

Для всех задач:

Входной файл:	<i>input.txt</i>
Выходной файл:	<i>output.txt</i>
Ограничение по времени:	1 секунда на тест
Ограничение по памяти:	64 МБ
Максимальное количество баллов за задачу:	100 баллов

Задача 1. Успеть на поезд!

Пассажир планирует путешествие из города Нск в город В. Ему подходит поезд, следующий из города М в город В через Нск. Он узнал дату и время выхода поезда из М и время пути до Нска. Он мог бы рассчитать время прибытия поезда в Нск, но знает, что впереди будут даты перевода часов в этих городах. Причем в каждом городе дата своя и время сдвига тоже свое, но будут ли часовые стрелки переведены вперед или назад, пока неизвестно.

Стрелки на его часах переводятся автоматически согласно текущему времени в городе, не выдавая предупреждений и сообщений. Пассажир не хочет следить за переводом стрелок, а решил приехать на вокзал к самому раннему моменту времени возможного прибытия поезда. От дома до вокзала ему добираться один час и на время выхода он решил завести будильник.

Будильник на часах пассажир настроил так, что тот сработает, если в какой-то момент время станет равным или больше, чем задано. При этом в часах перевод стрелок выполняется раньше других действий, т.е. если в какой-то момент времени нужно перевести стрелки, то проверка времени срабатывания будильника осуществляется уже после перевода стрелок.

Хотя даты перевода и время перевода в каждом городе могут быть разные, но при переводе все следуют правилу: если стрелки переводятся вперед, то перевод происходит в 00 часов 00 минут заданной даты, если стрелки переводятся назад, то перевод происходит в такое время, чтобы новым временем стало 00 часов 00 минут заданной даты.

Если из-за перевода стрелок время отправления «перепрыгивается», то поезд отправляется в наиболее раннее и близкое к назначенному, но не меньшее его по значению. Поезд отправляется из города М. по текущему времени согласно заданному расписанию.

Поезд в пути находится менее одного года, год отправления из М. и год прибытия поезда в Нск — не високосные.

Напишите программу, которая высчитывает, на какое время пассажиру надо поставить будильник.

Входные данные

В первой строке входного файла задана дата и время отбытия поезда из города М. в формате **дд.мм чч.мм**, дата возможного перевода стрелок и целое число D_1 ($0 < D_1 < 24$) — количество часов, на которые стрелки могут быть переведены, вперед или назад, в формате **дд.мм чч**.

Во второй строке входного файла задана дата возможного перевода стрелок в городе Нск и целое число D_2 ($0 < D_2 < 24$) — количество часов, на которые стрелки могут быть переведены, вперед или назад.

В третьей строке задано целое число — время следования поезда в часах от города М. до города Нск.

Выходные данные

В единственную строку выходного файла необходимо вывести дату и время на будильнике пассажира в формате, задаваемом во входном файле. Момент начала суток следует выводить как **00.00**.

Пример

<i>input.txt</i>	<i>output.txt</i>
31.08 10.10 01.09 1 03.09 2 18	01.09 03.10

Задача 2. Периметр

Для двух заданных окружностей найти периметр их пересечения.

Периметр пересечения двух окружностей — это сумма длин дуг, которые ограничивают область пересечения соответствующих кругов.

Входные данные

Входной файл состоит из двух строк, на каждой из которых содержатся по три целых, не превосходящих 10^9 по абсолютной величине, числа X , Y и R — координаты центра и радиус соответствующей окружности.

Выходные данные

В единственную строку выходного файла необходимо вывести периметр пересечения с точностью не менее 10^{-6} .

Примеры

<i>input.txt</i>	<i>output.txt</i>
0 0 5 8 0 5	12.870022
1 2 10 14 3 7	23.240467

Задача 3. Увеличение на единицу

Вам дана последовательность целых чисел A_1, A_2, \dots, A_N .

Операция инкремента добавляет единицу к любому из элементов последовательности. Вам нужно так применить к элементам последовательности ровно K операций инкремента, чтобы произведение элементов результирующей последовательности было максимальным.

Входные данные

В первой строке входного файла записано два целых числа N и K — длина последовательности и количество операций инкремента соответственно ($1 \leq N \leq 100$, $1 \leq K \leq 10000$).

