

Problem A. Ремонт двигателя

Input file: `repair.in`
Output file: `repair.out`
Time limit: 2 секунды
Memory limit: 64 мегабайта

Во время первого рейса крупнейшего байтландского лайнера случилась поломка двигателя. Старший механик приступил к ремонту, взяв в ассистенты оказавшегося на борту инженера — представителя судостроительной компании, которая занималась постройкой лайнера.

Механик сразу же понял, в чём проблема, так что инженеру оставалось только подавать механику инструменты. Но тут возникла проблема — инженер не был знаком с морскими названиями инструментов... Поэтому работа происходит следующим образом: механик произносит несколько морских названий (количество инструментов инженер различить может) инженер выкладывает какие-то инструменты, и, если набор инструментов совпадает с требуемым, механик забирает их, производит очередное действие и возвращает на место. Если набор не совпадает, механик произносит несколько не относящихся к делу морских терминов и не берёт ничего. Инженер запоминает всю информацию об успешных и безуспешных попытках и соотносит её с услышанными названиями инструментов, после чего использует вычисленную информацию. На выкладывание одного набора инструментов (вне зависимости от количества инструментов) требуется одна секунда. Сколько времени будет затрачено на ремонт двигателя в наихудшем случае?

Input

В первой строке входного файла задано одно целое число T ($1 \leq T \leq 30$) — количество тестовых примеров. Каждый тестовый пример начинается со строки, содержащей два целых числа N — количество инструментов, и K — количество действий, требуемых механику для ремонта ($1 \leq N \leq 1000$, $1 \leq K \leq 100$). Следующая строка тестового примера содержит N морских названий инструментов. Каждое название состоит из не менее, чем одного, и не более, чем 25 символов (допускаются строчные латинские буквы и знак '-'). Последующие K строк задают наборы инструментов, требуемые для каждого из действий. В i -й из этих строк сначала задаётся число a_i — количество инструментов, который требуется для i -го действия, затем идут a_i слов ($1 \leq a_i \leq N$) — наименования инструментов (гарантируется, что все они присутствуют во второй строке и в наборе не повторяются).

Output

Для каждого тестового примера в отдельной строке выведите одно число — максимальное время, которое будет затрачено на ремонт двигателя, взятое по модулю $2^{31} - 1$.

Example

repair.in
2 5 4 hrenovina fignya figovina figo-hrenovina hreno-figovina 2 fignya hreno-figovina 2 hrenovina hreno-figovina 1 figo-hrenovina 1 figovina 3 3 figovina hrenovina progrebina 1 figovina 1 hrenovina 1 progrebina
repair.out
19 6

Problem B. Рассказ штурмана

Input file: report.in
Output file: report.out
Time limit: 2 секунды
Memory limit: 64 мегабайта

Один из штурманов торгового флота Байтландии рассказывал о том, как корабль, на котором он плавал, оказался в тумане. Ожидая улучшения погоды, корабль двигался по контуру некоторого N -угольника.

Слушатели усомнились в рассказе штурмана, тем более, что он не запомнил порядок, в котором следовали прямые участки маршрута — только длины этих участков.

Ваша задача — выяснить, мог ли теоретически существовать описанный штурманом маршрут. Участок океана, по которому двигался корабль, при решении задачи считать плоским.

Input

В первой строке входного файла задано целое число N — количество сторон N -угольника ($3 \leq N \leq 100$). Во второй строке через пробел перечислены N целых чисел L_i — длины сторон в милях ($1 \leq L_i \leq 100$).

Output

Если N -угольника с названным штурманом сторонами не существует, выведите 'NO', в противном случае выведите 'YES'

Example

report.in	report.out
3 1 1 1	YES
4 1 6 3 2	NO

Problem C. Восстановление бортовых журналов

Input file: `restore.in`
Output file: `restore.out`
Time limit: 2 секунды
Memory limit: 64 мегабайта

Во время рейсов байтландского флота капитаны фиксируют в бортовых журналах «особенные точки» маршрутов, в частности, самую северную и самую южную точку маршрута. Однако с прошествием времени некоторые бортовые журналы утрачиваются и в этом случае может возникнуть необходимость восстановить записи в журналах...

Кратчайший путь по искривленной поверхности — это не прямая, а так называемая геодезическая кривая. Поскольку поверхность Земли не плоская, то кратчайший путь между любыми двумя точками проходит не по прямому, а по геодезическому.

