



А. Первые пары

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

Входной / выходной файл: стандартный поток ввода / стандартный поток вывода

В прошлом учебном году Вася Пупкин начал обучение в X-OSPU (eXtreme Open Source Programming University), а в этом году он уже второкурсник. И теперь он, как и многие другие студенты, почувствовавшие себя старожилками, ленится ходить на первые пары. Но есть две проблемы. Во-первых, в X-OSPU очень строго относятся к срокам сдачи лабораторных и снижают баллы, если не сдать их вовремя. Поэтому всё-таки придётся ходить на все лабораторные. А во-вторых, Васе очень нравится преподаватель по имени Шера, поэтому на все его занятия он тоже хочет ходить.

В прошлом году Вася научился программировать под свой смартфон на RobotOS, поэтому теперь он хочет написать программу, которая будет автоматически устанавливать будильники на его смартфоне в зависимости от расписания (чтобы автоматизировать прогуливание, так сказать). А для этого необходимо сначала определить, в какие дни нужно приходить на первую пару, а в какие — нет.

Помогите Васе написать этот модуль для его программы. На вход модуля поступает расписание занятий на неделю, а на выход он должен выдать список дней недели, в которые таки нужно идти на первую пару.

Входные данные

На вход программы поступает расписание занятий. Оно состоит из пяти блоков — по одному для каждого рабочего дня недели. Блоки разделяются пустой строкой, но после последнего блока пустой строки нет.

Блок начинается с отдельной строки-заголовка, в которой указан день недели («Monday», «Tuesday», «Wednesday», «Thursday» или «Friday»). Блоки всегда упорядочены от понедельника к пятнице. За строкой-заголовком следует одна или более строк, описывающих занятия в этот день.

Каждое занятие описывается одной строкой, которая содержит через пробел: номер пары (целое число от 1 до 9), тип занятия («lec» — лекция, «prc» — практика или «lab» — лабораторная) и имя преподавателя (от 1 до 20 маленьких и больших английских букв). Занятия упорядочены по номерам пар, но некоторые пары могут быть пропущены («окна»).

Выходные данные

Выведите список дней недели, в которые нужно идти на первые пары, упорядоченный от понедельника к пятнице. Названия дней недели должны быть такими же, как во входных данных, и их следует выводить по одному в строке.

Если ни в один день не нужно идти на первую пару, выведите единственную строку «Truant!» (что значит «Прогульщик!»).

Пример

Входные данные	Результат
Monday 1 prc Lyubchenko 2 prc Chera 3 lec Ruvinskaya	Wednesday Thursday Friday



Входные данные	Результат
<p>Tuesday 2 lab Ruvinskaya 3 lec Krisilov</p> <p>Wednesday 1 lab Kavickaya 2 lec Zinovatnaya 3 lec Lyubchenko</p> <p>Thursday 1 lec Chepa 2 lab Chepa 3 lab Zinovatnaya</p> <p>Friday 1 lab Lyubchenko 2 prc Ruvinskaya 3 prc Zinovatnaya</p>	
<p>Monday 1 lec Onishenko 2 prc Komlevaya 3 lab Kungurtsev</p> <p>Tuesday 1 lec Kungurtsev 2 lab Onishenko 3 lec Chepa</p> <p>Wednesday 2 lab Chepa 3 prc Kungurtsev</p> <p>Thursday 1 prc Onishenko 2 prc Chepa 3 lec Komlevaya</p> <p>Friday 1 lec Sinegub 2 lab Komlevaya 3 prc Sinegub</p>	<p>Truant!</p>

Не забывайте, что для удобства вы можете скачать электронную версию условия с примерами.



В. Правильное выражение

Ограничение по времени: для C/C++ — 1 секунда, для Java — 2 секунды

Ограничение по памяти: 256МБ

Входной / выходной файл: стандартный поток ввода / стандартный поток вывода

Однажды Федя гулял со своей девушкой по улицам Молводанки и наткнулся на вопиющее безобразие: кто-то исписал стену школы баллончиком краски. На удивление парня, на стене не было привычных ему бессмысленных фраз или ругательных слов. Лишь математическое выражение вида равенства двух чисел. Чувство прекрасного и математический склад ума не давали Феде покоя, ведь он понимал, что выражение требует доработки. Вспомнив о том, что он студент X-OSPU, лучшего университета города, он решил восстановить математическую справедливость! А именно: добавить в левой части выражения несколько (возможно ноль) знаков «+», чтобы выражение имело смысл.

