

Задача А. Geogebra2

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Скорее всего все пользователи программы Geogebra для рисования геометрических фигур. Вас пригласили работать над новой программой Geogebra2. Одна из его фишек будет это векторный режим, который будет помогать решать задачи через векторные вычисления. Вам поручили продумать способ хранения вычислений.

Вектор представляет из себя направленный отрезок. Вектор можно задать через начальную точку и конечную. Основная операция, которая происходит с векторами это сложение. Если хотим сложить вектор AB с вектором CD , то должны параллельно передвинуть вектор CD , чтобы он начинался из точки B и вектор, полученный из точки A и новой точки второго вектора, будет их суммой. В общем случае, если два произвольных вектора, то результат сложения AB и CD никак короче переписать не можем. Если конец первого совпадает с началом второго, то $AB + BD = AD$. Вторая важная операция это отрицание, тогда $-AB = BA$.

Вы знаете, что самая частая операция у пользователей это сложение векторов. Вы хотите представить результат сложения векторов самым коротким образом для того, чтобы данные занимали наименьшее число памяти.

Формат входных данных

Единственная строка содержит последовательность операций пользователя. Она представляет собой непустую беспробельную строку, состоящую из имён векторов (двух заглавных латинских букв), разделённых знаками сложения и вычитания. Перед первым слагаемым может стоять минус. Количество слагаемых не превосходит 10^5 .

Формат выходных данных

Выведите минимальную строку, равную сумме векторов, согласно формату выше. Если сумма равна 0, то выведите 0. Если ответов несколько, разрешается вывести любой.

Примеры

<code>stdin</code>	<code>stdout</code>
ZA+VR	VA+ZR
PQ+QS-PS	0
AD+DB+DR+BE	AE+DR

Задача В. Производящие функции

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

В 10й параллели ЛКШ проходят много увлекательных тем. Недавно они проходили применения *FFT*, использование матриц для подсчета комбинаторных объектов. Теперь они знают такой мощный инструмент как производящие функции. Два ученика Вова и Данис решили попрактиковаться в них.

Они нашли два числа p, q . Обозначим $a_0 = p, a_1 = q$. Затем они построили линейную рекуренту $a_n = a_{n-1} + a_{n-2}$. Они начали выводить формулу подсчета n -го члена последовательности за $O(1)$. Быстро справившись с задачей, решили улучшить ее. После того как они находят a_n , они сортируют цифры a_n в порядке возрастания. То есть если $a_{n-1} = 123, a_{n-2} = 975$, то $a_n = 1098$ и после сортировки оно становится числом 189.

Теория производящих функций не позволяет находить быстро n -й член этой последовательности и просят вас о помощи. Для удобства они отсортировали цифры p, q также в порядке возрастания.

Формат входных данных

В первой строке даны три целых числа — p, q и n ($0 \leq p, q \leq 100, 0 \leq n \leq 10^{18}$). Гарантируется, что числа p и q расположены в порядке неубывания.

Формат выходных данных

Выведите n -й член последовательности.

Примеры

stdin	stdout
1 1 2	2
3 9 3	12

Замечание

Во втором примере последовательность 3, 9, 12, 12.

Задача С. Ремонтные работы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Асфальт на дороге, соединяющей въезд в Берендеевы Поляны и четвёртый комповник, окончательно пришёл в негодность, поэтому руководство базы приняло решение его заменить. Процесс замены они решили организовать следующим образом.

Дорогу планировали разбить на m сегментов равной длины, начиная с некоторой точки в пределах m метров от въезда, при этом часть сегментов могли оказаться «хорошими» и не содержать повреждений. На каждый повреждённый сегмент решено было отправить одного рабочего, который отремонтировал бы этот участок.

Дорога содержит n повреждений, i -е из которых можно представить как точку с координатой x_i (въезд на базу имеет координату 1). В первых m метрах дороги повреждений нет. Пусть целое число k ($1 \leq k \leq m$) — точка начала разбиения. Тогда сегмент под номером j , начинающийся в точке $k + (j - 1) \cdot m$, содержит все повреждения от своего начала (включительно), до начала следующего сегмента (невключительно).

Так как число рабочих ограничено, ваша задача — выбрать k так, чтобы повреждённых сегментов оказалось как можно меньше.

