

Задача А. Diccionario

Имя входного файла: `diccionario.in`
Имя выходного файла: `diccionario.out`
Ограничение по времени: 2 секунды

Portuñol — это особый язык, который появился в Латинской Америке. Поскольку почти половина Латинской Америки говорит на португальском (Português) и почти половина — на испанском (Español), то, естественно, произошло смешивание языков.

Каждое слово в Portuñol может быть получено путем конкатенации непустого префикса португальского слова и непустого суффикса испанского слова. Префикс слова — это любое слово, которое может быть получено путем отбрасывания нуля или более символов в конце слова. Суффикс слова — это любое слово, которое может быть получено путем отбрасывания нуля или более символов в начале слова. Даже само название языка является объединением префикса слова Português (Portu) и суффикса слова Español (ñol).

Конечно, не все возможные варианты объединения двух слов приводят к чему-то осмысленному или даже произносимому, но это не столь важно.

Вам будет предоставлено два непустых списка слов. Первый список будет содержать слова на португальском языке, а второй — на испанском. Ваша задача — написать программу, которая подсчитает количество различных слов в языке Portuñol, которые могут быть построены с помощью описанных выше правил.

Формат входного файла

Первая строка каждого теста содержит пару чисел P и S ($1 \leq P, S \leq 1000$) — количество португальских и испанских слов, соответственно. Далее следуют P строк, которые содержат слова португальского языка, по одному в строку. Последующие S строк содержат слова португальского языка, по одному в строку. Слово представляет собой строку, каждый символ которой может быть одной из 26 строчных латинских букв (от **a** до **z**).

Гарантируется, что в одном тесте не будет двух совпадающих друг с другом португальских слов, а также что сумма длин всех португальских слов не будет превышать 10^5 . Аналогичными свойствами обладает и список испанских слов.

Формат выходного файла

Необходимо вывести единственное число — количество различных слов на языке Portuñol, которые могут быть получены конкатенацией непустого префикса слова из первого списка и непустого суффикса слова из второго списка.

Примеры

diccionario.in	diccionario.out
3 3 mais grande mundo mas grande mundo	182
1 5 a aaaaa aaaaaa aaaaaaa a aaaaaaaaa	9
1 1 abc abc	8

Примечание

Обратите внимание, что если одно и же слово может быть построено разными способами, учитывать его нужно только один раз. Также обратите внимание, что входные списки слов являются всего лишь тестами для вашей программы и не обязательно будут являться списками настоящих слов из словарей португальского или испанского языков.

Задача В. ДНК

Имя входного файла: `dna.in`
Имя выходного файла: `dna.out`
Ограничение по времени: 2 секунды

Наконец-то марсоходу *Curiosity* удалось обнаружить вещества, похожие на ДНК марсиан! Известно, что в состав ДНК человека входят четыре нуклеиновых основания: аденин (A), гуанин (G), цитозин (C) и тимин (T). Однако согласно поступившим с космического аппарата данным, ДНК инопланетян устроены сложнее: специальный анализатор выделил в составе марсианской макромолекулы 26 различных нуклеотидов, которым были присвоены обозначения — прописные латинские буквы от A до Z включительно. Молекула марсианской ДНК в своей первичной структуре имеет вид цепочки нуклеотидов, которую удобно представить как строку длины N . Учёные подозревают, что разнообразие структурных элементов не имеет большого значения для сохранения генетической информации, а важнейшую роль в кодировании белков играют отрезки ДНК, которые содержат ровно два различных нуклеотида. Помогите специалистам выделить из строки символов, описывающей ДНК, подстроку максимальной длины, в которой встречается ровно два различных символа.

Формат входного файла

В первой строке записано целое число N ($1 \leq N \leq 1\,000\,000$) — длина строки. Во второй строке записаны N прописных букв латинского алфавита без пробелов. Гарантируется, что среди них есть по крайней мере два различных.

Формат выходного файла

Выведите одно целое число — длину искомой подстроки.

Примеры

<code>dna.in</code>	<code>dna.out</code>
9 ACCCGGTC	5

Задача С. Поиск файлов

Имя входного файла: `file.in`
Имя выходного файла: `file.out`
Ограничение по времени: 4 секунды

В операционной системе вашего компьютера есть возможность поиска текстового файла по его содержанию. Содержимое каждого файла представляет собой непустую строку из строчных латинских символов.