Все попытки девушки уговорить Федю бросить это занятие и пойти кататься на качелях и есть мороженое закончились фиаско. Федя всё внимание уделял лишь стене. В отчаянии девушка Феде обращается к вам, как к программисту, за помощью. Напишите программу, которая по заданному выражению восстановит математическую справедливость. Это очень важно для девушки, ведь качели и мороженое ждать не будут!

Входные данные

В первой строке входного файла записано выражение вида «a=b», где $0 \leq a \leq 10^{5000}$, $0 \leq b \leq 5 \cdot 10^3$. В левой части выражения возможны лидирующие нули. Гарантируется, что количество цифр в левой части не превышает значения $5 \cdot 10^3$.

Выходные данные

Если решение существует, то выведите строку «Possible», затем с новой строки правильное выражение. В противном случае выведите «Impossible».

Пример

Входные данные	Результат
312=6	Possible 3+1+2=6
1216479=192	Possible 12+164+7+9=192
125=10	Impossible

Примечание

Если в исправленном выражении подпортить изначальные цифры или их порядок — чувство прекрасного Феде будет нарушено, и вы незамедлительно получите «Presentation Error».



С. Упертость Марксинуса

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

Входной / выходной файл: стандартный поток ввода / стандартный поток вывода

Марксинус днями и ночами не спит, только и делает, что думает о ней... О единице в числе. Да, 1 — наилучшая цифра, по мнению Марксинуса, и он уверен, что такая прекрасная цифра должна встречаться как можно больше раз в числах.

Марксинус довольно упертый и доверяет только вычислительной технике, с людьми дело иметь он не особо хочет, особенно если тема спора затрагивает его любимую... Да-да, конечно же единицу. А упертость Марксинуса заключается в том, что он уверен, что чисел от 1 до 10^n , в записи которых встречается единица, больше, чем тех, в которых она не встречается.

Докажите или опровергните его мысль. В случае, если Марксинус прав, выведите «YES», иначе — «NO».

Входные данные

Одно целое число n ($1 \leq n \leq 10^6$).

Выходные данные

В случае, если Марксинус прав, выведите «YES», иначе — «NO».

Пример

Входные данные	Результат
1	NO



D. Дерево Фенвика

Ограничение по времени: 5 секунд

Ограничение по памяти: 256МБ

Входной / выходной файл: стандартный поток ввода / стандартный поток вывода

Петя Массивчиков, друг Васи Пупкина, в этом году тоже поступил в X-OSPU на ту же кафедру что и Вася — кафедру СПО (Самого Программного Обеспечения). И вот, Петя просит своего друга-второкурсника помочь ему с лабораторной на тему дерева Фенвика. «Мы не учили дерево Фенвика, — сказал Вася. — А конспект есть?». К несчастью, конспект Пети трудно разобрать, но самое главное ребята всё же смогли прочитать:

Дерево Фенвика ... очень быстро ... сумму элементов на отрезке ... Нужен массив той же длинны n ... сначала заполнить нулями ... Изменение элемента idx с old на new ... $for (i=idx; i < n; i = i || i+1) fenwick[i] += old - new$; ... Сумма от 0 до r ... $sum = 0$; $for (i=r; i >= 0; i = (i \&\& i+1) - 1) sum += fenwick[i]$; ... Сумма от l до r ... $sum(r) - sum(l-1)$...

А вот и само задание:

Дан массив с индексацией от L до R включительно, который изначально заполнен нулями. Программа должна обрабатывать запросы двух типов:

1. *set idx new* — нужно поменять значение элемента с индексом idx на new .
2. *sum l r* — нужно вывести сумму всех чисел в массиве, от L до R .

