Задача А. Шахматная мастерская

Имя входного файла:	`chess.in`
Имя выходного файла:	`chess.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Хозяин мастерской по изготовлению шахматных досок гроссмейстер Хосе Раулевич Капабланка был очень зол. «Ну кто так раскрашивает доски?! Ну скажи мне, Бобби, разве я тебя так учил раскрашивать доски?!» — спрашивал он у своего подмастерья.

А дело было вот в чем. Недавно его мастерская получила заказ на изготовление нестандартной шахматной доски размером N на M. Саму шахматную доску из дорогой породы дерева он изготовил, а раскрасить ее поля он поручил своему подмастерью и ученику Бобби. Бобби, однако, справился с этой задачей очень плохо. Он раскрасил доску так, что некоторые соседние поля оказались покрашены в один цвет. А такого на шахматной доске никогда не было и быть не может!

Теперь у Бобби есть всего одна ночь на исправление своей ошибки. Казалось бы, времени много. Но все усложняется тем, что перекрасить поле шахматной доски — достаточно сложная задача, ведь надо аккуратно снять старый слой краски. Поэтому Бобби хочет перекрасить наименьшее возможное число полей. Помогите ему — напишите программу, которая найдет какие поля доски ему надо перекрасить.

Единственным требоанием к шахматной доске, которая должна получиться в результате, является то, что любые два соседних поля покрашены в разные цвета. Ну, и, конечно, любое поле должно быть либо белым, либо черным.

Соседними полями на шахматной доске Хосе Раулевич считает поля, имеющие общую сторону.

Формат входного файла

Первая строка входного файла содержит два целых числа: N и M $(1 \le N, M \le 100)$. Далее идут N строк по M символов в каждой. j-ый символ i-ой из этих строк равен `W`, если Бобби покрасил соответствующее (т.е. находящееся на пересечнии i-ой горизонтали и j-ой вертикали)поле в белый цвет, и `B` — если в черный.

Формат выходного файла

В первой строке выходного файла выведите минимальное количество k полей, которые должен перекрасить Бобби.

Далее выведите k строк, описыающих поля, которые он должен перекрасить. Описание каждого поля должно состоять из двух чисел: i $(1 \le i \le N)$ и j $(1 \le j \le M)$, задающих горизонталь и вертикаль, на пересечении которых находится данное поле.

Ни одно поле не должно быть указано в этом списке дважды.

Может оказаться так, что $k = 0$, и Хосе Раулевич зря кричал на Бобби. Впрочем, это объясняется тем, что гроссмейстер уже весьма стар и его зрение далеко не идеально.

Примеры

chess.in	chess.out
4 4 BBBB BBBB BBBB BBBB	8 1 1 1 3 2 2 2 4 3 1 3 3 4 2 4 4
3 3 WBW BWB WBW	0

Задача B. Транспортная компания

Имя входного файла:	`company.in`
Имя выходного файла:	`company.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Транспортная компания «Crash & Co» занимается перевозкой запчастей для машин. Недавно в России открылось N отделений этой компаниии. Руководство компании приняло решение купить часть дорог между своими отделениями, чтобы обезопасить свои фуры от аварий. При этом купленных дорог должно быть по возможности минимальное количество, но они должны давать возможность добраться из любого отдления в любое другое. Вам как программисту отдела покупок было поручено в кратчайшие сроки посчитать количество различных вариантов купить требуемый набор дорог.

Формат входного файла

В первой строке входного файла записано количество отделений компании $N, 1 \leq N \leq 20$ и число различных дорог, соединяющих отдления $M, 1 \leq M \leq 5000$. В следующих M строках записаны пары различных номеров отделений $a_i, b_i, 1 \leq a_i, b_i \leq N$ соединенных дорогой.

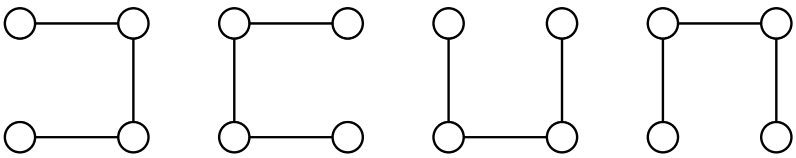
Формат выходного файла

В первой строке выходного файла запишите требуемое количество способов.