Для осуществления поиска нужно указать системе ключ, который также должен быть непустой строкой из строчных латинских букв. В результате поиска система выдаст список всех файлов, содержимое которых включает указанный ключ в качестве подстроки.

Строка S является подстрокой строки T , если она содержит все символы из S в непрерывной последовательности. Например, строки `foofoo`, `cafoo`, `foota` и `foo` содержат строку `foo` в качестве подстроки, тогда как строки `foa`, `fofo`, `oo` и `oof` — не содержат.

Внезапно вам стало очень интересно узнать, каждое ли подмножество файлов на вашем компьютере можно найти. Подмножество файлов можно найти, если существует хотя бы один ключ, в результате поиска по которому система вернет вам данный список файлов.

Дано описание содержимого вашего компьютера. Ваша задача — посчитать количество непустых подмножеств файлов, которые можно найти.

Формат входного файла

Первая строка каждого теста содержит целое число F — количество файлов ($1 \leq F \leq 60$). Последующие F строк описывают содержимое каждого из файлов, по одному в строку. Содержимое файла представляет собой непустую строку из не более чем 10^4 символов. Каждый символ файла является одной из 26 строчных латинских букв.

Формат выходного файла

Необходимо вывести единственное число — количество непустых подмножеств файлов, которые можно найти.

Примеры

<code>file.in</code>	<code>file.out</code>
6 form formal malformed for man remake	11
3 cool cool old	3

Задача D. Лягушка-путешественница 2

Имя входного файла: frog2.in
Имя выходного файла: frog2.out
Ограничение по времени: 15 секунд

Лягушка-путешественница очень любит знакомиться с миром, передвигаясь на гусях-лебедях. Она уже не раз облетала окрестности и вполне уверена, что не найдёт больше ничего интересного, кроме n болот, расположенных в линию на расстоянии ℓ вёрст между соседними, причём живёт она на краю этого скучного мира, то есть в одном из двух самых удалённых друг от друга болот. Поэтому очередной отпуск Лягушка решила провести, наслаждаясь любимыми местами и дружным кваканьем со старыми знакомыми. При этом она собирается пролететь m раз от своего болота к любимому, самому далёкому от своего, болоту и обратно и хочет посетить каждое из n болот, кроме разве что одного, хотя бы по одному разу. При этом гуси-лебеди не могут преодолевать расстояние более 3ℓ вёрст за один перелёт, поэтому Лягушка будет вынуждена делать остановки достаточно часто, но всё же не собирается менять направление движения в любом болоте, кроме своего и любимого.

И вдруг Лягушке стало интересно, как долго она сможет проводить отпуск подобным образом, если она не хочет, чтобы один был похожим на другой. Похожими она будет считать два отпуска, если порядок остановок между перелётами полностью совпадает.

Формат входного файла

Единственная строка содержит два целых числа m ($1 < m \leq 10$) и n ($1 < n \leq 10^{12}$) — количество двухсторонних перелётов и количество болот.¹

Формат выходного файла

Единственная строка должна содержать остаток от деления числа способов провести отпуск на 1 000 000 000.

Примеры

	frog2.in	frog2.out
	4 4	65535

¹При подготовке условий личного тура автором были перепутаны ограничения на n и m . Когда эту оплошность обнаружили, условия уже были розданы... Для случайно появившейся новой задачи в экстренном порядке были подготовлены решения и наборы входных данных, но автор посчитал, что новая задача достаточно отличается от исходной, и настоял включить оригинал в командный тур.

Задача Е. Острова

Имя входного файла: islands.in
Имя выходного файла: islands.out
Ограничение по времени: 2 секунды

Одно небольшое малоизвестное государство размещается на N островах, омываемых бескрайними водами Тихого океана. На трёх островах расположены три больших города — крупные промышленные узлы государства. Правительство разрабатывает проект строительства автомобильных мостов между некоторыми парами островов, чтобы обеспечить надёжную транспортную связь между городами. Цель — сделать возможным беспрепятственный проезд на автомобиле из любого города в любой другой. Инженеры разработали M проектов мостов между парами островов и оценили стоимость реализации всех проектов (каждая пара островов рассматривалась не более одного раза, причём строительство кольцевых мостов из острова в него же также по понятным причинам никого не интересует). Мосты предполагают двухстороннее движение транспорта по ним. Строить все мосты дорого, да и нет необходимости: достаточно связать лишь три острова, на которых располагаются города, возможно не напрямую, а через другие острова. Какой минимальный объём финансирования потребуется для этого?