— Точно? Ты уверен, что Николай Алгоритмович так и сказал? — спросил Вася.

— Стопроцентно! Я с доски списывал, а Николай Алгоритмович на доску со своего листика переписал. — ответил Петя.

— Ну тогда это совсем просто! — обрадовался Вася, отложил конспект в сторону и начал кодить.

Попробуйте и вы закодировать эту лабораторную.

Входные данные

В первой строке находятся три целых числа: L, R ($0 \leq L, R \leq 10^5$) — минимальный и максимальный возможный индекс массиве и Q ($1 \leq Q \leq 10^5$) — количество запросов. Далее следует Q строк, описывающих запросы.

Каждый запрос описывается типом (строка «set» или «sum») и двумя аргументами; все три элемента разделены пробелами. Для запросов типа «set» первый аргумент — целое число idx ($L \leq idx \leq R$), а второй — целое число new ($-1000 \leq new \leq 1000$). Для запросов типа «sum» первый и второй аргументы — это целые числа l и r соответственно ($L \leq l, r \leq R$).

Выходные данные

Для каждого запроса типа «sum» выведите ответ в отдельной строке.

Пример

Входные данные	Результат
1 7 9	41
set 1 17	42
set 2 3	



Входные данные	Результат
set 3 54 set 4 -16 set 6 -21 set 7 4 sum 2 4 set 5 1 sum 2 5	
42 42 2 set 42 42 sum 42 42	42

Е. День рождения

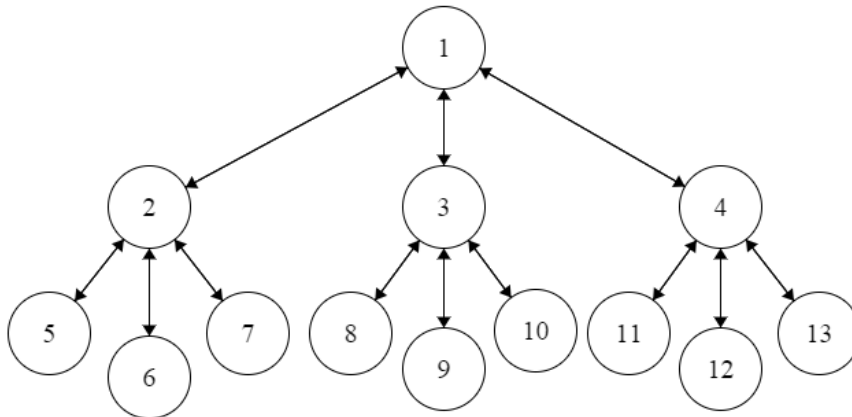
Ограничение по времени: для C/C++ — 1 секунда, для Java — 3 секунды

Ограничение по памяти: 256МБ

Входной / выходной файл: стандартный поток ввода / стандартный поток вывода

У Андрея сегодня день рождения! В этот светлый день он решил пригласить к себе n друзей, чтобы разделить с ними свою радость, выпить компоту и покушать торт. Чтобы никто не отказался от приглашения, Андрей решил оплатить проезд туда и обратно для каждого из друзей. Осталось лишь рассчитать, сколько для этого надо отложить от стипендии гвирен.

Город, в котором живут ребята, представляет собой бесконечное дерево с корнем в вершине 1, где у каждой вершины есть ровно k детей. Вершины в дереве нумеруются послойно: сначала корень получает номер 1. Затем выбирается вершина с минимальным номером, которая еще не имеет детей и получает детей слева направо заполненными минимальными доступными номерами. Например, первые 3 слоя троичного дерева выглядят так:



Все люди в городе живут в одной из вершин дерева. Каждая вершина является отдельным районом. Между двумя районами A и B есть прямая дорога, если A — прямой родитель B , либо B — прямой родитель A . По городу можно проехать только с помощью одноместного такси, которое стоит 1 гвирну за одно ребро.

Поскольку в ВУЗе X-OSPU с недавнего времени отменили индексацию на стипендию, каждая гвирна стала на вес золота. Поэтому Андрей хочет безошибочно спрогнозировать свои затраты и настоятельно просит вас помочь ему в этом деле.

