

Задача А. Два охранника

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Бесконечную клетчатую плоскость охраняют два робота: Леголас и Гимли. Леголас охраняет клетки (x, y) , для которых $1 \leq x \leq a$ и $1 \leq y \leq b$. Гимли охраняет клетки (x, y) , для которых $1 \leq x \leq c$ и $1 \leq y \leq d$. Сколько клеток находится под охраной хотя бы одного робота?

Формат входных данных

В первой строке заданы четыре целых числа: a , b , c и d ($1 \leq a, b, c, d \leq 10^4$).

Формат выходных данных

Выведите число клеток, находящихся под охраной хотя бы одного робота.

Пример

стандартный ввод	стандартный вывод
2 6 3 4	16

Задача В. Расписание конференции

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Мария — очень занятой профессор. Она участвует во многих университетских мероприятиях, поэтому найти время для посещения ещё одной конференции ей непросто. Скоро состоится новая конференция, но Мария пока не знает, в какие дни сможет присутствовать лично.

Конференция длится n дней. Некоторые дни отведены под стендовые доклады, а остальные — под лекции приглашённых докладчиков. Мария планирует посетить несколько дней конференции подряд: с ℓ -го по r -й день включительно для некоторых $1 \leq \ell \leq r \leq n$. Пока она не знает, какой именно отрезок дней сможет выбрать. Однако она подметила любопытное совпадение: для всех возможных пар (ℓ, r) число отрезков, начинающихся с дня стендовых докладов, равно числу отрезков, начинающихся с дня лекций.

Мария рассказала об этом другу, но тот ответил, что это невозможно: не существует расписания конференции длительностью n дней, обладающего таким свойством.

Определите, существует ли такое расписание, и, если существует, постройте его.

Формат входных данных

Единственная строка входных данных содержит целое число n — число дней конференции ($2 \leq n \leq 365$).

Формат выходных данных

Выведите «Yes», если существует расписание конференции, в котором число отрезков, начинающихся с дня стендовых докладов, равно числу отрезков, начинающихся с дня лекций. В противном случае выведите «No».

Если ответ — «Yes», во второй строке выведите строку длины n , описывающую подходящее расписание. Её i -й символ должен быть «L», если в i -й день проводятся лекции, или «P», если в этот день проводятся стендовые доклады. Если подходящих расписаний несколько, выведите любое из них.

Все буквы в ответе можно выводить в любом регистре.

Примеры

стандартный ввод	стандартный вывод
2	nO
4	Yes PLLP
7	YES LlPppPL

Задача С. Коневая вилка

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Николай увлёкся быстрыми шахматами, где в конце партии на ход остаются считанные секунды. Каждый раз, делая ход, нужно проверить, не поставит ли вражеский конь вилку, одновременно атакуя две фигуры Николая.

По заданным позициям двух фигур определите число клеток, с которых вражеский конь сможет атаковать обе эти фигуры одновременно.

Напомним, что шахматный конь атакует на две клетки по вертикали и одну по горизонтали либо на две клетки по горизонтали и одну по вертикали.

Формат входных данных

В первой строке задано целое число t ($1 \leq t \leq 1000$) — число наборов входных данных. Далее следуют t наборов.

Каждый набор входных данных задаёт на отдельной строке координаты двух различных клеток, где стоят фигуры Николая. Описания двух клеток разделены пробелом. Каждая клетка описывается буквой (от «a» до «h») и цифрой (от 1 до 8) без пробела между ними.

Формат выходных данных

Для каждого набора входных данных выведите на отдельной строке число позиций, из которых вражеский конь может сделать вилку.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 a1 a8 a1 a3	0 1

Задача D. Рейтинг товара

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Прогуливаясь по просторам маркетплейса в поисках игрушек для своего сына Павлика, Анна обратила внимание на рейтинги товаров, которые представляли собой вещественные числа от 1 до 5 с двумя знаками после запятой.