Следующая строка содержит N целых чисел, записанных через пробел, представляющих последовательность A_1, A_2, \dots, A_N . Все элементы последовательности по модулю не превосходят 50.

Выходные данные

В выходной файл нужно вывести измененную последовательность после применения K операций инкремента, которые дают максимальное произведение. Числа выводить в одну строку через пробел.

Вычисление оптимального произведения последовательности производить с абсолютной или относительной погрешностью, не превосходящей 10^{-10} .

Примеры

<i>input.txt</i>	<i>output.txt</i>
4 6 5 3 1 -1	5 3 3 3
10 500 50 50 50 50 50 50 50 50 50 50	100 100 100 100 100 100 100 100 100 100

Задача 4. Киномания

Маша и Света очень любят кино. Они посмотрели большое количество фильмов, и придумали следующую игру. Им нужно попытаться найти связь между известными им актерами, занятыми в фильмах. В этой игре количество фильмов, которые связывают двух актеров, называется их «степенью разделения»

**ПЕРВЫЙ ОТБОРОЧНЫЙ ЭТАП
ВСЕСИБИРСКОЙ ОТКРЫТОЙ ОЛИМПИАДЫ ШКОЛЬНИКОВ ПО ИНФОРМАТИКЕ**
4 ноября 2012 года

Для примера рассмотрим двух актеров А и В. Если они вместе снимались в одном фильме, то их степень разделения равна 1 — их связывает один фильм. Однако, если они никогда не снимались ни в одном фильме, но актер А появляется в фильме М1 вместе актером С, и В снимался в другом фильме М2 с тем же актером С, то в этом случае их степень разделения равна двум, т.к. их связывают два фильма.

Ваша задача – написать программу, которая для двух заданных актеров по списку фильмов с перечисленными занятыми в них актерами вычисляет их степень разделения.

Входные данные

В первой строке входного файла записаны без пробелов две буквы латинского алфавита, которые обозначают актеров, для которых нужно вычислить степень разделения. Следующая строка содержит целое число N — количество фильмов ($1 \leq N \leq 20$).

Далее в N строках задана информация о каждом фильме. Это последовательность больших латинских букв, записанных без пробелов, которые кодируют актеров, занятых в соответствующем фильме. Два различных актера кодируются различными буквами. Количество актеров, занятых в одном фильме может быть от 2 до 26, включительно.

Выходные данные

В выходной файл нужно вывести одно целое число — степень разделения заданных актеров. Если перечисленные файлы не связывают указанных актеров, то вывести 0.

Примеры

<i>input.txt</i>	<i>output.txt</i>
СХ 4 ABCD EFGH XYZW ZHA	3

Задача 5. Треугольники

Рассмотрим треугольник бесконечного размера, разбитый на более мелкие треугольники, как показано на рисунке. Два маленьких треугольника считаются соседними, если они имеют одну общую сторону (но не общую вершину). Поскольку каждый треугольник имеет три стороны, то у него может быть не более трех соседей.

По заданному номеру треугольника сравнительно легко можно определить визуально номера его соседей. Например, для треугольника с номером 13 соседними являются треугольники, имеющие номера 7, 12 и 14.

Ваша задача написать программу, которая по заданному номеру треугольника вычисляет сумму номеров треугольников, соседних с ним.

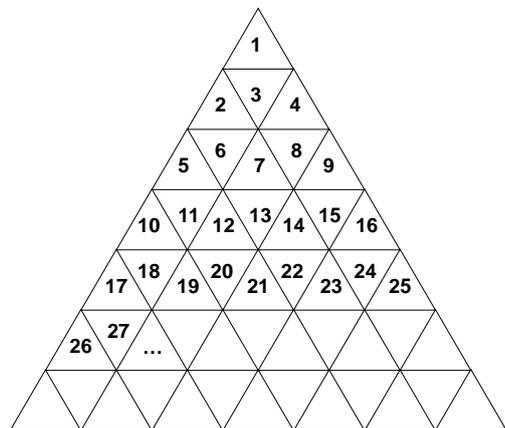
Входные данные

Во входном файле записано одно целое число N — номер треугольника ($1 \leq N < 10^8$).

Выходные данные

В выходной файл нужно вывести сумму номеров треугольников, соседних с треугольником, имеющим номер N .

Пример



<i>input.txt</i>	<i>output.txt</i>
13	33