Problem A. Египетские дни рождения

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 second(s)
Memory limit:	512 MiB

В недавней археологической экспедиции в Египте Василий Ильич и его коллеги нашли каменную плиту, на которой было много похожих записей. По ряду причин археологи решили, что это — списки рабов, задействованных на строительстве пирамид: их имена, дни рождения, родные города. Василий Ильич очень интересуется, как же надсмотрщики различали рабов, если у них совпадали все эти параметры. Для начала он хочет определить, сколько среди записей на плите различны. Помогите ему.

Input

Первая строка содержит число n - количество записей на плите. Далее n строк, содержащих записи. Каждая строка состоит из маленьких латинских букв и не является пустой. Гарантируется, что суммарная длина строк не превосходит $5 \cdot 10^6$.

Output

Выведите количество различных записей.

Examples

standard input	standard output
4 a aa aab aa	3
3 a a a	1

Problem B. K-я строка

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

Реализуйте структуру данных, которая поддерживает следующие операции:

- добавить в словарь строку S;
- найти в словаре k-ю строку в лексикографическом порядке.

Известно, что изначально словарь пуст.

Input

Первая строка входного файла содержит натуральное число N — количество команд ($N \leq 10^5$). Последующие N строк содержат по одной команде каждая.

Команда записывается либо в виде числа k, либо в виде строки S, которая может состоять только из строчных латинских букв. Гарантируется, что при запросе k-й строки она существует. Также гарантируется, что сумма длин всех добавляемых строк не превышает 10^5.

Output

Для каждого числового запроса k выходной файл должен содержать k-ю в лексикографическом порядке строчку из словаря на момент запроса. Гарантируется, что суммарная длина строк в выходном файле не превышает 10^5.

Example

standard input	standard output
7 pushkin lermontov tolstoy gogol gorkiy 5 1	tolstoy gogol

Problem C. Шифр Бэкона

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

Программисту Васе не повезло — вместо отпуска его послали в командировку на научную конференцию. «Надо повышать уровень знаний», — сказал начальник, «Важная конференция по криптографии, проводится во Франции — а там шифровали еще во времена Ришелье и взламывали чужие шифры еще во времена Виета.»

Вася быстро выяснил, что все луврские картины он уже где-то видел, вид Эйфелевой башни приелся ему еще раньше, чем мышка стерла его с коврика, а такие стеклянные пирамиды у нас делают надо всякими киосками и сомнительными забегаловками. Одним словом, смотреть в Париже оказалось просто не на что, рыбу половить негде, поэтому Васе пришлось посещать доклады на конференции.

Один из докладчиков, в очередной раз пытаясь разгадать шифры Бэкона, выдвинул гипотезу, что ключ к тайнам Бэкона можно подобрать, проанализировав все возможные подстроки произведений Бэкона. «Но их же слишком много!» — вслух удивился Вася. «Нет, не так уж и много!» — закричал докладчик, — «Подсчитайте, и вы сами убедитесь!»

Тем же вечером Вася нашел в интернете полное собрание сочинений Бэкона. Он написал программу, которая переработала тексты в одну длинную строку, выкинув из текстов все пробелы и знаки препинания. И вот теперь Вася весьма озадачен — а как же подсчитать количество различных подстрок этой строки?

Input

На входе дана непустая строка, полученная Васей. Строка состоит только из строчных латинских символов. Ее длина не превосходит 2 000 символов.

Output

Выведите количество различных подстрок этой строки.

Examples

standard input	standard output
aaba	8

Problem D. Байтландский словарь

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 second(s)
Memory limit:	512 MiB

Байтланские учёные установили, что слова байтландского языка образуются путём конкатенации двух других (возможно, совпадающих) слов этого языка, при этом одно слово может иметь несколько вариантов происхождения. По заданному словарю языка требуется сказать, сколькими способами можно получить каждое из его слов.

Input

В первой строке входного файла находится целое число N ($1 \le N \le 10^5$) — количество слов в словаре. В следующих N строках записаны сами слова, состоящие из строчных латинских букв, их длина не превышает 50 символов. Все слова в словаре различны. Можно использовать одно слово дважды.

Output

Выведите N строк, содержащих по одному целому числу. Число в i-й строке должно означать количество способов получить i-ое слово путем соединения каких-либо двух других слов из этого же словаря.

Example

standard input	standard output
5 x xx xxx xxxx xxxxx	0 1 2 3 4

Problem E. Витя и странный урок

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 second(s)
Memory limit:	512 MiB

Сегодня на уроке Витя изучал очень интересную функцию — *mex*. *Mex* набора чисел — минимальное неотрицательное число, не присутствующее в наборе чисел. Например, $mex([4, 33, 0, 1, 1, 5]) = 2$, а $mex([1, 2, 3]) = 0$.

Витя очень быстро разобрался со всеми задачами учителя, а сможете ли вы?

Даны массив, состоящий из n неотрицательных целых чисел, и m запросов. Каждый запрос характеризуется одним числом x и заключается в следующих последовательных шагах:

- Выполнить операцию побитового сложения по модулю 2 (xor) каждого элемента массива с числом x.
- Найти mex полученного массива.

Учтите, что после каждого запроса массив изменяется.

Input

Первая строка содержит два целых положительных числа n и m ($1 \le n, m \le 3 \cdot 10^5$), обозначающие количество элементов в массиве и количество запросов соответственно.

Следующая строка содержит n неотрицательных целых чисел a_i ($0 \le a_i \le 3 \cdot 10^5$), представляющих элементы исходного массива.

Каждая из следующих m строк содержит запрос — одно неотрицательное целое число x ($0 \le x \le 3 \cdot 10^5$).

