

Problem A. Allies

Input file: `allies.in`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 mebibytes

У Миши и Толи M и N друзей соответственно. Из них K друзей знают Мишу и Толю одновременно. Помогите узнать, сколько друзей знают или Мишу, или Толю, но не обоих вместе.

Input

Первая строка входного файла содержит одно целое число T — количество тестов ($1 \leq T \leq 10^4$). В каждой из следующих T строк содержится три целых числа M, N, K ($0 \leq M, N \leq 10^9$, $0 \leq K \leq \min(M, N)$).

Output

Для каждого теста в отдельной строке необходимо вывести одно число — количество друзей, которые знают или Мишу, или Толю, но не обоих вместе.

Examples

<code>allies.in</code>	<code>standard output</code>
3	3
3 2 1	13
20 23 15	0
30 30 30	

Problem B. Bizons

Input file: **bizons.in**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 mebibytes

Как-то раз известная в узком кругу команда Bizons приехала в город Б. для участия в олимпиаде. Первым делом им надо было поселиться в отеле *H*, но ребята так долго были в пути, что очень проголодались. Поэтому было принято решение заглянуть в кафе по пути в *H*, но, так как участникам команды скучно ходить по одним и тем же местам, они договорились выбрать такой маршрут, в каждой точке которого они окажутся ровно один раз. По счастливой случайности у одного из участников команды в смартфоне оказалась карта города, которая представляет собой прямоугольное поле, состоящее из N строк и M столбцов. Каждая клетка этого поля принимает одно из следующих значений:

- ‘#’ — данная клетка находится в ремонте (проход по ней запрещен);
- ‘.’ — пустая клетка;
- ‘S’ — клетка, из которой ребята хотят начать свой маршрут;
- ‘H’ — клетка, в которой находится отель;
- ‘C’ — клетка, в которой располагается кафе;

Выясните, может ли команда реализовать свой план.

Input

Первая строка входного файла содержит одно целое число T — количество тестов ($1 \leq T \leq 40$). Каждый тест начинается с пары целых чисел N и M ($1 \leq N, M \leq 100$).

Каждая из последующих N строк содержит по M символов в соответствии с условием задачи, при этом i -й символ в j -й строке задаёт клетку на пересечении j -й строки и i -го столбца карты. Гарантируется, что на карте присутствует ровно один символ ‘S’, ровно один символ ‘H’ и ровно один символ ‘C’.

Гарантируется, что размер входного файла не превосходит 128К.

Output

Для каждого теста в отдельной строке выведите “Yes”, если команде Bizons удалось добраться до отеля указанным способом. В противном случае выведите “No”.

Examples

bizons.in	standard output
2	Yes
1 3	No
SCH	
3 3	
S#.	
H#.	
..C	

Problem C. Currency

Input file: `currency.in`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 256 mebibytes

Студент Илья изо всех сил старается экономить. Но в кармане у него осталось лишь 100 рублей, а до стипендии ещё далеко. В поисках легкого и быстрого заработка Илья зарегистрировался на валютном рынке «Forest». Илья внимательно прослушал несколько видеосеминаров и запомнил, что сейчас выгодно вкладываться в доллар. Илье стало любопытно, какую максимальную сумму он мог бы заработать. Для этого он выбрал две валюты: доллар США и гонконгский доллар, собрал статистику по курсам этих валют за некоторый период времени (не дольше одного месяца) и просит вас написать программу, которая поможет вычислить максимально возможную выручку за этот период в рублях.

Илья предполагает, что он начал участвовать в торгах на рынке в начале этого периода и что его стартовый капитал был равен 100 рублям. Каждый день Илья может обменивать валюты друг на друга по текущему курсу в любом количестве (в том числе и дробном). Однако специфика рынка такова, что для перевода денег из одной валюты в другую обязательно придётся совершать операцию через рубли (продавать первую валюту за рубли и затем за рубли покупать вторую валюту).

Input

Первая строка входного файла содержит одно целое число T — количество тестов ($1 \leq T \leq 10^4$). Далее заданы сами тесты.

Каждый тест в первой строке содержит одно целое число N — количество дней в прогнозе ($1 \leq N \leq 31$). Далее следуют N строк. i -я из этих строк содержит четыре вещественных числа A_i, B_i, C_i, D_i : курс, по которому Илья может купить доллар США, курс, по которому Илья может продать доллар США, курс, по которому Илья может купить гонконгский доллар и курс, по которому Илья может продать гонконгский доллар в i -й день, соответственно ($0.01 \leq A_i, B_i, C_i, D_i \leq 10^4$, $B_i \leq A_i$, $D_i \leq C_i$).