Формат входного файла

В первой строке входного файла записаны через пробел целые числа N ($3 \leq N \leq 1000$) и M ($1 \leq M \leq 10\,000$) — соответственно количество островов и количество мостов, которые можно построить. Острова имеют номера от 0 до $N - 1$. В последующих M строках описываются мосты: каждая строка содержит три числа A , B ($0 \leq A, B < N$) и K ($1 \leq K \leq 1\,000\,000$), которые говорят о том, что строительство моста между островами A и B стоит K у. е. В последней строке записано три различных числа X , Y и Z ($0 \leq X, Y, Z < N$) — номера островов, между которыми следует организовать возможность сухопутного перемещения.

Формат выходного файла

Выведите ровно одно число — минимальную сумму (в у. е.), которую потребуется потратить на строительство мостов. Гарантируется, что решение существует.

Примеры

islands.in	islands.out
3 2 0 1 1 1 2 1 0 1 2	2
4 4 0 1 4 1 2 5 1 3 1 3 2 2 0 1 2	7

Примечание

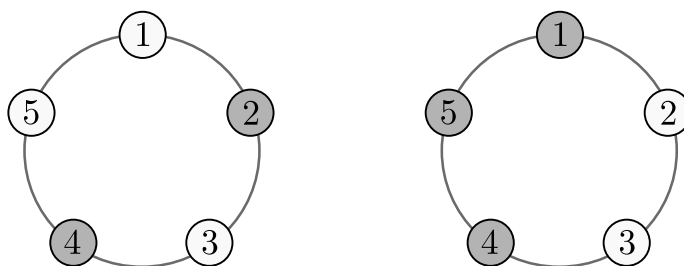
Во втором примере следует построить первый, третий и четвёртый мосты, потратив при этом $4 + 1 + 2 = 7$ у. е.

Задача F. Лампочки

Имя входного файла: `lamps.in`
Имя выходного файла: `lamps.out`
Ограничение по времени: 2 секунды

Имеется N лампочек, расположенных по кругу и пронумерованных числами от 1 до N по часовой стрелке. Изначально некоторые лампочки включены, а некоторые выключены. Имеется N переключателей; i -й переключатель позволяет одновременно изменить состояние трёх лампочек: i -й лампочки и двух соседних с ней — на противоположное состояние, а именно лампочку, которая не горит, зажечь, а светящуюся лампочку погасить.

К примеру, пусть изначальная конфигурация соответствует рисунку слева: лампочки 1, 3 и 5 светятся, а лампочки 2 и 4 погашены.



Воздействие на первый переключатель приведёт к тому, что лампочки 5 и 1 будут выключены, а лампочка 2 загорится (см. рисунок справа).

Ваша задача — путём минимального количества переключений добиться того, чтобы все лампочки горели.

Формат входного файла

В первой строке входного файла содержится целое число N ($4 \leq N \leq 10\,000$). В последующих N строках задано состояние всех лампочек: если в строке $i + 1$ записано слово `On`, то i -я лампочка изначально включена, а если там записано слово `Off`, то лампочка выключена.

Формат выходного файла

Выведите в выходной файл единственное число — минимальное количество переключений, которые требуется выполнить, чтобы зажечь все лампочки. Если это сделать невозможно, выведите строку `Epic fail`.

Примеры

<code>lamps.in</code>	<code>lamps.out</code>
4 On Off On Off	2
6 On On On Off On On	Epic fail

Примечание

В первом примере можно сначала воспользоваться вторым, а затем четвёртым переключателем.

Задача G. Полицейские и воры

Имя входного файла: `police.in`
Имя выходного файла: `police.out`
Ограничение по времени: 3 секунды

Дорожная сеть Байтландии насчитывает M автострад, проложенных между N городами. Каждая автострада допускает двустороннее движение, однако нет возможности переехать с одной автострады на другую вне городов. Каждую пару городов связывает не более одной автострады.