Примеры

company.in	company.out
4 4 1 2 2 3 3 4 4 1	4
4 3 1 2 2 3 3 4	1

Варианты покупки дорог для первого примера изображены на рисунке:



Во втором примере единственный удовлетворяющий условию вариант — купить все имеющися дороги.

Задача C. Китайская игра "Цзяньшидзы"

Имя входного файла:	`czjanshdzy.in`
Имя выходного файла:	`czjanshdzy.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

В древнем Китае было придумано множество интересных игр. Все знают очень известную игру «Majong». В данной задаче мы рассмотрим другую интересную китайскую игру «Цзяньшидзы». Кратко сформулируем правила игры:

Из двух куч каменей, двое играющих поочередно могут брать:

1) Произвольное ненулевое число камней из одной кучи.

2) Одновременно по одинаковому произвольному ненулевому числу камней из обеих куч.

Выигрывает тот, кто возьмет своим ходом последний камень.

Не сложно определить в каких случаях выигрывает первый, а в каких случаях выигрывает второй игрок. Вообще задача сводиться к нахождению, так называемых “проигрышных” пар (a_i, b_i), означающих, что при кучах камней содержащих a_i и b_i камней проигрывает тот, кто в данный момент. Вам предстоит решить эту задачу, но немного в другом виде.

Обозначим проигрышную, для ходящего позицию, когда в кучках a и b камней — (a, b). Так как порядок куч не играет роли, то всегда будем считать, что $a \leq b$. Упорядочим все проигрышные позиции в лексикографическом порядке, иначе говоря:

$(a_1, b_1) < (a_2, b_2)$, если $(a_1 < a_2) \vee ((a_1 = a_2) \wedge (b_1 < b_2))$.

Занумеруем проигрышные пары, начиная с 0. Ваша задача — найти k-ю пару “проигрышных” куч.

Формат входного файла

В первой строке находится число $N, (1 \leq N \leq 1000)$ - количество тестов в файле. В следующих N строках содержатся числа $k_i, (0 \leq k_i \leq 10^9)$ — порядковый номер пары “проигрышных” куч, которую требуется найти.

Формат выходного файла

В N строках выходного файла должны находится пары чисел (a_{k_i}, b_{k_i}).

Примеры

czjanshdzy.in	czjanshdzy.out
3 0 1 2	0 0 1 2 3 5

Задача D. Жизнь цвета индиго

Имя входного файла:	`indigo.in`
Имя выходного файла:	`indigo.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Мальчику Севе очень нравится цвет индиго (это такой темно-синий цвет). Сева всячески старается окружить себя вещами этого цвета.

Скоро Новый год, и Сева решил изготовить гирлянду с лампочками, чтобы украсить ей свою комнату. Он купил n патронов для лампочек и соединил $(n-1)$-им проводом так, что в гирлянде не образовалось ни одного замкнутого контура. Его гирлянда, таким образом с точки зрения математики, представляет собой дерево. Для того, чтобы гирлянда была готова, осталось совсем немного — надо только добавить провод для подключения гирлянды в электрическую сеть и вкрутить разноцветные лампочки в патроны.

У Севы есть лампочки трех разных цветов: синего, фиолетового и индиго. При этом, в некоторые патроны из соображений красоты нельзя устанавливать лампочки определенных цветов. Также, из соображений красоты, в гирлянде не должно быть двух лампочек одного цвета, напрямую соединенных проводом. Разумеется, Сева хочет, чтобы в гирлянде было как можно больше лампочек его любимого цвета.

Помогите Севе. Напишите программу, которая найдет максимальное количество лампочек цвета индиго, которые можно установить в собранную Севой гирлянду.

Формат входного файла

Первая строка входного файла содержит целое число n — количество патронов для лампочек в Севиной гирлянде ($1 \le n \le 50000$).

Последующие $(n-1)$ строка содержат каждая по два числа: u_i и v_i ($1 \le u_i, v_i \le n$) — номера патронов, соединенных соответствующим проводом.

Далее следуют n строк с описанием патронов. Кажная из них не пуста и состоит из не более, чем трех различных символов «`I`», «`B`» или «`V`». i-ая из этих строк описывает i-ый патрон, а именно: если она содержит символ «`I`», то в i-ый патрон можно устанавливать лампочку цвета индиго, «`B`» — можно устанавливать лампочку синего цвета, «`V`» — можно устанавливать лампочку фиолетового цвета.