Значения заданы точно и выражаются вещественными числами не более, чем с двумя знаками после десятичной точки. Гарантируется, что сумма всех N не превосходит $1.7 \cdot 10^5$.

Output

Для каждого теста в отдельной строке необходимо вывести одно число — максимальную сумму в рублях, которую Илья мог заработать на торгах, начиная со 100 рублями в кармане, с двумя знаками после десятичной точки. Если у Ильи в последний день торгов остаётся валюта, она переводится в рубли по текущему курсу. Ещё ни один участник рынка «Forest» не становился миллиардером, а это значит, что у Ильи гарантированно не получится суммы, большей 10^9 в рублёвом эквиваленте.

Examples

currency.in	standard output
2	625.00
4	100.00
2 1 10 9	
2.75 2.75 2.2 2.1	
18 10 18 10	
7.5 7.4 9.5 9.4	
2	
2.25 0.65 1.01 0.46	
0.29 0.22 1.62 0.03	

Problem D. Delivery

Input file: `delivery.in`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 256 mebibytes

В связи с нестабильной ситуацией на рынке, программист Вася стал подрабатывать курьером. Часто Васе требуется найти нужную квартиру, в которую требуется выполнить доставку. В связи с этим Вася, вспомнив свой программистский опыт, решил сесть за код.

Вася знает параметры дома, куда ему нужно доставить продукцию — количество подъездов N , этажность дома M и номер квартиры F . На каждом этаже находится четыре квартиры. Однако часть домов спроектированы так, что в некоторых из подъездов находится магазин, соответственно, на первом этаже в таких подъездах расположено не четыре, а три квартиры, при этом Вася знает номера подъездов, где находятся магазины.

Помогите Васе вычислить номер подъезда и этаж, куда ему необходимо сделать доставку. В каждом доме квартира с номером 1 находится на первом этаже первого подъезда. Если квартира с номером i находится на j -м этаже в k -м подъезде, и квартира с номером $i + 1$ не помещается на j -й этаж, она находится на $j + 1$ -м этаже того же подъезда; если в доме j этажей, то она находится на первом этаже $k + 1$ -го подъезда. Подъезды и этажи занумерованы, начиная с единицы.

Input

Первая строка входного файла содержит целое число T — количество тестов, $1 \leq T \leq 500$. Далее следуют тесты. Первая строка каждого теста содержит целые числа N, M, F, P ($1 \leq N, M \leq 10^9$, $1 \leq F \leq \min(4MN - P, 10^{18})$, $0 \leq P \leq \min(N, 10^5)$) — количество подъездов в доме; этажность дома; номер квартиры; количество подъездов, где находится магазин, соответственно. Если $P > 0$, то тест содержит и вторую строку, в которой записаны P целых чисел a_i ($1 \leq a_1, a_2, \dots, a_P \leq N$), разделённых пробелом — номера подъездов, в которых находятся магазины.

Гарантируется, что сумма всех значений P во входном файле не превосходит $1.6 \cdot 10^6$.

Output

Для каждого теста в отдельной строке необходимо вывести два числа — номер подъезда и этаж, куда необходимо подняться Васе.

Example

<code>delivery.in</code>	<code>standard output</code>
2	1 1
6 9 1 0	2 1
6 9 36 1	
1	

Problem E. Editor

Input file: `editor.in`
Output file: `standard output`
Time limit: 8 seconds
Memory limit: 256 mebibytes

В рамках программы импортозамещения вам поручено разработать один из модулей будущего отечественного текстового редактора. Модуль должен вычислять расстояние между двумя строками. Редактор умеет выполнять две операции:

1. Удалить несколько последовательных символов из строки X .
2. Вставить несколько последовательных символов в строку X .

Стоимость каждой операции равна квадратному корню из K , где K — количество вставленных или удаленных символов. Расстояние между строками X и Y вычисляется как минимальная стоимость операций, необходимых, чтобы превратить X в Y .

Input

Первая строка входного файла содержит целое число T — количество тестов ($1 \leq T \leq 100$). Далее следуют тесты. Первая строка каждого теста содержит натуральные числа N и M ($1 \leq N, M \leq 200$) — длины строк X и Y . Следующие две строки содержат X и Y , состоящие из заглавных букв латинского алфавита.