Формат входных данных

В первой строке находятся два целых числа m, n ($1 \leq m, n \leq 10^5$) — длина сегмента в метрах и число повреждений на дороге, соответственно.

Вторая строка содержит целые числа $x_1, \dots, x_n$ ($m < x_1 < x_2 < \dots < x_n \leq 2 \cdot 10^9$) — координаты повреждений.

Формат выходных данных

В первой строке выведите минимальное число рабочих, требующееся для починки асфальта.

Во второй строке выведите через пробел все возможные целые точки начала разбиения, на которых достигается оптимальный ответ, в возрастающем порядке.

Примеры

стандартный ввод	стандартный вывод
3 5 4 5 7 8 9	2 1
4 3 7 14 15	2 1 2 4
2 10 3 4 7 8 12 13 14 15 20 21	7 1 2

Замечание

В первом примере оптимально будет взять $k = 1$. Тогда сегменты будут такими: $[1, 4)$, $[4, 7)$, $[8, 11)$. Таким образом, только два сегмента будут содержать повреждения.

Задача D. Сложная задача

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

Сегодня у параллели 3 было занятие, на котором рассказывали про дерево отрезков. Школьники узнали, что это дерево, построенное на массиве длины n , выглядит следующим образом:

1. Корень дерева соответствует отрезку $[1, n]$.
2. Если какой-то вершине дерева соответствует отрезок $[a, b]$ длины хотя бы два, то её левому сыну соответствует отрезок $[a, mid]$, а правому — $[mid + 1, b]$, где

$$mid = \left\lfloor \frac{a + b}{2} \right\rfloor$$

$\lfloor x \rfloor$ означает округление вниз.

Ключевым свойством такого дерева является то, что произвольный отрезок можно без наложений и пропусков покрыть не очень большим количеством отрезков, соответствующих вершинам дерева. Обозначим минимальное число таких отрезков, необходимое для покрытия отрезка $[l, r]$ за $cover(l, r)$.

В конце лекции преподаватель решил проверить, насколько хорошо школьники поняли тему, и выдал им следующее задание. Он взял некоторое число n и попросил учеников посчитать сумму значений $cover(l, r)$ по всем $\frac{n(n+1)}{2}$ возможным отрезкам $[l, r]$. Так как ответ мог бы быть очень большим, он разрешил дать ответ по модулю $10^9 + 7$.

Школьники справились с заданием. А вы сможете?

Формат входных данных

В первой строке содержится единственное число t ($1 \leq t \leq 1000$) — количество наборов входных данных. Далее следует их описание.

Каждый набор описывается одной строкой, в которой находится целое число n ($1 \leq n \leq 10^{18}$) — размер массива, на котором построено дерево отрезков.

Формат выходных данных

Для каждого набора в отдельной строке выведите одно число — ответ на вопрос учителя по модулю $10^9 + 7$.

Пример

стандартный ввод	стандартный вывод
1 3	7

Задача Е. Простая задача

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 512 мегабайт

Чем занимаются школьники в старших параллелях? Правильно, пишут сложные алгоритмы. Лёша сегодня послушал лекцию по ТЧ, и теперь ему надо решать контест. Но вот незадача, Лёша очень ленив, поэтому не хочет этим заниматься, поэтому он поручил вам решение первой задачи из тура. Осталось понять, кто ленивее жюри или Лёша.

Условие задачи такое:

Дано число t ($1 \leq t \leq 1000$).

Далее на t строках вводится по два числа l_i, r_i ($1 \leq l_i \leq r_i \leq 10^{12}$). Необходимо вывести количество простых чисел p , для которых выполнено $l_i \leq p \leq r_i$.

Формат входных данных

Если вам это действительно важно, в первой строке вводится число t ($1 \leq t \leq 1000$) — количество запросов. В следующих t строках, если почему-то вам это правда важно, вводится по два числа l_i, r_i ($1 \leq l_i \leq r_i \leq 10^{12}$) — описание запросов.

Формат выходных данных

Если очень сильно хотите, выведите на i -й строке количество простых чисел p таких, что $l_i \leq p \leq r_i$.

Примеры

стандартный ввод	стандартный вывод
2 1 1 2 3	0 2
4 2 2 3 3 4 4 5 5	1 1 0 1
5 1 10 1 100 1 1000 1 10000 1 100000	4 25 168 1229 9592

Задача F. Тест на совместимость

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Маша и Петя решили узнать, насколько они подходят друг другу как пара. В гороскоп они не верят, но узнать их совместимость как-то надо. Они предположили, что совместимость зависит от того, как они генерируют случайность. Чем у них больше похожа случайность, тем более совместимы. Осталось придумать, как определить схожесть случайностей.

