые имеются), выделенные соответствующим отступом. Названия разделов, а также названия всех подразделов, непосредственно вложенные в какой-то раздел или подраздел, должны быть отсортированы лексикографически.

Формат входного файла

В первой строке входного файла задано число N ($1 \leq n \leq 300$). Далее следуют N строк вида `text1/text2`, где `text1` — название подраздела, а `text2` — название раздела или подраздела на единицу меньшего уровня, в который он вложен. Названия состоят из не более, чем 40 символов с кодом от 32 до 127. Название не может содержать знак `/`, не может заканчиваться пробелом или содержать более двух пробелов подряд.

Формат выходного файла

В выходной файл выведите оглавление в следующем формате. Название каждого из разделов записано с начала новой строки, при этом названия отсортированы лексикографически по возрастанию ASCII-кодов их символов. Под названием каждого раздела записаны с отступом в 4 пробела названия подразделов первого уровня, отсортированных — внутри каждого раздела — лексикографически по возрастанию ASCII-кодов их символов, и так далее, то есть под названием каждого подраздела i -го уровня записаны с отступом в $4(i+1)$ пробелов названия подразделов $i+1$ -го уровня, отсортированные — внутри каждого подраздела i -го уровня — лексикографически по возрастанию ASCII-кодов их символов.

Пример

standard input	standard output
9	Individual
ACM ICPC/Team	Codeforces
NEERC 2009/ACM ICPC	TopCoder
TopCoder/Individual	Voronezh ;)
Open Cup/Team	Yandex Open 2010
Finals 2010/ACM ICPC	trainings at home
Yandex Open 2010/Individual	Team
Voronezh ;)/Individual	ACM ICPC
Codeforces/Individual	Finals 2010
trainings at home/Individual	NEERC 2009
	Open Cup

Задача D. Password

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Во время обсуждения различных методов жульничества на соревнованиях по программированию один из участников обсуждения рассказал, что пароли в некоторой тестирующей системе выдаются вовсе не случайным образом.

Зарегистрировав с помощью системы автоматической регистрации нескольких пользователей, он выяснил, что пароль состоит из не более, чем 10 знаков — заглавных латинских букв и цифр. При этом для всех паролей, выданных в один и тот же день, длина пароля фиксирована и позиции букв и цифр определены точно. В начале каждого дня пароль состоит из букв **A** и нулей.

Каждый следующий выданный пароль строится следующим образом: последний символ пароля увеличивается на 1 (то есть если на последнем месте стоит буква, то берётся следующая в латинском алфавите, если цифра — соответствующая цифра увеличивается на 1). Если последней стояла цифра **9** или буква **Z**, то в следующем пароле на их месте будет **0** или **A** соответственно, а первый с конца символ будет обработан аналогичным образом. После того, как будет выдан пароль, состоящий из букв **Z** и девяток, система «уходит на профилактику» до конца дня. Первоначальный шаблон (то есть расположение букв и цифр) в начале каждого дня выбирается случайным образом.

По заданному Вам паролю вычислите предшествовавший ему пароль. Гарантируется, что в пароле будет хотя бы один символ, отличный от **A** и **0**.

если вероятность успешной сдачи задачи равна p , а общее количество сданных участником задач равно M .

Формат входного файла

Во входном файле задана пароль — непустая строка длины 10 или менее символов, каждый из которых является или заглавной латинской буквой, или цифрой. Гарантируется, что среди этих символов хотя бы один будет отличаться от **A** или **0**.

Формат выходного файла

Выведите пароль, предшествующий данному паролю при вышеописанной системе генерации паролей.

Пример

standard input	standard output
2A	1Z
YANDEX2010	YANDEX2009
TCM	TCL

Задача E. Blind Luck and Blind Submits

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Прочитав мнение известного problemsetter-а о том, что в соревнованиях, проводимых по системе ТСМ, большое значение имеет удача, один участник после того, как ему в траве случайно попался четырёхлистный клевер, решил сдавать «втёмную» все задачи одну за другой, в порядке их следования в тексте условий — причём даже без предварительной компиляции. В этом случае вероятность получить в результате +1.5, а не 0, оказалась — независимо от задачи, и, что самое главное, от клевера — равна p .

После того, как у участника «упали» N задач подряд, он пожаловался, что клевер какой-то неправильный — случилось неблагоприятное событие с шансом $(1 - p)^N$. Ему возразили, что дело не в клевере, а в незнании математики: ведь на самом деле турнир состоял из M задач, а значит, вероятность совсем другая.

Помогите незадачливому участнику посчитать, какова вероятность «падения» не менее N задач подряд, если вероятность успешной сдачи задачи равна p , а общее количество сданных участником задач равно M .

Формат входного файла

Во входном файле заданы 3 числа M , N и p ($0 \leq M \leq 2000$, $0 < N \leq M$, $0 \leq p \leq 1.0$), M и P — целые) — соответственно количество задач, необходимое количество «падений» и вероятность того, что задача не «упадёт».

Формат выходного файла

Выведите требуемую вероятность с точностью до 10^{-7} .

Пример

standard input	standard output
1 1 0.3	0.7
1997 2 0.99	0.17934616439188586
1569 35 0.8862111833857929	1.2507406416421154E-30

Задача F. Go onsite!

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

По регламенту экспериментального турнира по программированию, проведённого в некотором королевстве, вышедшим в онсайт-раунд участникам оплачивается билет в обе стороны на дилижансе, а перелёт на драконе оплачивается только в случае невозможности проезда на дилижансе.

Местная авиакомпания «Dragonflot» в рекламных целях решила стать эксклюзивным перевозчиком онсайт-раунда турнира. Для этого отделу практической магии поручено вызвать ливневые дожди над отдельными участками дорог так, чтобы проехать по ним было невозможно.

Схема дорог королевства является деревом. Движение на всех дорогах двустороннее. В одной из вершин дерева находится столица, в пригороде которой пройдёт онсайт-раунд. Города, из которых прибывают участники онсайта, находятся в «листьях» дерева (то есть в каждый из них входит только одна дорога). Для каждого участка дороги между городами известна стоимость размытия этого участка дорог силами отдела практической магии.

Ваша задача — определить минимальную стоимость такого размытия дорог, что ни из одного города, из которого прибывают участники онсайта, нельзя было добраться по сети дорог до столицы (за исключением, возможно, самой столицы, если она является «листом» дерева).

Формат входного файла

В первой строке входного файла заданы 2 целых числа: M — количество городов в королевстве, N — номер вершины, в которой находится столица ($1 \leq M \leq 1000$, $1 \leq N \leq M$). Далее идут $M - 1$ строк — описания дорог. Каждая из дорог определяется тремя целыми числами: b_i , e_i и l_i — городами, которые она соединяет, и ценой размытия ($1 \leq b_i, e_i \leq M$, $0 \leq l_i \leq 1000$). Каждая дорога описана во входном файле ровно один раз.

Формат выходного файла

Выведите минимальную суммарную стоимость размытия дорог, необходимого для того, чтобы отделить все города, из которых прибывают участники онсайта, от столицы.

Пример

standard input	standard output
2 2	3
1 2 3	