Гарантируется, что не более, чем в 50 тестах $M + N$ превосходит 200, а в остальных $M + N$ не превосходит 20.

Output

Для каждого теста в отдельной строке необходимо вывести расстояние между X и Y с двумя знаками после десятичной точки.

Example

<code>editor.in</code>	<code>standard output</code>
1 4 4 ACDD ADGD	2.00

Note

Чтобы превратить строку “ACDD” в “ADGD”, необходимо выполнить две операции: “ACDD → ADD → ADGD”, а именно: удаление одного символа ‘C’ и вставку одного символа ‘G’. Соответственно, расстояние равно 2.

Problem F. Fairyland

Input file: `fairyland.in`
 Output file: `standard output`
 Time limit: 3 seconds
 Memory limit: 128 mebibytes

Учёный-волшебник Квазислав находится на грани великого открытия. Остался последний эксперимент: Квазислав собирается открыть портал при помощи запуска аппарата, который он пока называет просто «Чудо-Машина». Помимо волшебной пыли и стержня плутония, для катализации процессов «Чудо-Машины» необходимы несколько плазмоедов.

Перед Квазиславом расположены в ряд N заряженных плазмоедов, пронумерованных последовательными целыми числами от 0 до $N - 1$. i -ый из плазмоедов имеет заряд a_i . Квазислав должен взять некоторую группу плазмоедов, расположенных подряд. Для Квазислава важным параметром является *хаотичность* группы, определяемая как минимальный модуль разности между зарядами двух плазмоедов в этой группе.

Для того, чтобы подобрать оптимальные параметры эксперимента, Квазислав хочет проверить несколько групп плазмоедов, каждая группа характеризуется номерами её самого левого плазмоеда L_k и самого правого плазмоеда R_k . К сожалению, L_k и R_k вычисляются не так просто, так что он просит Вас о помощи. Если Вы ответите на его запросы достаточно быстро, то возможно, Вы станете первым человеком, не только вошедшим в открытый портал, но и вышедшим из него.

Input

Первая строка входного файла содержит целое число $1 \leq T \leq 10$ — количество тестов. Далее заданы сами тесты. Первая строка каждого теста содержит целое число N ($2 \leq N \leq 10^4$) — количество плазмоедов. Во второй строке содержатся N целых чисел a_i , разделённых пробелами — заряды плазмоедов ($-10^9 \leq a_i \leq 10^9$).

В третьей строке содержится число M ($1 \leq M \leq 10^5$) — количество запросов. В каждой из следующих M строк задаются параметры для вычисления L_k и R_k — пара целых чисел x_k, y_k . L_k в соответствующем запросе вычисляется по формуле $L_k = (x_k + ans_{k-1}) \bmod N$, а R_k — по формуле $R_k = (y_k + ans_{k-1}) \bmod N$, где ans_i — ответ на i -ый запрос, при этом $ans_0 = 0$.

Гарантируется, что $L_k < R_k$ и $0 \leq x_k, y_k < N$ ($1 \leq k \leq M$), сумма всех N не превосходит $5 \cdot 10^4$, сумма всех M не превосходит $5 \cdot 10^5$.

Output

На каждый запрос выведите ответ в отдельной строке.

Example

<code>fairyland.in</code>	<code>standard output</code>
2	9
6	3
1 10 2 5 -10 0	1
4	5
0 1	0
4 1	
3 2	
2 4	
3	
7 7 7	
1	
0 2	

Note

Границы запросов в первом тестовом примере: $[0, 1]$, $[1, 4]$, $[0, 5]$, $[3, 5]$.

Problem G. Godfathers

Input file: `godfathers.in`
 Output file: *standard output*
 Time limit: 2 seconds
 Memory limit: 256 mebibytes

Два главы крупных кланов дон Ромарио Магдини и дон Вито Демидетти встретились в центре затихшего города, чтобы выяснить раз и навсегда, кому он принадлежит. «Этот город тесен для нас двоих» — сказал Вито. «Давайте решим вопрос относительно мирным путём, дорогой дон», — сказал Ромарио. — «Предлагаю сыграть в игру; кто проиграет, тот уходит из города». После небольшой паузы, Вито сказал: «Я знаю как Вы, дорогой дон, хорошо играете, и даже знаю Вашу любимую игру. Предлагаю немного изменить её правила». «Хорошо», — сказал Ромарио, — «но я хожу первым».