Каждый из них написал бинарную строку длины n , выбирая на i -й позиции случайный символ из $\{0, 1\}$. Затем они подумали найти что-то общее в их случайности и посчитали наибольшую общую подпоследовательность. С генерацией строк они справились, а вот посчитать НОП у них не хватило мощностей компьютера. Придумайте им способ посчитать НОП.

Формат входных данных

В первой строке вводится одно целое число n — размер строк ($1 \leq n \leq 10^5$).

Во второй строке вводится подряд строка Пети длины n .

В третьей строке вводится подряд строка Маши длины n .

Гарантируется, что обе строки случайны.

Формат выходных данных

Выведите длину НОП их строк.

Примеры

стандартный ввод	стандартный вывод
5 01110 00100	3
3 001 110	1

Задача Г. Экзамен

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	512 мегабайт

День X. В одном известном ВУЗе началась сессия и сегодня будет экзамен по программированию. Как и все экзамены, он проходит очно и принимают его преподаватели. Всего существует n слотов для записи на экзамен.

Все студенты готовятся по разному. Кто-то готовился заранее и готов сдавать с утра. Кто-то начал учить в ночь перед экзаменом и использует время по максимуму, чтобы сдать на тройку. Для i -го слота известно, что на него записалось a_i студентов.

Преподаватели хоть и не готовятся к экзаменам, но у них есть разные дела в течении дня, поэтому не могут принимать весь день. Известно, что в i -й слот b_i преподавателей могут принимать экзамен.

Слотов очень много и никаких задержек не должно быть, поэтому если $b_i < a_i$ для какого-то слота, то экзамен сорвется. Преподаватели более уважаемые люди, поэтому они решили некоторых студентов передвинуть на другие слоты, чтобы было выполнено условие $a_i \leq b_i$. Студентов не хочется очень сильно обижать. Если студента со слота i передвинуть на слот j , то он обидится на $|i - j|$ на преподавателей.

Преподаватели просят найти минимально возможное максимальное недовольство студентов. Помогите им.

Формат входных данных

Первая строка содержит одно целое число t ($1 \leq t \leq 10^5$) — количество наборов входных данных. Далее следуют описания наборов входных данных.

В первой строке набора вводится целое число n — количество слотов для записи ($1 \leq n \leq 10^6$).

Во второй строке набора вводится массив целых чисел $a_1, a_2, \dots, a_n$ — количество студентов на каждый слот ($0 \leq a_i \leq 10^8$).

В третьей строке набора вводится массив целых чисел $b_1, b_2, \dots, b_n$ — количество преподавателей на каждом слоте ($0 \leq b_i \leq 10^8$).

Гарантируется, что сумма n по всем наборам входных данных не превосходит 10^6 .

Формат выходных данных

Для каждого набора выведите минимальное возможное максимальное недовольство студентов.

Пример

стандартный ввод	стандартный вывод
8	0
4	-1
0 2 3 9	1
0 2 3 9	2
3	3
2 2 2	3
1 2 2	2
4	4
0 3 0 0	
1 1 1 0	
5	
3 0 3 0 0	
0 3 0 2 1	
6	
0 0 6 0 0 0	
3 0 0 0 0 3	
6	
2 2 2 0 0 0	
0 0 0 2 2 2	
7	
1 0 0 5 0 0 0	
0 3 0 0 0 3 0	
5	
1 0 0 0 0	
0 0 0 0 1	

Задача Н. Баланс в ЛКШ

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	8 секунд
Ограничение по памяти:	512 мегабайт

Программа в Летней Компьютерной Школе постоянно развивается, и каждый год в разных параллелях придумывают и тестируют разные форматы обучения. В частности, в этом году в одной из параллелей решено было дать несколько «разнобоев» — туров, содержащих задачи на различные темы.

Для их подготовки преподаватели уже успели подобрать n задач и расположить их в том порядке, в котором хотят давать их на смене. При этом каждая из задач имеет свой *уровень идейности*, для i -й по счёту задачи он равен a_i . Уровень идейности тура определяется как квадрат суммы уровней идейности задач в нём. И, наконец, уровень идейности всей линейки разнобоев — это сумма уровней идейности всех туров.