Output

Для каждого запроса выведите ответ на него в отдельной строке.

Examples

standard input	standard output
2 2 1 3 1 3	1 0
4 3 0 1 5 6 1 2 4	2 0 0

Problem F. Костюмы для актёров

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 second(s)
Memory limit:	512 MiB

Актерская труппа одно очень известного театра готовится к премьере. Они уже давно придумали сценарий, распределили роли, каждый актер уже даже выучил свой текст! Осталось разобраться только с костюмами.

Выбор подходящих костюмов - вовсе не простая задача, как может показаться. Каждый актер порой вынужден переодеваться прямо во время спектакля! Конечно, это требует времени, так что иногда успеть вовремя выйти на сцену в новом костюме гораздо труднее, чем правильно сыграть свою роль.

К счастью, в премьерной постановке каждый костюм будет состоять из не более чем 30 занумерованных от 0 до 29 частей, каждая из которых покрашена либо в белый, либо в черной цвет. Актеры заметили, что чем сильнее отличается новый костюм от уже надетого, тем дольше придется переодеваться. Чтобы как-то оценить, какие костюмы труднее всего надевать, а какие легче всего, они решили сопоставить каждому костюму свое число по следующему принципу: пусть части костюма с номерами $y_1, y_2, \ldots, y_k$ имеют черный цвет, тогда данному костюму соответствует число $2^{y_1} + 2^{y_2} + \ldots + 2^{y_k}$.

Введя таким образом соответствие между числами и костюмами, актеры заметили, что время переодевания одного костюма в другой равно битовому исключающему ИЛИ (т.е. xor-у) соответствующих им чисел. Например, если один костюм соответствует числу 6, а другой числу 5, то поменять один из них на другой актер может за $6 \oplus 5 = 3$ секунды (так как $6 = 2^2 + 2^1$, а $5 = 2^2 + 2^0$, то переодеть потребуется части с номерами 0 и 1, что займет $2^0 + 2^1 = 3$ секунды); в случае же костюмов с номерами 15 и 21 на переодевание потребуется $2^4 + 2^3 + 2^1 = 24$ секунд (ибо $15 = 2^3 + 2^2 + 2^1 + 2^0$, $21 = 2^4 + 2^2 + 2^0$).

Теперь им интересно узнать для каждого имеющегося костюма: если актер наденет этот костюм, то какое наименьшее и наибольшее время на переодевание он может потратить? Так как вся труппа очень занята репетициями, они попросили вас помочь им решить эту задачу.

Input

В первой строке находится единственное целое число n ($2 \le n \le 2 * 10^5$) — количество костюмов, имеющихся в театре.

Во второй строке через пробел перечислены n чисел, соответствующие имеющимся костюмам: $x_1, x_2, \ldots, x_n$ ($0 \le x_i < 2^{30}$).

Output

Выведите n строк, $i-$я из которых должна содержать два числа, записанных через пробел: минимальное и максимальное время, необходимое для переодевания i-го костюма.

Examples

standard input	standard output
2 58 40	18 18 18 18
3 8 7 8	0 15 15 15 0 15

Problem A. Словарь

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 second(s)
Memory limit:	512 MiB

Дан набор слов и текст, требуется определить для каждого слова, присутствует ли оно в тексте как подстрока.

Input

В первой строке дан текст (не более 10^6 строчных латинских букв). Далее дано число M — количество слов в словаре.

В следующих M строках записаны слова (не более 30 строчных латинских букв). Слова различны и отсортированы в лексикографическом порядке.

Суммарная длина слов в словаре не более 10^5.

Output

M строк вида `Yes`, если слово присутствует, и `No` иначе.

Example

standard input	standard output
trololo 3 abacabadabacaba olo trol	No Yes Yes

Problem B. Вирусы

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

Комитет По Исследованию Бинарных Вирусов обнаружил, что некоторые последовательности единиц и нулей являются кодами вирусов. Комитет изолировал набор кодов вирусов. Последовательность из единиц и нулей называется безопасной, если никакой ее сегмент (т.е. последовательность из соседних элементов) не является кодом вируса. Сейчас цель комитета состоит в том, чтобы установить, существует ли бесконечная безопасная последовательность из единиц и нулей.

Input

Первая строка входного файла `virus.in` содержит одно целое число N, равное количеству всех вирусных кодов. Каждая из следующих n строк содержит непустое слово, составленное из символов `0` и `1` — код вируса. Суммарная длина всех слов не превосходит 30 000.

Output

Первая и единственная строка выходного файла должна содержать слово:

- TAK — если бесконечная, безопасная последовательность из нулей и единиц сушествует;
- NIE — в противном случае.

Example

standard input	standard output
3 01 11 00000	NIE
3 011 11 0000	TAK

Problem C. Под-бор

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

Бором называется подвешенное дерево, на каждом из рёбер которого написано по символу, причём символы, написанные на рёбрах, выходящих из общей вершины-родителя, различны. Будем называть направление от родителя к детям "вниз". Назовем *вхождением строки s в бор* такую вершину бора, от которой можно пройти несколько шагов вниз таким образом, что встретившиеся символы образуют строку s.

Даны бор и несколько строк, найдите сумму количеств вхождений этих строк в этот бор.

Input

В первой строке входного файла записано единственное число n, $1 \le n \le 100\,000$ — количество вершин бора. В следующих n строках описаны вершины бора. В $(i+1)$-й строке описаны дети i-й вершины: число k_i ее детей, затем k_i пар из номера вершины-ребёнка и символа, написанного на соответствующем ребре. Номер родителя всегда меньше номера ребёнка; корнем бора является вершина номер 1.