Доны играют в следующую игру. Дано целое число, записанное в десятичной системе счисления в виде строки, возможно, с ведущими нулями. Участники делают ходы по очереди. Ход заключается в удалении из записи числа одной цифры так, что после этого действия получается либо чётное число, либо пустая строка. Если перед началом хода строка пуста или ход сделать нельзя, то игрок, который должен сделать ход, проигрывает.

Определите, кто выиграет при оптимальной игре обеих сторон, если дон Ромарио ходит первым.

Input

В первой строке ввода содержится целое число T — количество тестов ($1 \leq T \leq 200$). Далее следуют T строк, каждая из которых содержит одно целое десятичное число, состоящее не более чем из 10^4 цифр, возможно, с ведущими нулями.

Гарантируется, что размер входного файла не превосходит $1.3 \cdot 10^6$ байт.

Output

На каждый тест выведите в отдельной строке “Romario”, если выиграет дон Ромарио, или “Vito”, если выиграет дон Вито.

Examples

<code>godfathers.in</code>	<code>standard output</code>
3	Romario
1	Vito
10	Vito
11	

Problem H. Hit

Input file: `hit.in`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 mebibytes

Дан выпуклый многоугольник и несколько точек. Для каждой точки необходимо определить, принадлежит ли она многоугольнику. Точки на границе многоугольника считаются принадлежащими ему.

Input

В первой строке ввода содержится целое число T — количество тестов ($1 \leq T \leq 100$).

Первая строка каждого теста содержит два целых числа N и M — количество вершин многоугольника и количество интересующих нас точек ($3 \leq N \leq 1000$, $1 \leq M \leq 1000$), соответственно.

Каждая из последующих N строк содержит по два целых числа — координаты очередной вершины многоугольника. Вершины перечислены в порядке обхода, причём возможны варианты как «против часовой стрелки», так и «по часовой стрелке». Далее следуют M строк, содержащих список интересующих нас точек, заданных в аналогичном формате. Все координаты являются целыми числами и по модулю не превосходят 10^9 .

Гарантируется, что многоугольник не имеет самопересечений, является строго выпуклым и каждый его угол лежит в интервале от 0 до 180 градусов, сумма всех N не превосходит $7 \cdot 10^4$, а сумма всех M не превосходит $5 \cdot 10^4$.

Output

На каждый запрос выведите в отдельной строке “`Yeap`”, если точка принадлежит многоугольнику или его границе и “`Nope`” в противном случае.

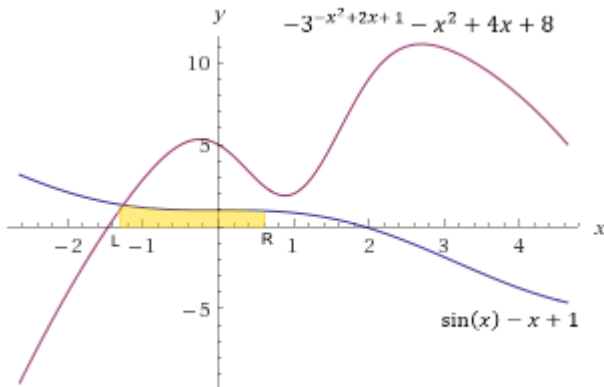
Example

<code>hit.in</code>	<code>standard output</code>
2	Yeap
4 3	Yeap
1 1	Nope
1 -1	Nope
-1 -1	
-1 1	
0 0	
0 1	
0 2	
3 1	
1 0	
1 1	
0 0	
0 1	

Problem I. Integral

Input file: `integral.in`
 Output file: `standard output`
 Time limit: 2 seconds
 Memory limit: 256 mebibytes

На плоскости задана функция $f(x) = \min(-3^{-x^2+2x+1} - x^2 + 4x + 8, \sin(x) - x + 1)$.



Найдите площадь фигуры, ограниченной сверху графиком функции $f(x)$, снизу — осью OX , а слева и справа — заданными границами L и R . Абсолютная погрешность (т.е. разница между вычисленным приближенным значением и точным значением) для площади должна быть менее 10^{-6} .

Input

Первая строка входного файла содержит одно целое число T — количество тестов ($1 \leq T \leq 100$). В каждой из следующих T строк содержится два вещественных числа L и R ($L < R$, $-10^8 \leq L, R \leq 10^8$).

Output

Для каждого теста в отдельной строке вывести значение площади фигуры с точностью до шести знаков после запятой.