Увы, Байтландию захлестывает вал преступности, и лица, совершившие преступление в одном городе, обычно успевают скрыться в другом до тех пор, пока их смогут поймать по горячим следам. . .

В «светлых головах» из Министерства внутренних дел Байтландии созрела мысль: построить на некоторых автострадах пропускные пункты. Каждый такой пункт может после совершения преступления активизироваться и ограничить движение по соответствующей автостраде, выполняя досмотр всех проезжающих автомобилей.

Однако организация, оснащение и функционирование каждого такого пункта — дело хлопотное и дорогостоящее. Поэтому специалисты аналитического отдела Департамента дорожной полиции обратились к Вам за помощью. Конечно, секретных материалов о возможных местах совершения преступления, местах, где могут скрыться преступники и плане расстановки пропускных пунктов Вы не получите — не тот уровень допуска. . . Вместо этого Вам предлагается ответить на один из двух вопросов:

- может ли преступник добраться из города A в город B , минуя досмотр, если пропускной пункт активизирован на дороге, соединяющей города C и D ;
- может ли преступник добраться из города A в город B , минуя досмотр, если пропускные пункты активизированы на всех дорогах, ведущих в город C .

Попытайтесь ответить на такие вопросы.

Формат входного файла

Первая строка содержит значения N и M ($2 \leq N \leq 100\,000$, $1 \leq M \leq 500\,000$). Далее следуют M строк, каждая из которых описывает одну автостраду и содержит два числа X и Y ($1 \leq X, Y \leq N$, $X \neq Y$). Пары (X, Y) и (Y, X) не присутствуют во входном файле одновременно.

Далее следуют строка со значением Q ($1 \leq Q \leq 300\,000$) и Q строк, описывающих каждый запрос и содержащих четыре или пять чисел.

Первое число задаёт тип запроса (1 или 2). В случае запроса первого типа далее следуют четыре числа A, B, C, D — номера городов. Для запроса второго типа следуют три числа A, B, C — номера городов. Нумерация городов начинается с единицы.

Формат выходного файла

Выведите Q строк, каждая из которых должна содержать ответ на очередной запрос. Выведите в эту строку YES, если преступникам удастся миновать досмотр, и NO в противном случае.

Примеры

police.in	police.out
13 15	YES
1 2	YES
2 3	YES
3 5	NO
2 4	YES
4 6	
2 6	
1 4	
1 7	
7 8	
7 9	
7 10	
8 11	
8 12	
9 12	
12 13	
5	
1 5 13 1 2	
1 6 2 1 4	
1 13 6 7 8	
2 13 6 7	
2 13 6 8	

Задача Н. Свободные от квадратов

Имя входного файла: `square-free.in`
Имя выходного файла: `square-free.out`
Ограничение по времени: 4 секунды

Целое число n будем называть свободным от квадратов, если оно не делится нацело ни на один полный квадрат, больший единицы (полный квадрат — целое число x , для которого существует такое целое число y , что $x = y^2$).

Научному сообществу нужно быстро определять, является ли заданное натуральное число свободным от квадратов. Напишите программу, которая будет это делать!

Формат входного файла

Первая строка входного файла содержит одно целое число T ($1 \leq T \leq 10^3$) — количество чисел для проверки. Каждая из T последующих строк содержит по одному целому положительному числу. Все числа во входном файле не превосходят триллиона.

Формат выходного файла

Выходной файл должен содержать ровно T строк. i -я строка должна отвечать на запрос, является ли i -е число из ввода свободным от квадратов. Вывод вашей программы должен строго соответствовать формату вывода в примере. Каждая из T строк должна заканчиваться символом перевода строки.

Примеры

<code>square-free.in</code>	<code>square-free.out</code>
2	991 is square-free
991	2097151 is not square-free
2097151	

Задача I. Разбиение на команды по численности

Имя входного файла: `teams.in`
Имя выходного файла: `teams.out`
Ограничение по времени: 1 секунда

Необходимо определить, можно ли произвольно разбить N студентов на две команды, численность которых отличается не более чем в два раза, если известно, что в любой команде должны быть студенты, которые обязательно знакомы друг с другом. Круг знакомств задается матрицей A размера $N \times N$ с элементом A_{ij} , равным 1, если i -й студент знаком со студентом j , и равным 0, если i -й студент не знаком со студентом j .