Входные данные

В первой строке входного файла записаны три целых числа: n , k и s ($2 \leq n \leq 10^5$, $2 \leq k \leq 10$, $1 \leq s \leq 10^{18}$) — количество друзей Андрея, количество детей каждой вершины дерева и номер вершины, в которой проживает Андрей. Далее в n строках записаны целые числа p_i ($1 \leq p_i \leq 10^{18}$) — номер вершины, в которой проживает i -ый друг Андрея.

Выходные данные

Одно число — количество необходимых гвирен на транспортировку всех друзей.



Пример

Входные данные	Результат
3 2 1 2 6 7	10
5 3 6 47 61 14 6 11	30



Ф. Зеркальная точка

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

Входной / выходной файл: стандартный поток ввода / стандартный поток вывода

Заданы две точки на плоскости. Найти третью точку, симметричную первой относительно второй.

Входные данные

В первой строке заданы через пробел целочисленные координаты x и y первой точки, а во второй строке — координаты второй точки ($-10^9 \leq x_1, y_1, x_2, y_2 \leq 10^9$).

Выходные данные

Выведите через пробел x и y координаты третьей точки.

Пример

Входные данные	Результат
1 1 3 2	5 3



Г. В поисках сокровищ

Ограничение по времени: для C/C++ — 1 секунда, для Java — 4 секунды

Ограничение по памяти: 256МБ

Входной / выходной файл: стандартный поток ввода / стандартный поток вывода

Очень сложно заставить себя встать рано в понедельник утром и прийти на первую пару. Но что поделывать, если ты студент X-OSPU и первой парой злостный деканат поставил «Системные операции» с Валентином Дмитриевичем. «Надо быть однозначно», — думает почти каждый студент. Такого же мнения и студент Денис Максимюк, для которого, в отличие от остальных, пара — это лишний повод поиграть в любимые игры на своем ноутбуке. И вот на выходных он скачал новую игру, в которую собирается играть всю ближайшую учебную неделю.

В игре главной целью является поиск и сбор сокровищ по туннелям подземелья. В подземелье есть n комнат, которые соединены $n-1$ туннелем. В начале игры главный персонаж всегда стоит в комнате под номером 1. Гарантируется, что из начальной комнаты есть пути во все остальные, причем каждый такой путь однозначен. В комнатах, откуда нет дальнейшего пути, находятся сундуки с золотыми монетами. Таким образом, подземелье представляет собой ориентированное дерево с корнем в вершине 1. Комнаты с сундуками в таком дереве являются листьями.

Игра происходит следующим образом. На каждом шагу игрок стоит в одной из комнат и выбирает туннель, по которому пойдет дальше. Поскольку в подземелье темно, то игрок выбирает один из них равновероятно. Туннели очень ненадежны, поэтому каждый туннель имеет свою вероятность обвалиться во время прохождения игрока по нему. После успешного прохождения туннеля назад вернуться уже нельзя — риск оказаться под завалами очень велик. Если игрок попал в комнату, в которой находится сундук, он немедленно забирает все монеты себе и игра заканчивается.

Всё было бы хорошо, но в некоторых комнатах находятся монстры. Сила монстра в комнате под номером i равна $monster_i$. К счастью, игрок обладает силой, которая в начале игры равна $power$. Если на текущий момент сила игрока превышает силу монстра, то вероятность победы над ним равна единице. Иначе вероятность пройти этого монстра равна отношению силы игрока к силе монстра. Если игрок успешно одолевает монстра, то его сила увеличивается на $lcpd(v, u)$, где v — номер комнаты, где находится игрок, u — номер комнаты, где находится монстр, а $lcpd$ (lowest common prime divisor) — наименьший общий простой делитель двух чисел (если такого числа не существует, то полагаем, что оно равно нулю). Так же за успешное прохождение монстра игрок получает $gcpd(v, u)$ золотых монет, где v и u — номера комнат, как и в случае с $lcpd$, а $gcpd$ (greatest common prime divisor) — наибольший общий простой делитель (если такого числа не существует, то полагаем, что он равен единице). Если же игрок не смог одолеть монстра, то он теряет $gcpd(v, u)$ монет. После поражения от монстра игра заканчивается.