В $(n+2)$-й строке записано количество m ($1 \le m \le 100\,000$) строк для поиска. В следующих m строках перечислены сами строки. Входные строки непусты, а их суммарная длина не превышает 100 000 символов.

Все символы, написанные на рёбрах, а также все символы, составляющие строки — маленькие латинские буквы.

Output

Выведите одно число — сумму количеств вхождений.

Example

standard input	standard output
7 2 2 a 4 b 2 3 a 6 b 0 1 5 b 1 7 b 0 0 4 b bb bbb bb	9

Problem D. Жужжащий профессор

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

В одном очень известном университете один очень известный профессор очень быстро произносил свои лекции, так, что ничего невозможно было разобрать. Студенты шутили по этому поводу, что он не говорит, а жужжит. Естественно, что про загадочного профессора никто абсолютно ничего не знал.

Но вот недавно Петя Булочкин решил предпринять исследование по изучению словарно- го запаса профессора. С этой целью он даже посетил одну лекцию и записал все сказанное на ней на диктофон. Затем, прокручивая дома запись с десятикратным замедлением, Петя смог записать все, что сказал профессор. Но вот незадача — профессор говорил так быстро, что, даже прослушивая замедленную запись, нельзя было точно сказать, где он делал па- узы между словами. Таким образом, у Пети есть некоторый текст S, состоящий только из маленьких латинских букв — лекция, которая была прочитана профессором.

Петя решил, что те слова, которые профессор употреблял только один раз во время своей лекции, его не интересуют. Кроме того, понятно, что если профессор употреблял некоторое слово два или более раз, то существуют два неперекрывающихся вхождения этого слова в текст S. Назовем непустую строку T кандидатом в слова, если существуют два непере- крывающихся вхождения T в S. Теперь Петя хочет найти все строки, которые являются кандидатами в слова. И поможете ему в этом Вы.

Input

Единственная строка входного файла содержит от 1 до 3 000 маленьких латинских букв. Это и есть текст S, который прочитал профессор на лекции.

Output

Единственная строка выходного файла должна содержать одно число, равное количеству строк, являющихся кандидатами в слова.

Example

standard input	standard output
bbaabbbabb	7

Problem E. Remarkable Substrings

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 second(s)
Memory limit:	512 MiB

Пусть S - строка, состоязая из строчных латинских букв. Непустая подстрока T строки S называется *замечательной* строкой степени k относительно S, если строка $k \cdot T = T + T + \ldots + T$ (т.е. T, записанная k раз) есть подстрока S.

Более формально, $S = U + k \cdot T + V$ где U и V суть некоторые (возможно, пустые) строки.

Для данной строки S, найдите максимально возможный ранг x, т.ч. существует строка T, являющаяся замечательной строка ранга x относительно S.

Input

Единственная строка - строка S ($1 \le |S| \le 10^6$).

Гарантируется, что S состоит исключительно из строчных латинских букв.

Output

Выведите одно число - ответ на задачу.

Examples

standard input	standard output
aaabc	3
abacacacacaba	4

Problem A. Выпуклый многоугольник

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

Многоугольник – замкнутая ломаная.

Простым называется многоугольник без самопересечений, самокасаний.

Выпуклым называется многоугольник, ограничивающий выпуклую область.

Выпуклой называется область A, обладающая свойством $P, Q \in A \Rightarrow [P, Q] \subseteq A$.

Input

В первой строке одно число N ($3 \le N \le 100\,000$). Далее в N строках по паре целых чисел — координаты очередной вершины простого многоугольника в порядке обхода по или против часовой стрелки.

Output

Одна строка «YES», если приведённый многоугольник является выпуклым, и «NO» в противном случае.

Examples

standard input	standard output
3 0 0 0 1 1 0	YES

Problem B. Выпуклая оболочка

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 MiB

Дано N точек на плоскости. Нужно построить их выпуклую оболочку. Гарантируется, что выпуклая оболочка является невырожденной.

Input

В первой строке число N ($3 \le N \le 10^5$). Следующие N строк содержат пары целых чисел x и y ($-10^9 \le x, y \le 10^9$) – координаты точек.

Output

В первой строке выведите N – число вершин выпуклой оболочки. Следующие N строк должны содержать координаты вершин в порядке обхода. Никакие три подряд идущие точки не должны лежать на одной прямой.

Example

standard input	standard output
3 0 0 1 0 1 1	3 0 0 1 1 1 0

Problem C. Селфи на плоскости

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 second(s)
Memory limit:	512 MiB

Для парада параллелей несколько школьников решило сделать групповое селфи. Если вдруг кто-то не знает, селфи — это такой модный вид фотографии, когда человек сам снимает себя и своих товарищей, как правило, на мобильный телефон. Школьники уже выбрали места, где будут стоять во время съёмки, и теперь пытаются определить, чьим телефоном будет сделана фотография.

В рамках данной задачи будем считать каждого школьника точкой на плоскости. Ребята решили, что фотографию стоит делать из точки, в которой находится какой-то школьник, причём угол обзора, в который помещаются все остальные точки, соответствующие школьникам, должен быть как можно меньше. Помогите им определить минимальный угол обзора на фотографии, вмещающей всех людей.

Более формально, ответ на задачу — такое минимальное неотрицательное вещественное число градусов α, что существует некоторый человек и угол величиной в α градусов с вершиной в позиции этого человека, содержащий всех людей (при этом человек может находиться на границе угла).

Длиной палки для селфи можно пренебречь.

Input

В первой строке ввода задано единственное число n ($1 \le n \le 200\,000$) — количество людей.