Вам известны широта и долгота начального и конечного пункта рейса. Эти пункты отстоят друг от друга не более, чем на $1/3$ экватора. Необходимо определить широты самой северной и самой южной точек маршрута с учётом того, что корабль идёт по кратчайшему пути. При расчете маршрута следует пользоваться предположением, что Земля имеет сферическую форму.

Input

Входной файл состоит из двух строк. Первая строка содержит два целых числа Lat_1 и $Long_1$, разделенных пробелом, — широта и долгота порта отправления корабля в градусах. Во второй строке записаны еще два целых числа, разделенных пробелом, Lat_2 и $Long_2$ — широта и долгота порта прибытия корабля в градусах ($-90 \leq Lat_1, Lat_2 \leq 90$, $-180 \leq Long_1, Long_2 \leq 180$, 0 градусов — широта экватора).

Output

В выходной файл нужно вывести через пробел два целых числа — широты самой северной и самой южной точек маршрута, в градусах, округлённые до ближайших целых.

Example

<code>restore.in</code>	<code>restore.out</code>
45 0 30 180	90 30

Problem D. Лоцманы

Input file: river.in
Output file: river.out
Time limit: 2 секунды
Memory limit: 64 мегабайта

Некоторые крупные торговые города Байтландии находятся не на берегу моря, а на крупных реках. При судоходстве по рекам мало иметь точный атлас, на который нанесены все отмели — без лоцмана, хорошо знающего свой участок, здесь даже сейчас обойтись достаточно непросто. Что уж говорить о прошлых веках...

Согласно давним традициям байтландского флота, капитан вознаграждает лоцмана не только деньгами, но и «лоцманским обедом» — дорогой сигарой, бутылкой крепкого напитка и большой копчёной рыбиной, пойманной в дальних морях. Со временем количество поднимающихся по рекам кораблей увеличивалось, так что услуги лоцманов становились всё более и более востребованными. Теперь уже не всякий обед удовлетворял лоцмана; хорошо хоть, что на каждом участке реки список «желательных» сортов табака и алкоголя и рыбы был одинаков для всех лоцманов. Разумеется, у капитанов были все эти списки.

Один корабль возвращался из дальнего торгового рейса. Капитан заранее заготовил для выплат по две сигары каждого из сортов, которые могут заинтересовать лоцманов, по две бутылки дорогого алкоголя каждого вида и по две различных копчёных рыбины всевозможных видов. Кораблю требуется пройти N последовательных участков реки, для каждого из которых известен список принимаемых лоцманами на данном участке сигар, напитков и видов рыбы. Капитан хочет выяснить, насколько далеко он сможет пройти при оптимальном распределении существующих сигар, напитков и рыбы между лоцманами (на одном участке достаточно брать на борт только одного лоцмана с данного участка).

Input

В первой строке входного файла задано одно целое число T — количество тестовых примеров ($1 \leq T \leq 100$). В первой строке каждого тестового примера задано целое число N ($1 \leq N \leq 100$) — количество участков, которые необходимо пройти кораблю. Далее следуют N групп строк по три строки в каждой. Первая строка в каждой группе задаёт список сортов сигар, принимаемых лоцманом на соответствующем участке. Строка состоит из целого числа n_s ($1 \leq n_s \leq 10$) — количества принимаемых сортов сигар и n_s строк, состоящих из не менее, чем одной и не более, чем 32 строчных и прописных английских букв — названий соответствующих сортов. Аналогично вторая строка группы задаёт список наименований допустимых на данном участке реки алкогольных напитков, третья — допустимых видов рыбы. Ограничения по всем трём группам товаров одинаковы.

Output

Для каждого тестового примера в отдельной строке выведите одно целое число — максимальное количество последовательных участков реки, которое корабль сможет пройти при оптимальном распределении выделенных для выплат лоцманам ресурсов.

Example

river.in	river.out
1	3
4	
3 Havana FidelCastro MosTabak	
2 Martini Martell	
3 Shark Ivasi Barracuda	
2 Havana FidelCastro	
2 Martell Vodyara	
1 Ivasi	
2 Havana FidelCastro	
2 Vodyara Martini	
1 Ivasi	
3 Havana FidelCastro MosTabak	
2 Martini Martell	
1 Ivasi	

Problem E. Разгрузка судна

Input file: `rollout.in`
Output file: `rollout.out`
Time limit: 2 секунды
Memory limit: 64 мегабайта

При адаптации уже известной вам из предыдущих раундов новейшей автоматизированной системы обслуживания складов к погрузке-разгрузке контейнеровозов выяснилось, что в трюме контейнеровоза (в отличие от обычного здания склада) имеются переборки и другие элементы конструкции судна, препятствующие перемещению робота.