Формат входного файла

Первая строка содержит количество студентов N ($1 \leq N \leq 1000$). Затем идут N строк, которые задают матрицу знакомств A (каждой строке матрицы соответствует отдельная строка, числа в строках разделены пробелами).

Формат выходного файла

Если можно разбить студентов на две команды, численность которых отличается не более чем вдвое, то первая строка содержит сообщение YES, а вторая строка содержит номера студентов, которые попали в команду с наибольшей численностью студентов (номера студентов в порядке возрастания их номеров, студенты нумеруются с единицы). Если нельзя разбить студентов на две команды, численность которых отличается не более чем в два раза, то единственная строка содержит сообщение NO.

Примеры

<code>teams.in</code>	<code>teams.out</code>
7 0 1 1 1 1 0 0 1 0 1 1 1 0 0 1 1 0 1 1 0 0 1 1 1 0 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0	NO
6 0 1 1 1 0 0 1 0 1 1 0 0 1 1 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0	YES 1 2 3 4

Задача J. Вампиры

Имя входного файла: `vampires.in`
Имя выходного файла: `vampires.out`
Ограничение по времени: 2 секунды

Ох уж эта мода на вампиров... Фильмы про вампиров, книги про вампиров, куклы-вампирики и так далее... Не будем же отставать от моды и дадим вам задачу про вампиров.

У вас есть карта тёмного и сырого подземелья, на которой указано местоположение всех вампиров. Подземелье разделено на $N \times M$ квадратов. Для каждого квадрата указано, свободен он («.»), занят стеной («#») или вампиром («V»). Вампир за одну секунду может переместиться из своего квадрата в смежный по стороне не занятой стеной квадрат (при этом два и более вампиров могут оказаться в одном квадрате). Ваша задача — найти на карте наиболее безопасное место, то есть такой свободный квадрат, что минимальное время, которое требуется любому вампиру, чтобы добраться до этого квадрата, максимально. Конфигурация подземелья такова, что в нём обитает по крайней мере один вампир, есть хотя бы один свободный квадрат, а до любого свободного квадрата за конечное время может добраться какой-либо вампир.

Формат входного файла

В первой строке входного файла заданы через пробел два целых числа N и M ($1 \leq N, M \leq 2000$). Во второй и последующих $N - 1$ строках даётся описание подземелья: каждая строка состоит из M символов «.», «#» или «V».

Формат выходного файла

Выведите одно число — минимальное время движения вампира до наиболее безопасного квадрата на поле.

Примеры

<code>vampires.in</code>	<code>vampires.out</code>
<pre>5 5 V..#. ##.V. V...## #####</pre>	<pre>3</pre>

Задача К. Червоточины

Имя входного файла: wormholes.in
Имя выходного файла: wormholes.out
Ограничение по времени: 2 секунды

В одной далёкой звёздной системе есть n планет, некоторые из этих планет соединены червоточинами (гипотетическими топологическими особенностями пространства-времени, представляющими собой в каждый момент времени «туннель» в пространстве).

На одной из планет зародился опаснейший вирус «апокалипсис». И королю этой системы Ивану III ничего не остаётся, как уничтожить все червоточины, чтобы избежать заражения всех планет. Для уничтожения червоточины нужно взорвать мегабомбу на одной из планет, ею соединяемых.

Иван III не знает, на какой из планет зародился вирус, но он точно знает, что взрыв мегабомбы уничтожит все живое на планете. Помогите королю определить минимальное число бомб, которые нужно взорвать, чтобы уничтожить все червоточины.

Формат входного файла

В первой строке входного файла записаны два целых числа n и m ($1 \leq n \leq 23$, $0 \leq m \leq \frac{n(n-1)}{2}$) — количество планет и червоточин соответственно. Далее в m строках описываются червоточины. В каждой из этих строк записаны два различных целых числа x и y ($0 \leq x, y < n$) — номера соединённых червоточиной планет (планеты нумеруются целыми числами от 0 до $n - 1$). Между парой планет может быть не более одной червоточины.

Формат выходного файла

Выведите одно целое число — минимальное число бомб, необходимое для подрыва всех тоннелей.

Примеры

wormholes.in	wormholes.out
5 4 0 1 1 2 2 3 3 4	2
5 8 0 1 0 4 1 2 1 4 2 4 2 3 3 0 3 4	3