Далее следуют n строк, каждая из которых содержит по два числа x_i и y_i ($-10^9 \le x_i, y_i \le 10^9$) — координаты позиции i-го человека. Некоторым людям может нравиться одно и то же место, поэтому позиции некоторых людей могут совпадать.

Output

Выведите единственное вещественное число α — минимальный угол обзора в градусах.

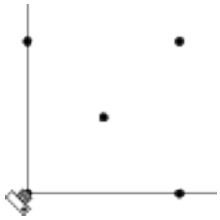
Ваш ответ будет считаться правильным, если его абсолютная или относительная погрешность относительно ответа жюри не превосходит 10^{-6}. Более формально, если ваш ответ — ans, а ответ жюри — $jury$, ваш ответ будет признан правильным, если $\frac{|ans-jury|}{\max(1,jury)} \le 10^{-6}$.

Examples

standard input	standard output
5 1 1 3 1 2 2 1 3 3 3	90.00000000000000000
3 1 0 2 1 3 2	0.00000000000000000

Note

Иллюстрация для первого примера:



Problem D. Жадный король

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

Жил-был жадный король, который однажды приказал главному архитектору окружить королевский замок стеной. Король был настолько жаден, что не желал слушать рассказы архитектора о красивой кирпичной стене с прекрасным силуэтом и изящными высокими башнями. Вместо этого он приказал окружить замок стеной, затратив минимальное количество камня и времени, но потребовал, чтобы стена не подходила к замку ближе, чем на заданное расстоние. Если король узнает, что архитектор потратил не минимально возможное количество ресурсов, то архитектор лишится головы. Более того, король потребовал, чтобы архитектор сразу же предложил проект стены с указанием минимального количества ресурсов, необходимых для постройки.

Вы должны помочь архитектору сохранить голову, написав программу для поиска минимальной длины стены, удовлетворяющей условиям короля.

Задачу упрощает то, что замок короля имеет форму многоугольника и расположен на равнине. Архитектор уже ввел систему координат и точно измерил координаты вершин замка в футах.

Input

Первая строка входного файла содержит числа N и L, разделенные пробелом. N ($3 \le N \le 1000$) — это количество вершин в королевском замке, а L ($1 \le L \le 1000$) — минимальное количество футов, на которое стена может приближаться к замку.

Следующие N строк описывают координаты замка в порядке обхода по часовой стрелке. В каждой строке через пробел записаны целые числа x_i и y_i, разделенные пробелом ($-10000 \le x_i, y_i \le 10000$), которые обозначают координаты i-ой вершины. Все вершины различны, и никакие две стороны не пересекаются кроме как по вершинам.

Output

Выведите минимальную длину стены в футах, удовлетворяющей условиям короля с точностью не менее 6 знаков после запятой.

Examples

standard input	standard output
9 100 200 400 300 400 300 300 400 300 400 400 500 400 500 200 350 200 200 200	1628.3185307180

Problem E. Внутри или вовне?

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 second(s)
Memory limit:	512 MiB

Несколько выпуклых многоугольников вложены друг в друга так, что второй многоугольник помещён в первый, третий — во второй и так далее. Каждый следующий многоугольник лежит строго внутри предыдущего (то есть контуры многоугольников не имеют общих точек). После этого стороны многоугольников стираются и остаётся только набор вершин.

Ваша задача — восстановить всю конструкцию, то есть для каждой вершины сообщить номер многоугольника, которому она принадлежит. Отметим, что точка и отрезок являются выпуклыми многоугольникам с 1 и 2 сторонами соответственно.

Input

Первая строка входа содержит одно целое число N ($1 \le N \le 2 \cdot 10^4$) — количество вершин. В i-й из последующих N строк заданы два целых числа x_i и y_i — координаты i-й точки. Координаты не превосходят 10^4 по абсолютной величине.

Output

Выведите N строк, по одной на каждую вершину. Каждая строка должна содержать одно целое число — номер выпуклого многоугольника для данной вершины (многоугольники занумерованы так, что внешний имеет номер 1, лежащий непосредственно в нём — номер 2 и так далее).

Example

standard input	standard output
5 0 0 4 4 0 4 1 1 4 0	1 1 1 2 1

Problem F. Money, Money, Money!

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

Для финансового обеспечения подготовки к Чемпионату Европы Оргкомитет открыл в Аргентинско-Ямайском банке банковский валютный счёт, номинированный в евро.

В каждый из последующих N дней Оргкомитет производит некоторые операции с евро: или получает средства от спонсоров чемпионата — тогда на валютный счёт приходит положительная сумма, или рассчитывается с субподрядчиками — тогда с валютного счёта снимается некоторая сумма (то есть «приходит отрицательная сумма»).

В конце каждого дня банк разрешает перевести всю сумму евро, находящуюся на валютном счету, на основной счёт Оргкомитета, номинированный в гривнах, по установленному условиями вклада курсу. Обратная операция условиями вклада не разрешена.

Курс обмена каждый день меняется по следующему правилу: на K-ый день один евро может быть обменян на K гривен. За каждый обмен банк забирает T гривен комиссионных. Таким образом, если в конце K-го дня на счету Оргкомитета есть S евро, и принимается решение обменять их на гривны, на основной счёт приходят $SK - T$ гривен (конечно, S может быть и отрицательным). В конце N-го дня, по завершении времени действия валютного вклада, банк принудительно конвертирует сумму (или долг) в гривны на основном счету Оргкомитета и закрывает валютный счёт.