Трюм представляет собой прямоугольное помещение $M \times N$ без внутренних стен, разделённое на клетки размером 1×1 . Некоторые из клеток заняты элементами конструкции судна, в некоторых из оставшихся клеток размещены контейнеры, причём в каждой клетке может быть не более одного контейнера.

В одной из клеток трюма, не занятых элементами конструкции или контейнерами, находится лифт. В момент времени 0 из лифта выходит робот, задача которого — загрузить все контейнеры в лифт. Робот может двигаться вправо, влево, вверх или вниз (если не мешают борта корабля или элементы его конструкции). В момент, когда робот оказывается на клетке с контейнером, он может или взять контейнер, или продолжить движение. Одновременно робот может держать только один контейнер. Проходить через клетки с колоннами робот не может ни в каком случае.

Как только робот с контейнером оказывается в клетке с лифтом, он немедленно ставит контейнер в лифт. Лифт является достаточно просторным, чтобы вместить и робота, и все контейнеры. Ставить контейнер где-либо, кроме лифта, робот не может.

На одно перемещение между клетками робот затрачивает одну секунду, временем на погрузку или выгрузку контейнеров можно пренебречь. За какое минимальное время робот сумеет загрузить все контейнеры в лифт?

Input

Первая строка входного файла содержит одно целое число — количество тестовых примеров T ($1 \leq T \leq 100$). Далее задаются тестовые примеры. В первой строке каждого тестового примера заданы два целых числа M и N — размеры трюма ($1 \leq M, N \leq 500$). Каждая из последующих M строк содержит по N символов — эти строки описывают схему трюма. i -й символ в j -й строке равен 'X', если в соответствующей клетке находится элемент конструкции судна, 'L', если в соответствующей клетке находится лифт, 'B', если в ней находится контейнер, и '.', если клетка пуста. Гарантируется, что в каждом тестовом примере ровно одна клетка занята под лифт.

Output

Для каждого тестового примера выведите одно целое число — минимальное время, за которое робот сможет загрузить все контейнеры в лифт. Если собрать все контейнеры невозможно, выведите -1 .

Example

rollout.in	rollout.out
2	12
3 3	-1
X.B	
.L.	
B.B	
3 3	
B.B	
.LX	
.XB	

Problem F. Помещения на судне

Input file: `rooms.in`
Output file: `rooms.out`
Time limit: 2 секунды
Memory limit: 64 мегабайта

При внесении судна в реестр байтландского флота записываются не только водоизмещение или параметры ходовой части, но и количество рабочих помещений в трюме судна.

Трюм судна имеет прямоугольную форму. Все переборки состоят из отрезков прямых линий. Переборки параллельны бортам судна.

Вам задан план судна, обладающий следующими свойствами:

- все помещения на судне имеют прямоугольную форму;
- каждая клетка плана соответствует либо борту или переборке, либо помещению;
- карта развёрнута так, что борта судна параллельны осям координат;
- помещения разделены одинарными переборками (то есть двойных переборок не бывает);
- борта имеют одинарную толщину (то есть не бывает переборок непосредственно рядом с бортом);
- вертикальные переборки обозначаются символами '|', горизонтальные — символами '-', пересечения переборок — символами '+';
- в помещениях на плане может быть написан некий текст, поясняющий их назначение. Текст может содержать символы от 32 до 127 включительно, кроме символов '-', '|' и '+'.

По заданному плану требуется подсчитать общее количество помещений в трюме корабля.

Input

Карта выглядит как массив из N строк по M ASCII-символов в каждой. Входной файл состоит из $1 \leq N \leq 10^4$ строк, в каждой из которых записано ровно по $1 \leq M \leq 1000$ символов. Каждый символ имеет код не менее 32 и не более 127, при этом гарантируется, что карта удовлетворяет условиям задачи.

Output

В выходной файл нужно выдать единственное число — количество помещений в трюме

Example

<code>rooms.in</code>	<code>rooms.out</code>
<pre>+-----+-----+ Machine Treasury +-----+ *??* +-----+-----+</pre>	3