Формат выходного файла

В выходной файл выведите единственное число: ответ на задачу. Если же при наложенных ограничениях Севе вообще не удастся собрать гирлянду, выведите число -1.

Примеры

indigo.in	indigo.out
3 1 2 2 3 IB IV IB	2

Задача E. Фруктовый сок

Имя входного файла:	`juice.in`
Имя выходного файла:	`juice.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Женя недавно купил себе новую соковыжималку. Теперь по утрам он и его братья и сестры пьют свежевыжатый фруктовый сок. А это, между прочим, очень полезно!

Недавно они поняли, что можно пить сок, выжатый не только из одного вида фруктов, как, например, апельсиновый, но и различные смеси, например, виноградно-яблочный.

В Жениной семье все очень любят сок, поэтому могут утром выпить не один стакан, причем разных видов сока. Например, его сестра Катя очень любит грейпфрутовый и апельсиновый соки. Женя, как наиболее технически грамотный человек, каждое утро занимается приготовлением соков.

Опишем подробнее, как работает соковыжималка. В нее загружаются фрукты, они проходят отжим в центрифуге, обезвоженная мякоть сбрасывается в отдельный резервуар, а сок попадает в специальную емкость.

Основная проблема состоит в том, что эту емкость иногда приходится мыть. Например, если после приготовления апельсинового сока, необходимо приготовить яблочный, то емкость надо мыть, иначе получится апельсиново-яблочный сок. Более формально, пусть сок A состоит из компонентов $a_1, \ldots, a_n$, а сок B — из компонентов $b_1, \ldots, b_m$. Сок B можно готовить после сока A, если любой из компонентов a_i является компонентом сока B (т.е. $\exists j : b_j = a_i$). В противном случае емкость для сока надо помыть.

Женя не очень любит мыть посуду, поэтому хочет мыть емкость как можно меньшее число раз. Помогите ему.

Формат входного файла

Первая строка входного файла содержит количество N различных соков, которые требуется приготовить ($1 \le N \le 300$). Каждая из последующих N строк описывает один из соков. Описание сока состоит из числа k его компонентов ($1 \le k \le 300$) и списка этих компонентов. Каждый из компонентов сока описывается словом длиной до 30 символов из строчных и прописных букв латинского алфавита. Прописные и строчные буквы различаются. Различные компоненты имеют различные названия.

Формат выходного файла

В выходной файл выведите минимальное количество раз, которое Жене придется помыть емкость для сока. Учитывайте при этом, что емкость для сока надо помыть и после приготовления последней порции сока.

Примеры

juice.in	juice.out
4 1 Apple 2 Apple Orange 1 Orange 2 Orange Pineapple	2
3 1 Apple 1 Orange 1 Mango	3

Задача F. Магараджа

Имя входного файла:	`maharaja.in`
Имя выходного файла:	`maharaja.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Магараджа — это шахматная фигура, сочетающая возможности ферзя и коня. Таким образом, магараджа может ходить и бить на любое количество клеток по диагонали, горизонтали и вертикали (т.е. как ферзь), а также либо на две клетки по горизонтали и на одну по вертикали, либо на одну по горизонтали и на две по вертикали (как конь).

Ваша задача — найти число способов расставить на доске n на n ровно k магараджей так, чтобы они не били друг друга.

Формат входного файла

Входной файл содержит два целых числа: n и k ($1 \le k \le n \le 12$).

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

maharaja.in	maharaja.out
3 1	9
4 2	20

Задача G. Красивые последовательности

Имя входного файла:	`sequence.in`
Имя выходного файла:	`sequence.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Математики — люди, весьма ценящие красоту. Они умеют находить ее даже в объектах, к которым, по мнению большинства, это понятие даже не может быть применимо.

Например, недавно один математик, занимавшийся изучением последовательностей натуральных чисел, заметил, что некоторые из последовательностей отличаются необычайной красотой.

Особенно красивыми он посчитал последовательности $a_1, \ldots, a_n$ длины n из чисел от 1 до k, обладающие следующими свойствами:

1. $a_1 = 1$;
2. если $a_i = M$, $i > 1$, то найдется $j < i : a_j = M - 1$, либо $M = 1$.