Следует обратить внимание на то, игрок вступает в схватку с монстром только в том случае, когда хочет пройти в комнату, где монстр находится. Из этого следует, что монстр охраняет вход в комнату. Если игрок уже находится в комнате, то, в силу своей доброй души, не станет трогать монстра. Монстр так же не станет трогать игрока.

Результатом игры является количество монет, которые успел собрать игрок. Изначально у игрока нет ни одной монеты при себе. Окончанием игры считается момент, когда игрок достиг комнаты с сундуком, либо погиб в результате схватки с монстром, либо остался под завалами туннеля. Победой в игре считается найденный сундук с монетами, а остальные исходы — поражением. Следует отметить, что при поражении игрок не теряет все монеты, а остается с тем количеством, что было на момент события, которое привело к окончанию игры. Это, в частности, означает, что, оказавшись под завалами туннеля, игрок не теряет монет, которые он собрал



раньше. Важно обратить внимание, что возможна ситуация, где количество монет к концу игры будет отрицательным.

«Ух, как всё запутанно», — подумал Денис. Пройдя эту игру очень много раз, он подсчитал среднее значение монет, которые он собрал. Теперь ему надоело играть, и он решил проверить, насколько подсчитанное среднее значение монет отличается от математического ожидания количества монет за игру. Денис не очень силен в теории вероятности, поэтому просит вас помочь подсчитать ему значение математического ожидания количества монет после прохождения игры.

Входные данные

В первой строке входного файла записано два целых числа n и $power$, где n — количество комнат в подземелье ($1 \leq n \leq 10^5$), а $power$ — начальная сила игрока ($0 \leq power \leq 10^3$). Далее идут n строк — описание комнат.

Описание комнаты начинается с двух целых чисел — $monster_i$ и m_i , где $monster_i$ — сила монстра в i -ой комнате ($1 \leq monster_i \leq 10^3$), а m_i — количество туннелей, которые ведут из данной комнаты. Если $monster_i = 0$, следует считать, что в комнате нет монстра. Далее следуют m_i пар значений $(u, p_{i,u})$, первое из которых целое и обозначает номер комнаты, куда ведет туннель, а второе — значение вида « a/b », где $0 \leq a \leq b \leq 100$, a и b — целые, и обозначает дробное число вероятности успешного прохождения туннеля. Если же $m_i = 0$, то затем следует одно целое число g_i ($0 \leq g_i \leq 10^5$) — количество золотых монет, которые находятся в сундуке комнаты i .

Выходные данные

Одно вещественное число — значение математического ожидания количества золотых монет, которые сможет собрать игрок за время прохождения игры. Ответ следует выводить с точность не менее 10^{-6} .

Пример

Входные данные	Результат
3 100 0 2 2 1/2 3 1/3 0 0 200 0 0 100	66.6666666666667
3 10 0 2 2 1/2 3 1/2 20 0 50 0 0 10	8.7500000000000



Н. Многоугольник по серединам сторон 2

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

Входной / выходной файл: стандартный поток ввода / стандартный поток вывода

Задано нечетное количество (N) точек на плоскости: A_1, A_2, \dots, A_N . Построить многоугольник размерности N , у которого N подряд идущих сторон имеют точки A_1, A_2, \dots, A_N в качестве середин. Вывести координаты вершин многоугольника в порядке обхода, начиная со стороны, у которой середина A_1 , затем — A_2 и т.д. Гарантируется, что ответ существует.

Входные данные

В первой строке находится число N ($3 \leq N \leq 15$) — количество исходных точек. В следующих N строках заданы через пробел целочисленные x и y координаты точек A_1, A_2, \dots, A_N ($-2^{31} \leq x_i, y_i \leq 2^{31} - 1$).

Выходные данные

Выведите целочисленные координаты вершин многоугольника, каждую точку в отдельной строке.