Оргкомитет Чемпионата Европы поручил Вам максимизировать суммарное количество гривен, которое будет переведено на основной счёт Оргкомитета за время действия валютного вклада.

Input

В первой строке входного файла содержится количество тестовых случаев $Q \leq 20$. Каждый тестовый случай содержит два целых числа, N и T, разделенных одним или несколькими пробелами ($1 \leq N \leq 34567$, $0 \leq T \leq 255$). Следующая строка содержит N целых чисел $-1000 \leq A_i \leq 1000$, разделённых пробелами, обозначающих количество евро, поступающих (в случае, изменение вклада на валютном счёте в каждый из N дней соответственно.

Output

Для каждого тестового случая выведите в отдельную строку выходного файла максимально возможное количество гривен, которое можно получить при заданных условиях на валютный счёт Оргкомитета за всё время действия валютного вклада.

Example

standard input	standard output
1 7 1 -10 3 -2 4 -6 2 3	17

Problem A. Паросочетание

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 second(s)
Memory limit:	512 MiB

Двудольным графом называется неориентированный граф (V, E), $E \subseteq V \times V$ такой, что его множество вершин V можно разбить на два множества A и B, для которых $\forall (e_1, e_2) \in E$ $e_1 \in A$, $e_2 \in B$ и $A \cup B = V$, $A \cap B = \varnothing$.

Паросочетанием в двудольном графе называется любой набор его несмежных рёбер, то есть такой набор $S \subseteq E$, что для любых двух рёбер $e_1 = (u_1, v_1)$, $e_2 = (u_2, v_2)$ из S $u_1 \neq u_2$ и $v_1 \neq v_2$.

Ваша задача — найти максимальное паросочетание в двудольном графе, то есть паросочетание с максимально возможным числом рёбер.

Input

В первой строке записаны два целых числа n и m ($1 \leq n, m \leq 250$), где n — число вершин в множестве A, а m — число вершин в B.

Далее следуют n строк с описаниями рёбер — i-я вершина из A описана в $(i+1)$-й строке файла. Каждая из этих строк содержит номера вершин из B, соединённых с i-й вершиной A. Гарантируется, что в графе нет кратных ребер. Вершины в A и B нумеруются независимо (с единицы). Список завершается числом 0.

Output

Первая строка выходного файла должна содержать одно целое число l — количество рёбер в максимальном паросочетании. Далее следуют l строк, в каждой из которых должны быть два целых числа u_j и v_j — концы рёбер паросочетания в A и B соотвественно.

Examples

standard input	standard output
2 2 1 2 0 2 0	2 1 1 2 2

Problem B. Минимальное контролирующее множество

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

Требуется построить в двудольном графе минимальное контролирующее множество, если дано максимальное паросочетание.

Input

В первой строке файла даны два числа m и n ($1 \le m, n \le 4\,000$) — размеры долей. Каждая из следующих m строк содержит список ребер, выходящих из соответствующей вершины первой доли. Этот список начинается с числа K_i ($0 \le K_i \le n$) — количества ребер, после которого записаны вершины второй доли, соединенные с данной вершиной первой доли, в произвольном порядке. Сумма всех K_i во входном файле не превосходит $500\,000$. Последняя строка файла содержит некоторое максимальное паросочетание в этом графе — m чисел $0 \le L_i \le n$ — соответствующая i-й вершине первой доли вершина второй доли, или 0, если i-я вершина первой доли не входит в паросочетание.

Output

Первая строка содержит размер минимального контролирующего множества. Вторая строка содержит количество вершин первой доли S, после которого записаны S чисел — номера вершин первой доли, входящих в контролирующее множество, в возрастающем порядке. Третья строка содержит описание вершин второй доли в аналогичном формате.

Examples

standard input	standard output
3 2 2 1 2 1 2 1 2 1 2 0	2 1 1 1 2

Problem C. День рождения

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

Митя знаком с m юношами и n девушками и хочет пригласить часть из них на свой день рождения. Ему известно, с какими девушками знаком каждый юноша, и с какими юношами знакома каждая девушка. Он хочет добиться того, чтобы каждый приглашённый был знаком со всеми приглашёнными противоположного пола, пригласив при этом максимально возможное число своих знакомых. Помогите ему это сделать!

Input

Входной файл состоит из одного или нескольких наборов входных данных. В первой строке входного файла записано число наборов k ($1 \le k \le 20$). В последующих строках записаны сами наборы входных данных.

В первой строке каждого набора задаются числа $0 \le m \le 150$ и $0 \le n \le 150$. Далее следуют m строк, в каждой из которых записано одно или несколько чисел — номера девушек, с которыми знаком i-й юноша (каждый номер встречается не более одного раза). Строка завершается числом 0.

Output

Для каждого набора выведите четыре строки. В первой из них выведите максимальное число знакомых, которых сможет пригласить Митя. В следующей строке выведите количество юношей и количество девушек в максимальном наборе знакомых, разделённые одним пробелом. Следующие две строки должны содержать номера приглашённых юношей и приглашённых девушек соответственно. Числа в каждой из этих двух строк разделяются ровно одним пробелом и выводятся в порядке возрастания. Если максимальных наборов несколько, то выведите любой из них. Разделяйте вывод для разных наборов входных данных одной пустой строкой.

Example

standard input	standard output
2 2 2 1 2 0 1 2 0 3 2 1 2 0 2 0 1 2 0	4 2 2 1 2 1 2 4 2 2 1 3 1 2

Problem D. Кубики

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

Родители подарили Пете набор детских кубиков. Поскольку Петя скоро пойдет в школу, они купили ему кубики с буквами. На каждой из шести граней каждого кубика написана буква.