Example

<code>integral.in</code>	<code>standard output</code>
3	1.994271
-10 0.5	1.169255
0.25 10	2.915947
-10 10	

Problem J. Joining

Input file: `joining.in`
 Output file: `standard output`
 Time limit: 2 seconds
 Memory limit: 256 mebibytes

Сергей Александрович собирается организовать на втором курсе группу со специальным учебным планом, базирующимся на методике МФТИ и ориентированным на спортивное программирование. За время летних каникул студенты принесли Сергею Александровичу коллективные заявления на перевод в эту группу или перевод из неё. Сергей Александрович поручил Саше составить список студентов, которые в итоге будут переведены. Саша решил взяться за работу по-научному.

Так как всего студентов, фигурирующих в заявлениях, не более 26, Саша обозначил студентов заглавными латинскими буквами. Авторов одного заявления он объединил в группы. Каждая группа состоит из некоторого (возможно, пустого, если всех авторов заявления уже успели отчислить) множества студентов и записывается как список соответствующих студентам букв, заключённый в фигурные скобки. Действия, указанные студентами в заявлениях, он обозначил следующим образом.

Пусть даны группы $g1$ и $g2$. Тогда операции создают новую группу по следующим правилам:

- $(g1+g2)$: операция '+' между группами означает, что результат — объединение групп, т.е. в итоговый список включаются все те студенты, которые имеются или в $g1$ или в $g2$.
- $(g1*g2)$: операция '*' между двумя группами означает, что результат — пересечение групп, т.е. только те студенты, которые находятся как в $g1$, так и в $g2$.
- $(g1-g2)$: операция '-' между группами означает, что результат — студенты группы $g1$, за исключением тех, которые перечислены в группе $g2$.
- Операции имеют приоритет: приоритет операции '*' выше, чем у '+' и '-'. Операции с равным приоритетом выполняются слева направо (т.е. имеют левую ассоциативность).
- Круглые скобки '(' и ')' используются для группировки операндов выражений аналогично тому, как они это делают в арифметических выражениях.

В результате у Саши получилась своеобразная формула. Ваша задача — по формуле вычислить итоговый состав новой группы.

Input

Первая строка входного файла содержит одно целое число T — количество тестов ($1 \leq T \leq 50$). Каждая из последующих T строк непуста и содержит формулу. Длина формулы не превышает 160 символов. В записи формулы использованы круглые и фигурные скобки, знаки '+', '*', '-' и большие латинские буквы.

Гарантируется, что запись формулы корректна в смысле условий задачи (то есть что в фигурных скобках содержится только какое-то (возможно, нулевое) количество заглавных латинских букв, заглавные латинские буквы не встречаются вне фигурных скобок, знаки действий стоят только между закрывающей и открывающей скобками (круглой или фигурной), формула, записанная в круглых скобках, является корректной формулой).

Гарантируется, что объём входного файла не превосходит 1500 байт.

Output

Для каждого теста в отдельной строке вывести буквы, которыми обозначены студенты, которых следует перевести в новую группу. Буквы вывести в алфавитном порядке и заключить в фигурные скобки.

Examples

joining.in	standard output
6	{ABC}
{ABC}	{ABCDEFGZ}
{ABC}+{DEFG}+{Z}+{}	{AB}
{ABE}*{ABCD}	{ABD}
{ABCD}-{CZ}	{ABCE}
{ABC}+{CDE}*{CEZ}	{CE}
(({ABC}+{CDE}))*{CEZ}	

Problem K. Kid

Input file: kid.in
Output file: *standard output*
Time limit: 5 seconds
Memory limit: 256 mebibytes

Кроха сын к отцу пришел, и сказала кроха: «Нам тут задачку по математике задали, помоги решить. Известно, что половина натурального числа является точным квадратом, а треть — точным кубом. Найти наименьшее натуральное число, удовлетворяющее этим условиям». Отец решил эту задачку и задумался — а нельзя ли решить и более общую задачу: для двух натуральных взаимно простых чисел A и B найти такое наименьшее натуральное число N , что $N/A = X^A$ и $N/B = Y^B$, где X и Y — натуральные числа.

А вы сможете решить такую задачу?

Input

Первая строка входного файла содержит одно целое число T — количество тестов ($1 \leq T \leq 10^4$). В каждой из следующих T строк содержатся два взаимно простых целых числа A и B ($2 \leq A, B \leq 10^4$).

Output

Для каждого теста в отдельной строке необходимо вывести остаток от деления наименьшего числа N , являющегося ответом к задаче, на $10^9 + 7$.

Example

kid.in	standard output
4	648
2 3	4608
9 2	573245882
9 10	907923460
10000 9999	