Теперь преподавателям осталось принять последнее решение: как конкретно задачи будут распределены по турам. Количество туров пока не фиксировано, но известно, что набор задач для каждого из туров должен являться **подотрезком** исходного набора задач. Естественно, каждая задача должна войти хотя бы в какой-нибудь тур, и никакая задача не может входить в два разных тура.

Преподаватели хотят сделать разнобои интересными и полезными, поэтому они хотят, чтобы общая идейность линейки была не очень большой и не очень маленькой. А именно, среди всех корректных разбиений задач на туры, отсортированных по убыванию общей идейности, они хотят взять k -е по счёту. Помогите им, найдите общую идейность этого разбиения.

Формат входных данных

В этой задаче несколько наборов входных данных. Гарантируется, что их количество не превышает 30.

Каждый набор задаётся двумя строками. В первой из них находятся два числа n, k ($1 \leq n \leq 1000, 1 \leq k \leq \min(2^{n-1}, 2000)$) — число задач и номер необходимого разбиения, соответственно.

Во второй строке набора находятся числа $a_1, a_2, \dots, a_n$ ($-10^6 \leq a_i \leq 10^6$) — уровни идейности задач в том порядке, в котором их планируют давать на смене.

После всех наборов идут два числа 0 0, записанные в отдельной строке.

Формат выходных данных

Для каждого набора входных данных в отдельной строке выведите одно число — ответ на задачу.

Пример

стандартный ввод	стандартный вывод
5 1	196
3 1 4 1 5	100
5 6	
3 1 4 1 5	
0 0	

Замечание

В первом примере максимально возможное значение общей идейности достигается, когда все задачи сгруппированы в один контекст, т.е. общая идейность равна $(3 + 1 + 4 + 1 + 5)^2 = 14^2 = 196$.

Во втором примере общие идейности разбиений, отсортированные по убыванию, выглядят так: 196, 130, 116, 110, 106, 100, 90, 70, 70, 68, 66, 62, 60, 60, 58, 52; значит, 6-е значение равно 100.

Задача I. Петя делает контекст

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

В этом году Петя впервые приехал преподавателем в ЛКШ. Уже завтра состоится его первая лекция, после которой он планирует выдать детям контекст, каждая задача в котором характеризуется целочисленной сложностью. Сейчас Петя меняет некоторые задачи с целью сделать контекст сбалансированным.

Процесс балансировки представляет собой последовательность действий, каждое из которых имеет один из четырёх типов:

1. Петя замечает, что задачи i и j на данный момент имеют **одинаковую** сложность.
2. Петя замечает, что задачи i и j на данный момент имеют **различную** сложность.
3. Петя меняет задачу под номером i на задачу **другой** сложности. При этом все предыдущие наблюдения про эту задачу могут стать неверными.
4. Пете интересно, можно ли на основании уже сделанных наблюдений сказать, что задачи i и j точно равны по сложности или точно не равны по сложности.

Помогите Пете: ответьте на интересующие его вопросы, и тогда он быстрее пойдёт спать.

Формат входных данных

В первой строке содержатся два числа n, q ($1 \leq n \leq 10^6, 1 \leq q \leq 5 \cdot 10^5$) — количество задач в контексте и число действий, соответственно.

Каждая из следующих q строк содержит описание очередного действия в одном из следующих форматов:

1. $1 \ i \ j$. Наблюдение первого типа про задачи i, j .
2. $2 \ i \ j$. Наблюдение второго типа про задачи i, j .
3. $3 \ i$. Изменение сложности задачи i .
4. $4 \ i \ j$. Вопрос Пети про задачи i, j .

Параметры действий являются целыми положительными числами и не превосходят n . Действия даются в хронологическом порядке.

Формат выходных данных

В ответ на каждый запрос четвёртого типа выведите **Yes**, если соответствующие задачи точно имеют одинаковую сложность, **No**, если они точно имеют разную сложность, и **Unknown** иначе.