Теперь Петя хочет похвастаться перед старшей сестрой, что научился читать. Для этого он хочет сложить из кубиков ее имя. Но это оказалось довольно сложно сделать - ведь разные буквы могут находиться на одном и том же кубике и тогда Петя не сможет использовать обе буквы в слове. Правда одна и та же буква может встречаться на разных кубиках. Помогите Пете!

Дан набор кубиков и имя сестры. Выясните, можно ли выложить ее имя с помощью этих кубиков и если да, то в каком порядке следует выложить кубики.

Input

В первой строке вводится число N (1 <= N <= 100) - количество кубиков в наборе у Пети. Во второй строке задано имя Петиной сестры - слово, состоящие только из больших латинских букв, не длиннее 100 символов. Следующие N строк содержат по 6 букв (только большие латинские буквы), которые написаны на соответствующем кубике.

Output

В первой строке выведите "YES" если выложить имя Петиной сестры данными кубиками можно, "NO" в противном случае.

В случае положительного ответа, во второй строке выведите M различных чисел из диапазона $1 \dots N$, где M — количество букв в имени Петиной сестры. i-е число должно быть номером кубика, который следует положить на i-е место при составлении имени Петиной сестры. Кубики нумеруются с 1, в том порядке, в котором они заданы во входных данных. Если решений несколько, выведите любое. Разделяйте числа пробелами.

Examples

standard input	standard output
2 AB AAAAAB AAAAAA	YES 2 1
3 ANNY AAAAAA NNNNNN YYYYYY	NO

Problem E. Замощение доминошками

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

Дано игровое поле размера $n \times m$, некоторые клетки которого уже замощены. Замостить свободные соседние клетки поля доминошкой размера 1×2 стоит a условных единиц, а замостить свободную клетку поля квадратиком размера 1×1 — b условных единиц.

Определите, какая минимальная сумма денег нужна, чтобы замостить всё поле.

Input

Первая строка входного файла содержит 4 целых числа n, m, a, b ($1 \le n, m \le 100$, $|a| \le 1\,000$, $|b| \le 1\,000$). Каждая из последующих n строк содержит по m символов: символ '"."' (точка) обозначает занятую клетку поля, а символ '"*"' (звёздочка) — свободную.

Output

В выходной файл выведите одно число — минимальную сумму денег, имея которую можно замостить свободные клетки поля (и только их).

Example

standard input	standard output
2 3 3 2 .** .*.	5

Problem F. Мошенники

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

Студенты придумали N тем для курсовых, название каждой состоит из двух слов из заглавных букв английского алфавита. Среди студентов есть читеры, которые вместо того, чтобы придумывать собственные названия, взяли оба слова из уже придуманных их коллегами тем. При этом в качестве первого слова они взяли первое слово одной из тем, а в качестве второго второе. Вам даны все N названий в произвольном порядке, определите максимальное число читеров среди студентов.

Input

Первая строка содержит число тестов T. Каждый тест содержит N и N пар слов — названия тем. Все названия различны. $1 \le T \le 100$, $1 \le N \le 1000$, длины слов от 1 до 20.

Output

Выведите максимальное число читеров. Формат вывода смотрите в примере.

Examples

standard input	standard output
3 3 HYDROCARBON COMBUSTION QUAIL BEHAVIOR QUAIL COMBUSTION 3 CODE JAM SPACE JAM PEARL JAM 2 INTERGALACTIC PLANETARY PLANETARY INTERGALACTIC	Case #1: 1 Case #2: 0 Case #3: 0

Problem G. Покрытие путями

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 second(s)
Memory limit:	512 MiB

Задан ориентированный ациклический граф. Требуется определить минимальное количество не пересекающихся по вершинам путей, покрывающих все вершины.

Input

Первая строка входного файла содержит целые числа n и m — количества вершин и рёбер графа соответственно ($2 \le n \le 1\,000$, $0 \le m \le 10^5$). В следующих m строках содержатся по два натуральных числа — номера вершин u и v, которые соединяет ребро (u, v).

Output

В первой строке выходного файла выведите натуральное число k — минимальное количество путей, необходимых для покрытия всех вершин.

Examples

standard input	standard output
3 3 1 3 3 2 1 2	1

Problem H. Контроль за НЛО

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

В маленьком городке М начала действовать служба контроля за незаконными полетами НЛО. Первая задача службы — выяснить, сколько НЛО действует в окрестности города. Агенты службы опросили множество свидетелей и составили список случаев встречи с НЛО, произошедших за одни сутки, с указанием места и времени наблюдения. Теперь аналитики хотят понять, сколько же на самом деле было НЛО. Из данных разведки известна максимальная скорость, с которой может лететь НЛО. Аналитики просят вас узнать, какое минимальное количество НЛО могли наблюдать свидетели.

Input

В первой строке входных данных содержатся целые числа n и v — количество случаев наблюдения и максимальная скорость НЛО ($1 \le n \le 100$, $1 \le v \le 10^4$). Следующие n строк содержат описания случаев встречи с НЛО в формате "hh:mm x y", где hh:mm — время встречи, x и y — координаты точки наблюдения НЛО (для простоты будем считать, что все встречи происходили на плоскости). Координаты по модулю не превышают 1000. Скорость выражена в км/ч, координаты — в км.

Output

Выведите в выходной файл одно число — минимальное возможное количество НЛО.

Examples

standard input	standard output
4 1 12:00 0 0 13:10 0 1 14:00 1 0 15:00 1 1	2

Problem A. Range Variation Query

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 second(s)
Memory limit:	512 MiB

В начальный момент времени последовательность a_n задана следующей формулой: $a_n = n^2 \bmod 12345 + n^3 \bmod 23456$.