Примером такой последовательности для $n = 5$, $k = 3$ является $1, 1, 2, 3, 1$, а последовательность $1, 3, 1, 4$ не является особенно красивой ни для каких n и k.

Ваша задача состоит в нахождении числа *особенно красивых* последовательностей для заданных n и k.

Формат входного файла

Входной файл содержит два целых числа: n и k ($1 \le n, k \le 500$).

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

sequence.in	sequence.out
3 3	5
7 6	876

В первом примере искомыми последовательностями являются: $1, 1, 1$; $1, 1, 2$; $1, 2, 1$; $1, 2, 2$; $1, 2, 3$.

Задача H. Манхеттенские улицы

Имя входного файла:	`streets.in`
Имя выходного файла:	`streets.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Система улиц Нью-Йоркского района Манхеттен весьма интересна. В Манхеттене есть n улиц, идущие с запада на восток (авеню), и m улиц, идущие с севера на юг (просто улицы). Ширина каждого авеню и каждой улицы равна d метров, а длина — l метров. При этом каждая улица пересекает каждый авеню и не имеет общих точек с другими улицами, а каждый авеню пересекает каждую улицу и не имеет общих точек с другими авеню.

Разумеется, все авеню и улицы имеют асфальтовое покрытие. Дорожно-ремонтные службы интересуются, сколько квадратных метров асфальта уложено на все авеню и улицы. На перекрестках, без сомнения, асфальт уложен в один слой.

Напишите программу, вычисляющую ответ на их вопрос.

Формат входного файла

Входной файл содержит четыре натуральных числа n, m, d, l $(1 \le n, m, d, l \le 10^9, l > md, l > nd)$.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

streets.in	streets.out
1 1 5 10	75
100 10 23 4560	11007800

Задача I. Строки

Имя входного файла:	`strings.in`
Имя выходного файла:	`strings.out`
Ограничение по времени:	4 секунды
Ограничение по памяти:	64 мегабайта

Циклическим сдвигом строки s называется строка $s_k s_{k+1} s_{k+2} \ldots s_{|s|} s_1 s_2 \ldots s_{k-1}$ для некоторого k, здесь $|s|$ — длина строки s.

Подстрокой строки s называется строка $s_i s_{i+1} \ldots s_{j-1} s_j$ для некоторых i и j.

Вам даны две строки a и b. Выведите количество подстрок строки a, являющихся циклическими сдвигами строки b.

Формат входного файла

В первой строке входного файла записана строка a ($1 \le |a| \le 10^5$). Во второй строке входного файла записана строка b ($1 \le |b| \le |a|$).

Обе строки состоят только из прописных символов латинского алфавита.

Формат выходного файла

В первой строке выходного файа выведите одно целое число — ответ на задачу.

Примеры

strings.in	strings.out
abcabc abc	4
abcabc acb	0
aaaaaaa aa	6
abaaaaaba aba	4

Задача J. Странная таблица

Имя входного файла:	`table.in`
Имя выходного файла:	`table.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Вася нарисовал таблицу из N строк и M столбцов и заполнил ее по строкам целыми неотрицательными числами по следующему правилу:

1) $A_{0,0} = 0$

2) $A_{i,j} = \min\{k \in Z_+ \mid (\forall i_0 \in Z_+, i_0 < i \to A_{i_0,j} \neq k) \wedge (\forall j_0 \in Z_+, j_0 < j \to A_{i,j_0} \neq k)\}$

Иными словами, $A_{i,j}$ ести минимальное целое неотрицательное число, которое не встречалось или выше или левее в таблице.

Например, при $N = 1$ и $M = 2$ получается следующая таблица:

$$\begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 3 \end{pmatrix}$$

Теперь ему интересно посчитать сумму

$$\sum_{i=0}^{n} \sum_{j=0}^{m} A_{i,j},$$

то есть сумму элементов таблицы. Ваша задача — проверить его вычисления и написать программу, вычисляющую эту сумму.

Формат входного файла

В первой строке входного файла содржаться числа N и M $(1 \leq N, M \leq 10^9)$

Формат выходного файла

В единственной строке входного файла должна содержаться искомая суммма.

Примеры

table.in	table.out
1 1	2
1 2	7