Пример

стандартный ввод	стандартный вывод
4 11	Yes
1 1 2	No
4 1 2	Unknown
2 3 4	Yes
4 3 4	No
4 1 3	No
1 1 3	Unknown
4 1 3	
4 1 4	
3 1	
4 1 3	
4 1 4	

Задача J. Настройка сети

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	10 секунд
Ограничение по памяти:	512 мегабайт

Как известно, чаще всего Летняя Компьютерная школа проводится на базе отдыха Берендеевы Поляны, но иногда появляются и другие места. В частности, в этом году ЛКШ было решено провести на совершенно новой базе, на которой никто из преподавателей раньше не был. И, разумеется, приехавшие раньше всех админы сразу же принялись за настройку сети.

Всего на базе они обнаружили n вышек, вершина каждой из которых характеризуется тройкой (x_i, y_i, z_i) (координаты и высота) и может быть представлена как точка на трёхмерной плоскости. Теперь админам необходимо решить, на какие вышки они поместят 4 имеющиеся у них Wi-Fi-точки.

Система из 4 вышек называется *устойчивой*, если вершины вышек образуют прямоугольник ненулевой площади. Админы очень хотят, чтобы система получилась устойчивой. Помогите им: скажите, сколькими разными способами они могут выбрать 4 вышки для точек, чтобы в результате получилась устойчивая система.

Формат входных данных

В этой задаче несколько наборов входных данных. Гарантируется, что их количество не превышает 30.

Первая строка каждого набора содержит число n ($4 \leq n \leq 1500$).

Следующие n строк набора содержат по три числа x_i, y_i, z_i ($-10^8 \leq x_i, y_i, z_i \leq 10^8$) — координаты вершины очередной вышки. Гарантируется, что все точки в наборе входных данных попарно различны.

После всех наборов идёт число 0, записанное в отдельной строке.

Формат выходных данных

Для каждого набора входных данных выведите одно число — количество четвёрок вышек, образующих устойчивую систему.

Пример

стандартный ввод	стандартный вывод
8 0 0 1 0 1 1 1 0 1 1 1 1 6 0 1 7 1 1 9 -1 1 8 -2 1 0	2

Задача К. Надежные опоры

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

В ЛКШ нужно провести электричество до хутора. Представим дорогу от ГК до хутора как прямую. Тогда в некоторых позициях будет стоять столбы через которые идут линии электропередачи. Пусть в точке x_i находится столб высоты h_i . Чтобы столб не упал, ему нужна опора под углом 45 градусов. Для удобства установки ее сделали в сторону хутора. Тогда опора будет соединять точки с координатам (x_i, h_i) и $(x_i + h_i, 0)$.

Одних проводов не хватает для надежного соединения. Где-то на этой дороге будет расположена специальная будка ширины w и высоты h для контроля электричества и оповещения админов. Чтобы было возможно подключить будку к общей сети, она должна находиться между столбом и соответствующей ему опорой. При этом будка может касаться опор и столбцов, но не может их пересекать.

Пока ничего не построили, нужно проверить может ли быть осуществлен такой план или нет. Вас как самых старательных учеников ЛКШ попросили это проверить.

Формат входных данных

В первой строке дано целое число n — количество столбов ($1 \leq n \leq 2 \cdot 10^5$).

В следующих n строках вводятся пары чисел x_i, h_i — место, где находится столб на прямой, и его высота ($1 \leq x_i, h_i \leq 10^9$).

В последней строке вводятся два целых числа w, h — размеры будки ($1 \leq w, h \leq 10^9$).

Формат выходных данных

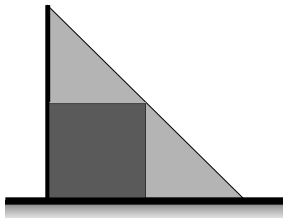
Выведите «YES», если такой план осуществим, и «NO» в ином случае.

Примеры

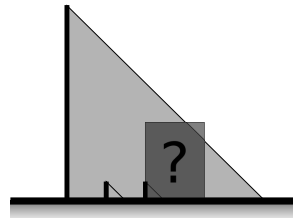
stdin	stdout
1 1 4 2 2	YES
3 1 10 3 1 5 1 3 4	NO

Замечание

Картинка к первому примеру:



Картинка ко второму примеру:



Задача L. Шифры прошлого

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	512 мегабайт

Срочная новость! Британские учёные обнаружили сохранившуюся табличку из n иероглифов $a_1, \dots, a_n$, написанную на языке древней цивилизации, письменность которой ещё не расшифрована. Известно, что она состоит из k символов, которые для упрощения записи были обозначены числами от 1 до k .