Анна задумалась, а какое минимальное число человек должно было проголосовать и с какими оценками, если каждый может поставить в качестве оценки целое число от 1 до 5, и среднее арифметическое оценок получилось **в точности** равно заданному числу (без округлений).

Формат входных данных

В первой строке записано вещественное число r — рейтинг товара ($1 \leq r \leq 5$). Число содержит до двух знаков после десятичной точки. Если дробной части нет, точка не ставится.

Формат выходных данных

В первой строке выведите минимальное число голосов k , которыми достигается этот рейтинг.

Во второй строке выведите пять целых чисел через пробел: число голосов с оценкой 1, с оценкой 2 и так далее. Суммарное число голосов должно быть равно k . Так как возможных ответов может быть несколько, выведите лексикографически наименьший ответ. Другими словами, минимизируйте число оценок 1, после этого минимизируйте число оценок 2 и так далее.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2.39	100 0 61 39 0 0
1	1 1 0 0 0 0

Задача Е. Вклад и пират

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Старый пират выкопал сундук с n монетами, чтобы купить новый корабль. Для покупки корабля покрупнее он планирует отдать монеты в банк под p процентов годовых и забрать их ровно через год, обязательно дождавшись закрытия последнего вклада.

В банке доступны вклады на любое целое число месяцев от 1 до 12. Проценты начисляются пропорционально сроку вклада: за каждый полный месяц хранения начисляется $p/12$ процентов от первоначальной суммы, размещённой на вкладе.

У банка есть ограничение: он не умеет возвращать дробную часть суммы. Поэтому, когда срок вклада заканчивается, пират получает только целое число монет. Дробный остаток при этом теряется.

В начале года пират кладёт все монеты на вклад от 1 до 12 месяцев. По окончании срока первого и последующих вкладов он может либо забрать всю накопленную сумму, либо полностью реинвестировать её, открыв новый вклад на один из доступных сроков.

Определите, какое максимальное число монет может оказаться у пирата ровно через год.

Формат входных данных

В единственной строке через пробел записаны целые числа n и p — изначальное число монет и годовой процент вклада ($1 \leq n \leq 10^6$; $0 \leq p \leq 100$).

Формат выходных данных

Выведите максимальное число монет у пирата ровно через год.

Примеры

стандартный ввод	стандартный вывод
8 100	18
239 1	241

Замечание

В первом примере у пирата изначально есть 8 монет. Пират может четыре раза подряд вкладывать средства на срок в 3 месяца:

- За первый квартал сумма вклада увеличится ровно на 2 монеты.
По окончании срока пират получит 10 монет.
- За второй квартал сумма вклада увеличится на 2,5 монеты.
По окончании срока пират получит 12 монет (дробная часть пропадает).
- За третий квартал сумма вклада увеличится ровно на 3 монеты.
По окончании срока пират получит 15 монет.
- За четвёртый квартал сумма вклада увеличится на 3,75 монеты.
По окончании срока пират получит 18 монет (дробная часть пропадает).

Задача F. Справедливость для Боба

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Алиса и Боб играют в следующую игру. Изначально есть таблица $n \times n$, в каждой клетке которой стоит либо буква «А», либо буква «В». Игроки ходят по очереди, начиная с Алисы. На своём ходу Алиса выбирает строку и удаляет её из таблицы. Боб, соответственно, на каждом своём ходу выбирает и удаляет столбец таблицы. Игра заканчивается, когда в таблице остаётся ровно одна клетка. Если в этой клетке стоит буква «А», то побеждает Алиса, а если буква «В», то побеждает Боб.

Определите, кто победит при правильной игре.

Формат входных данных

В первой строке дано целое число n ($2 \leq n \leq 5$) — число строк и столбцов в таблице. В следующих n строках задана таблица: по n английских букв в каждой строке, каждая из которых либо «А», либо «В».

Формат выходных данных

Выведите «Alice», если побеждает Алиса, и «Bob», если побеждает Боб.