Пример

Входные данные	Результат
3 1 1 3 2 2 1	0 0 2 2 4 2



I. Главный корпус

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

Входной / выходной файл: стандартный поток ввода / стандартный поток вывода

В честь празднования столетия основания X-OSPU руководство ВУЗа решило сделать косметический ремонт фасада главного корпуса. И вот уже все работы почти закончены. Осталось лишь заменить старую буквенную вывеску над главным входом на название ВУЗа — OSPU. Решение было единогласным — снять старую вывеску и заказать новую. «Стойте! — возразил декан Института системных компьютеров, — По контракту компания может нам заменить некоторые буквы, что может нам помочь сэкономить денег!».

По контракту, компания предоставляет возможность замены букв. Поменять даже несколько десятков букв будет стоить дешевле, чем закупка новой вывески. Замена одной буквы означает, что заменяется буква во всех позициях, где она встречается в вывеске. Например, если на вывеске было написано «ПРИМЕР», то после замены буквы Р на букву А получим «ПАИМЕА». Замены касаются лишь старых букв. Это означает, что при попытке замены букв А в полученном «ПАИМЕА» ничего не произойдет.

Руководство ВУЗа задумалось о том, можно ли применить такие замены, чтобы получить в строке нынешней вывески вхождение строки «ospu». Если такая возможность будет найдена, то далее достаточно лишь убрать все литеры левее и правее данной надписи. Вот только надпись слишком длинная, а до конца совещания остались считанные минуты... К счастью, у них в ВУЗе учится смыслённый программист Интегралус, который вызвался помочь руководству и написать программу, которая в миг определит возможность спасения бюджета X-OSPU.

У Интегралуса осталось 15 минут до конца совещания, а программа еще не готова. Помогите Интегралусу написать программу и спасти бюджет X-OSPU. Иначе студенты из-за недостатка средств на отопление вынуждены будут зимой сидеть дома, а летом учиться, что совершенно недопустимо!

Входные данные

В первой строке входного файла записана строка s ($1 \leq \text{length}(s) \leq 10^5$) — надпись на вывеске, что сейчас украшает главный корпус X-OSPU. Надпись состоит только из маленьких латинских литер.

Выходные данные

Если возможна замена, то выведите минимальное количество букв, которые надо заменить, чтобы получить строку «ospu» в качестве подстроки исходной строки. Иначе выведите «-1» (без кавычек).

Пример

Входные данные	Результат
onpu	1
sometext	3
anas	-1



Ж. Марксинус. Начало.

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

Входной / выходной файл: стандартный поток ввода / стандартный поток вывода

Давным-давно, когда Марксинус еще учился в школе, произошло событие, которое изменило его судьбу. В тот день ему ещё дорогу перебежал черный кот Ангенс. На уроке Совершенной Арифметики раздавали результаты самостоятельных работ, и, к сожалению, Марксинус получил «1». Будучи безумно разочарованным и поникшим, он ударился лбом об стол, на котором лежала та зловещая работа. Удар был настолько сильным, что частицы чернил, которыми была написана «1», попали в нейроны мозга Марксинуса. Очнувшись, он обнаружил, что в классе никого нет и бегом двинулся домой.

Проходя мимо магазинов и ларьков, наш герой начал чувствовать себя как-то странно, когда видел ценники, в числе которых содержалась единица (в то время как на числа без единиц он никак не реагировал). Не придав этому странному ощущению особого внимания, Марксинус пришел домой и лег спать, чтобы забыть об этом дне, будто его и не было вовсе.

На следующее утро Марксинус был полон сил и энтузиазма. Отправившись в школу, он заметил, что всё так же странно реагирует на числа. По дороге он решил заскочить в булочную, чтобы прикупить себе еды в школу. Продавщица всё никак не могла разглядеть ценник издали. Марксинус тоже не видел, но был уверен, что там написано «11 гвирен». Сказав это продавщице, Марксинус подошел к своему выбранному товару (тут читателю предоставляется выбор товара на свой вкус), и был приятно удивлен тому, что он оказался прав.

Слухи о супер-способности Марксинуса начали распространяться молниеносно. «Смотри, это ведь тот самый Единцевидящий!», — кричали мальчишки, дочитывая таблицу умножения на углу школы.