Требуется много раз отвечать на запросы следующего вида:

- найти разность между максимальным и минимальным значениями среди элементов $a_i, a_{i+1}, \ldots, a_j$;
- присвоить элементу a_i значение j.

Input

Первая строка входного файла содержит натуральное число k — количество запросов ($1 \le k \le 100\,000$). Следующие k строк содержат запросы, по одному на строке. Запрос номер i описывается двумя целыми числами x_i, y_i.

Если $x_i > 0$, то требуется найти разность между максимальным и минимальным значениями среди элементов $a_{x_i}, \ldots, a_{y_i}$. При этом $1 \le x_i \le y_i \le 100\,000$.

Если $x_i < 0$, то требуется присвоить элементу $a_{|x_i|}$ значение y_i. В этом случае $-100\,000 \le x_i \le -1$ и $|y_i| \le 100\,000$.

Output

Для каждого запроса первого типа в выходной файл требуется вывести одну строку, содержащую разность между максимальным и минимальным значениями на соответствующем отрезке.

Examples

standard input	standard output
7 1 3 2 4 -2 -100 1 5 8 9 -3 -101 2 3	34 68 250 234 1

Problem B. Хорошие дни

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

Билл разрабатывает новую математическую теорию, описывающую человеческие эмоции. Его последние исследования посвящены изучению того, насколько хорошие и плохие дни влияют на воспоминания людей о различных периодах жизни.

Недавно Билл придумал методику, которая описывает, насколько хорошим или плохим был день человеческой жизни с помощью сопоставления дню некоторого неотрицательного целого числа. Билл называет это число *эмоциональной значимостью* этого дня. Чем больше это число, тем лучше этот день. Билл полагает, что значимость некоторого периода человеческой жизни равна сумме эмоциональных значимостей каждого из дней периода, помноженной на минимум эмоциональных значимостей дней этого периода. Эта методика отражает то, что период, который в среднем может быть весьма неплох, бывает испорчен одним плохим днем.

Теперь Билл хочет проанализировать свою собственную жизнь и найти в ней период максимальной значимости. Помогите ему это сделать.

Input

Первая строка входного файла содержит число n — количество дней в жизни Билла, которые он хочет исследовать ($1 \le n \le 100\,000$). Оставшаяся часть файла содержит n целых чисел $a_1, a_2, \ldots, a_n$, все в пределах от 0 до 10^6 — эмоциональные значимости дней. Числа во входном файле разделяются пробелами и переводами строки.

Output

В первой строке выходного файла выведите максимальную значимость периода жизни Билла. Во второй строке выведите два числа l и r, означающие, что значимость периода с l-го по r-й день (включительно) в жизни Билла была максимально возможной.

Examples

standard input	standard output
6 3 1 6 4 5 2	60 3 5

Problem C. Биатлон

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

Во время лыжных соревнований N спортсменов стартуют с интервалом в 1 минуту. Скорость каждого лыжника на дистанции постоянна: i-й лыжник преодолевает 1 км за w_i минут. Длина трассы равна L км. Считается, что i-й лыжник обогнал j-го (совершил обгон), если он стартовал позже j-го, а пришёл к финишу раньше него. Подсчитайте суммарное число совершённых во время гонки обгонов.

Input

Первая строка входного файла содержит два целых числа N и L. Во второй строке через пробел расположены N целых чисел w_i. $1 \le N \le 500\,000$, $1 \le L \le 10^9$, $1 \le w_i \le 10^9$ при $i = 1, 2, \ldots, N$.

Output

Выведите единственное число — суммарное количество обгонов.

Examples

standard input	standard output
2 1 20 19	0
5 3 3 6 2 4 1	7

Note

Problem D. RMQ наоборот

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

Пусть нам дан массив $a_1, a_2, \ldots, a_n$. Определим $RMQ(i, j)$ как минимальное значение a_k для $k \in [i, j]$. Вам даны некоторые запросы $RMQ(i, j)$ и ответы на них, восстановите исходный массив a_i.

Input

В первой строке входных данных записаны два числа n и m ($1 \le n, m \le 100\,000$) — длина массива и количество запросов, для которых известны ответы, соответственно.

Далее следуют m строк, описывающих запросы. Каждая строка содержит три целых числа i, j и q ($1 \le i \le j \le n$, $-2^{31} \le q \le 2^{31} - 1$), означающих что $RMQ(i, j) = q$.

Output

Если подходящего массива не существует, то выведите «`inconsistent`» в единственной строке выходных данных.

В противном случае, в первой строке выходных данных должно быть записано слово «`consistent`». Во второй строке выведите соответствующий массив a_i ($-2^{31} \le a_i \le 2^{31} - 1$) длины n. Если правильных решений несколько, то разрешается вывести любое.

Examples

standard input	standard output
3 2 1 2 1 2 3 2	consistent 1 2 2
3 3 1 2 1 1 1 2 2 3 2	inconsistent

Problem E. Прибавление на отрезке

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 second(s)
Memory limit:	512 MiB

Реализуйте эффективную структуру данных для хранения массива и выполнения следующих операций: увеличение всех элементов данного интервала на одно и то же число; поиск максимума на интервале.

Input

В первой строке вводится одно натуральное число $N(1 \leq N \leq 100000)$ – количество чисел в массиве.

Во второй строке вводятся N чисел от 0 до 100000 – элементы массива.

В третьей строке вводится одно натуральное число $M(1 \leq M \leq 30000)$ – количество запросов.

Каждая из следующих M строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (m – найти максимум, a – увеличить все элементы на отрезке).