Примеры

стандартный ввод	стандартный вывод
2 BB AA	Alice
2 AB BA	Bob

Замечание

В первом примере Алиса побеждает, удалив первую строку и получив таблицу из одной строки «AA». Какой бы ход ни сделал Боб, в оставшейся клетке таблицы будет стоять буква «А».

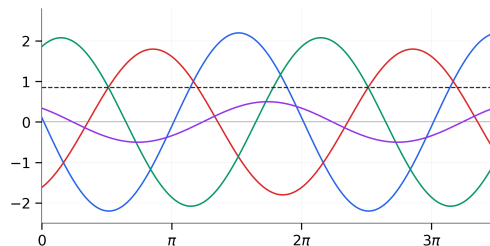
Во втором примере побеждает Боб: если Алиса удалит первую строку, то Боб удалит второй столбец, а если Алиса удалит вторую строку, то Боб удалит первый столбец. Независимо от развития событий, останется клетка с буквой «В».

Задача G. Выпрямление тока

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Арина — директор электростанции. На этой станции есть n электрогенераторов, мощность i -го генератора равна a_i . Каждый из них можно вращать и производить электрический ток. Для каждого генератора Арина может выбрать *фазу* — вещественное число ω_i . При вращении i -го генератора производится напряжение, изменяющееся по формуле $a_i \sin(t - \omega_i)$, где t — параметр, означающий текущее время.

От электростанции к городу тянется n проводов, каждый связан со своим генератором. Однако жителям города переменный ток бесполезен. Для преобразования переменного тока в постоянный все n проводов подводят к специальной будке — *трансформатору*, который выдаёт в город некоторое постоянное напряжение. Трансформатор способен выдавать напряжение U , если в любой момент времени напряжение хотя бы на одном генераторе не меньше U :



Сплошные линии выше показывают напряжения на генераторах, а пунктирная линия — максимально возможное постоянное напряжение на выходе трансформатора при данных фазах.

Подскажите Арине, какое наибольшее постоянное напряжение U можно получить на выходе, выбирая фазы ω_i оптимально.

Формат входных данных

Каждый тест содержит несколько наборов входных данных. В первой строке записано число наборов t ($1 \leq t \leq 10^3$). Далее следует их описание.

Первая строка каждого набора содержит целое число n — число электрогенераторов ($1 \leq n \leq 10^3$).

Вторая строка содержит n целых чисел $a_1, \dots, a_n$ — мощности генераторов ($1 \leq a_i \leq 10^3$). Мощности могут совпадать.

Гарантируется, что сумма n по всем наборам не превосходит 10^3 .

Формат выходных данных

Для каждого набора выведите одно вещественное число — максимальное достижимое напряжение постоянного тока. Ответ считается верным, если его абсолютная или относительная погрешность не превышает 10^{-6} . Формально, если вы вывели ответ p , а правильный ответ равен j , то ваш ответ будет принят, если $|p - j| \leq 10^{-6} \cdot \max\{1, |j|\}$.

Пример

стандартный ввод	стандартный вывод
5	-7
1	0
7	1
2	50
2 9	6.74155
3	
2 2 2	
4	
1 100 100 100	
4	
11 10 8 10	

Задача Н. Деление пополам на городских улицах

Ограничение по времени: 6 секунд
Ограничение по памяти: 1024 мегабайта

Владислав живёт в Петергофе. В Петергофе есть n перекрёстков, соединённых m односторонними дорогами, возможно, разной длины. Другими словами, город Петергоф можно представить как рёберно взвешенный ориентированный граф.

Через месяц Владислав сдаёт экзамен на права. Одна из частей экзамена — езда по городу: нужно успешно добраться от первого перекрёстка до n -го, перемещаясь по дорогам. Владислав не хочет, чтобы посреди экзамена машина заглохла, поэтому ему нужно подготовиться и хотя бы заправиться. Один миллилитр топлива стоит один рубль и позволяет проехать один метр. Таким образом, дело за малым: найти кратчайший путь от первого перекрёстка до n -го, вычислить его длину в метрах, и получится стоимость сдачи экзамена. Верно?