Слухи дошли и до местного злодея Зерро. Он был бездарным, оттого жутко завидовал Марксинусу. Зерро знал, что видя «1» в числе, у Марксинуса повышалось настроение и он был счастлив. Собственно, это и послужило основой для коварного плана Зерро. План был его довольно прост — стереть с лица Земли все «1», чтобы Марксинус никогда не был счастлив. Но будучи бездарным, он действовал в одиночку и довольно глупо: вытирая со всех мест единицы, он больше не возвращался к ним. А внимательные люди (те же продавцы) замечали это варварство и приписывали единицу обратно. Марксинус, проходя мимо испорченных ценников, стал замечать, что ему уже не так радостно, как когда-то.

Друг Марксинуса Андрей просит Вас помочь его другу, собрав такие числа, после которых Марксинус будет снова счастлив. Выясните, будет ли Марксинус счастлив, увидев данное число.

Входные данные

Одно число n , $0 \leq n \leq 10^{100000}$. Число может содержать ведущие нули.

Выходные данные

В случае, если Марксинус счастлив, выведите «Yes», иначе — «No».

Пример

Входные данные	Результат
54190661	Yes
008	No



К. Казалось бы, тернарный поиск

Ограничение по времени: для C/C++ — 0.1 секунды, для Java — 0.5 секунды

Ограничение по памяти: 256МБ

Входной / выходной файл: стандартный поток ввода / стандартный поток вывода

Вахтангенс был шокирован, когда узнал, что в X-OSPU кафедру прикладной математики переводят в Научный институт чисел и относительных статистических игр (сокращенно — НИЧОСИ). Но больше всего он был удивлен тому, что в новом институте кроме математики ребят будут обучать еще и программированию. Преподаватель Алексей Петрович знает, как Вахтангенс обожает математику, поэтому на первую лабораторную работу дал ему такое задание:

$$\left(\sqrt{-c\left(2p - \frac{(p+2c)p}{c}\right)}\right)^3 - (q-p)\left(p \cdot |p+q| + \sum_{k=1}^q q\right) - p^2 - 2p \sum_{k=1}^{p-1} k = r$$

«Вах, какой хороший уравнение!», — подумал Вахтангенс. Он быстро смог решить его на бумаге, но, к сожалению, у него не было достаточных навыков перенести все его старания в программный код. Помогите Вахтангенсу приспособиться к новому институту и перенести решение в код.

Входные данные

В единственной строке через пробел указаны числа c и r ($10^{-9} \leq c, r \leq 10^9$).

Выходные данные

Два целых числа p и q — решение уравнения. Выведите любое решение уравнения в виде двух чисел p и q , где значения $p, q \in \mathbb{N}$ и $p, q \leq 10^6$. Если решения не существует, примите p и q равными 0.

Пример

Входные данные	Результат
10 7	2 1
1011 2014	0 0



L. Ровесники

Ограничение по времени: для C/C++ — 1 секунда, для Java — 2 секунды

Ограничение по памяти: 256МБ

Входной / выходной файл: стандартный поток ввода / стандартный поток вывода

Для участия в некотором событии приглашены 2 группы «W» и «M» с одинаковым числом участников N . Так как ранее никто из группы «W» не был знаком ни с кем из группы «M», организаторы события предложили создать пары для знакомства с участниками из разных групп, а для более свободного общения поставили условие, чтобы разница в возрасте для участников каждой пары была не более d лет. Какое наибольшее число пар можно создать?

Входные данные

В первой строке через пробел указаны два целых числа: d ($0 \leq d \leq 3$) и N ($1 \leq N \leq 10^5$). Во второй строке через пробел указаны целые числа W_1, W_2, \dots, W_N ($1 \leq W_i \leq 100$). В третьей строке через пробел указаны целые числа M_1, M_2, \dots, M_N ($1 \leq M_i \leq 100$).

Выходные данные

Одно целое число — максимальное возможное количество пар.

Пример

Входные данные	Результат
2 6 5 6 13 4 4 4 14 9 8 2 3 7	5