Следом за m вводятся два числа – левая и правая граница интервала.

Следом за a вводятся три числа – левый и правый концы отрезка и число add, на которое нужно увеличить все элементы данного отрезка массива $(0 \leq add \leq 100000)$.

Output

Выведите в одну строку через пробел ответы на каждый запрос m.

Example

standard input	standard output
5 2 4 3 1 5 5 m 1 3 a 2 4 100 m 1 3 a 5 5 10 m 1 5	4 104 104

Problem F. Точки и отрезки

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

Дано n отрезков на числовой прямой и m точек на этой же прямой. Для каждой из данных точек определите, скольким отрезкам она принадлежит. Точка x считается принадлежащей отрезку с концами a и b, если выполняется двойное неравенство $\min(a, b) \le x \le \max(a, b)$.

Input

Первая строка содержит два целых числа n ($1 \le n \le 10^5$) — число отрезков и m ($1 \le m \le 10^5$) — число точек. В следующих n строках записаны по два целых числа a_i и b_i — координаты концов соответствующего отрезка. В последней строке записаны m целых чисел — координаты точек. Все числа во входном файле не превосходят по модулю 10^9.

Output

В выходной файл выведите m чисел — для каждой точки выведите количество отрезков, в которых она содержится.

Example

standard input	standard output
2 2 0 5 7 10 1 6	1 0
1 3 -10 10 -100 100 0	0 0 1

Problem G. Том Сойер и его друзья

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second(s)
Memory limit:	512 MiB

Друзья Тома Сойера по очереди красят забор разными красками. Каждый из них красит несколько идущих подряд секций забора в определенный цвет, при этом используемые цвета могут повторяться. Новая краска ложится поверх старой. Для каждой краски вычислите количество секций, которые будут покрашены этой краской после того, как все друзья закончат работу.

Input

В первой строке входного файла содержатся два целых числа: N ($1 \le N \le 10^9$) и K ($1 \le K \le 50000$) — количество секций в заборе и количество различных красок соответственно.

Во второй строке содержится единственное число M ($0 \le M \le 50000$) — количество друзей Тома Сойера.

Далее следуют M строк: в i-ой строке содержится информация о работе друга, который красил забор i-ым по счету, а именно 3 целых числа c_i, l_i, r_i ($1 \le c_i \le K$, $1 \le l_i \le r_i \le N$) — номер краски, которую использовал i-й друг, номер первой и номер последней покрашенной секции соответственно.

Output

Выведите в единственную строку выходного файла K целых чисел: i-ое число должно быть равно количеству секций, покрашенных i-й краской.

Example

standard input	standard output
5 3 4 1 3 4 2 4 5 3 2 3 1 5 5	1 1 2

Problem H. Points on the plane

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 секунды
Memory limit:	512 MiB

Есть квадратная клетчатая плоскость состоящая из $n \times n$ клеток ($1 \le n \le 1000$). Изначально в каждой клетке записано значение ноль. Ваша задача — написать программу, умеющую отвечать на следующие запросы:

- ADD x y — увеличить значение в ячейке x, y на 1.
- GET x_1 y_1 x_2 y_2 — вернуть сумму значений в прямоугольнике с углами в x_1, y_1 и x_2, y_2 соответственно.

Input

В первой строке входного файла содержится два числа — n и k — размер доски и число запросов соответственно. Следующие k строк содержат сами запросы. Гарантируется, что общее число запросов не превосходит 300 000.

Output

Для каждого запроса типа GET выведите в отдельную строку одно целое число — ответ на соответствующий запрос.

Examples

standard input	standard output
5 15 ADD 1 1 ADD 2 2 ADD 3 3 ADD 4 4 ADD 5 5 ADD 1 5 ADD 2 4 ADD 3 3 ADD 4 2 ADD 5 1 GET 1 1 5 5 GET 2 1 5 5 GET 1 2 5 5 GET 2 2 4 4 GET 3 3 3 3	10 8 8 6 2

Problem I. Farmer

Input file:	*standard input*
Output file:	*standard output*
Time limit:	10 seconds
Memory limit:	512 MiB

Пришла весна и фермер решил внести удобрения на своём участке. Участок фермера имеет длину x метров и ширину y метров. Процесс идёт в течение n дней.

В течение одного дня фермер может проделывать одно из следующих действий:

1. Увеличить на w продуктивность прямоугольного участка земли со сторонами, параллельными осям координат и вершинами в (x_1, y_1) и (x_2, y_2).
2. Подсчитать общую продуктивность участка земли $(x_1, y_1) - (x_2, y_2)$

Ваша задача — ответить на все запросы из действий второго типа.

Input

Первая строка входных данных содержит два целых числа x и y ($1 \le x, y \le 1\,000$). Следующая строка содержит одно целое число n ($1 \le n \le 100\,000$) — количество дней, в течение которых проходил процесс. Каждая из последующих n строк задаёт действия в очередной день.

1. 1 x_1 y_1 x_2 y_2 w - фермер удобряет участок ($1 \le x_1 \le x_2 \le x$, $1 \le y_1 \le y_2 \le y$, $-10\,000 \le w \le 10\,000$)
2. 2 x_1 y_1 x_2 y_2 - фермер хочет узнсть продуктивность участка ($1 \le x_1 \le x_2 \le x$, $1 \le y_1 \le y_2 \le y$)

Output

Для каждого запроса второго типа выведите суммарную продуктивность участка, фигурирующего в запросе.

Example

standard input	standard output
8 8 3 1 2 2 8 8 2 1 1 1 2 2 1 2 2 2 2 2	3