Также британские учёные обнаружили, что каждую из букв древние люди произносили с разным уровнем громкости. Эти уровни они также обозначили числами от 1 до k . Теперь перед ними стоит амбициозная задача — определить, какую букву с какой громкостью необходимо произносить. Учёные предположили, что при правильном распределении громкостей достигается минимально возможная *выразительность* известного им текста. *Выразительностью* текста называется сумма модулей разности громкостей соседних букв в этом тексте.

Теперь учёные обратились к вам с просьбой найти оптимальное распределение громкостей для найденной таблички. Помогите им, и наука вас не забудет!

Формат входных данных

В первой строке даны числа n, k ($1 \leq n \leq 10^5, 1 \leq k \leq 20$) — число иероглифов в табличке и число символов в письменности.

Во второй строке находятся числа $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq k$) — содержимое таблички.

Формат выходных данных

Выведите k , чисел i -е из которых обозначает громкость, с которой надо произносить иероглиф номер i .

Примеры

стандартный ввод	стандартный вывод
4 2 1 2 1 2	1 2
6 4 3 1 2 3 3 1	2 1 3 4

Задача М. Червячьи пути

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 7.5 секунд
Ограничение по памяти: 256 мегабайт

Червяк Аркадий поселился на двоичном поле n на m . Он уже изучил много свойств своего дома и придумал еще одно. Он зафиксировал для себя какое-то целое число k . Затем он встает в какую-то клетку таблицы и ходит по всем возможным путям (необязательно простым) длины k из своей клетки. При этом он на каждом шаге обязательно переходит в соседнюю клетку. Пока он идет по пути он запоминает символ который видит на клетке (0 или 1). Ему стало интересно, а сколько различных строк он может увидеть, если пройдет по всем путям. Для одной клетки он смог быстро посчитать, а вот для большего числа ему уже тяжело, помогите ему удовлетворить свое любопытство.

Формат входных данных

В первой строке даны четыре натуральных числа n, m, q, k — размер таблицы, количество запросов и длина пути ($1 \leq n, m \leq 1000$, $1 \leq q \leq 10^5$, $1 \leq k \leq 13$). В следующих n строках вводится двоичное поле размера $n \times m$. В следующих q строках вводятся пары x_i, y_i — новое местоположение Аркадия ($1 \leq x_i \leq n$, $1 \leq y_i \leq m$).

Формат выходных данных

Выведите q чисел — ответы на запросы.

Пример

стандартный ввод	стандартный вывод
5 7 10 3 1110101 0010011 1111000 0011010 1100000 4 7 5 4 5 3 3 3 5 7 1 5 4 1 4 6 2 1 1 4	3 4 4 2 2 2 4 2 4 4

Задача N. Перетасуй пароль

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Мечта любого школьника в ЛКШ — получить пароль от преподаского Wi-Fi. И вот однажды ночью Федя, Саша и Таня, школьники третьей параллели, узнали, что пароль состоит из трех символов «S», «i» и «s», но они не уверены, в каком порядке расположены эти буквы.

Чтобы не гадать, ученики придумали следующее: сначала каждый из них записал себе на бумажке один символ — так у них получилась какая-то тройка T_1, T_2, T_3 (Федя держит T_1 , Саша — T_2 и Таня — T_3). Они считают, что это текущая версия пароля.

Затем ребята хотят N раз сделать так, чтобы двое из них обменялись своими бумажками.

Школьники знают, что N — их счастливое число, а поэтому уверены, что итоговый пароль будет верным. Помогите понять ребятам, действительно ли N является счастливым числом, а именно проверьте, что после N обменов возможно получить правильный пароль P_1, P_2, P_3 .

Формат входных данных

В первой строке даны три символа, которые написаны на бумажках Феи, Саши и Тани — S_1, S_2, S_3 , которые образуют перестановку из символов «S», «i», «s». Во второй строке расположены символы пароля P_1, P_2, P_3 , которые образуют перестановку из символов «S», «i», «s». В последней строке расположено число N ($1 \leq N \leq 2 \cdot 10^{18}$).

Формат выходных данных

Выведите «Yes», если с помощью описанной операции школьники параллели 3 имеют возможность угадать пароль от преподаской сети, и «No» иначе.

Примеры

стандартный ввод	стандартный вывод
S i s S s i 5	Yes
S i s s i S 7	Yes