Не всё так просто. В этом месяце в Петергофе работает асфальтоукладочная компания. Она может за один рубль улучшить качество любой дороги, после чего трата топлива на ней сократится вдвое — например, если дорога имеет длину семь метров, то теперь проезд по ней требует лишь 3,5 мл топлива. Эту услугу можно заказать несколько раз как на одну дорогу, так и на несколько разных: например, за шесть рублей можно повысить топливную эффективность одной дороги в 32 раза и другой дороги в два раза.

Изначально в машине Владислава нет топлива, а качество ни одной из дорог города не улучшено. Найдите наименьшее число рублей, за которое Владислав сможет сдать экзамен. Если это число нецелое, округлите его вверх до ближайшего целого.

Формат входных данных

Каждый тест содержит несколько наборов входных данных. В первой строке записано число наборов t ($1 \leq t \leq 10^5$). Далее следует их описание.

Первая строка каждого набора содержит два целых числа n и m , обозначающие число перекрёстков и число дорог ($2 \leq n \leq 10^6$; $1 \leq m \leq 10^6$). Каждая из следующих m строк содержит три целых числа u , v и w , задающих дорогу — номера перекрёстков, на которых дорога начинается и заканчивается, а также длину этой дороги в метрах ($1 \leq u, v \leq n$; $u \neq v$; $1 \leq w \leq 10^6$). Никакая упорядоченная пара (u, v) не встречается более одного раза. Дороги (u, v) и (v, u) могут встречаться одновременно. Другими словами, в соответствующем ориентированном графе нет петель и кратных рёбер, но могут быть встречные рёбра.

Гарантируется, что сумма n по всем наборам не превосходит 10^6 , и сумма m по всем наборам не превосходит 10^6 .

Формат выходных данных

Для каждого набора выведите одно целое число — минимальную целую сумму в рублях, достаточную для сдачи экзамена. Если ни за какую сумму Владислав не сможет проехать от первого перекрёстка до n -го, выведите «-1».

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 2 1 1 2 8 3 2 1 2 3 2 3 3 2 1 2 1 1	4 5 -1
2 4 4 1 4 15 1 2 2 2 3 2 3 4 2 2 1 1 2 1	5 1

Задача I. Shuffler или Amogus

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Это интерактивная задача.

Компания Shuffler Inc. выпускает чёрные ящики. Каждый чёрный ящик принимает строки и выдаёт в ответ числа. Когда чёрный ящик видит строку впервые, он выдаёт случайное целое число, равномерно распределённое на отрезке $[0, 2^{24} - 1]$. Если же чёрный ящик уже получал такую строку раньше, он выдаёт то же число, что выдал на эту строку в предыдущий раз.

Компания Amogus Ltd. также выпускает чёрные ящики. Каждый чёрный ящик также принимает строки и выдаёт в ответ числа. Но не случайные: каждый такой чёрный ящик хранит в себе неприводимый многочлен p степени 24 над полем $\mathbb{F}_2$ (эти и следующие термины объяснены в конце условия). Условимся записывать многочлен $a_dx^d + \dots + a_2x^2 + a_1x^1 + a_0x^0$ над $\mathbb{F}_2$ двоичным числом $a_d \dots a_2a_1a_0$. Когда такой чёрный ящик получает строку s из n символов, он:

- заменяет каждый символ на 8-битную двоичную запись его ASCII-кода,
- записывает эти коды подряд слева направо, получая двоичное число из $8n$ цифр, и дописывает в конец 24 нуля,
- интерпретирует это число из $8n + 24$ цифр как запись многочлена t над полем $\mathbb{F}_2$,
- вычисляет остаток r от деления t на p ,
- записывает его двоичным числом и выдаёт это число в десятичной записи.

Весь этот процесс называется вычислением CRC-24 с неприводимым многочленом p от заданной строки s .

Вы видите перед собой чёрный ящик, который создан либо Shuffler Inc., либо Amogus Ltd. Вы можете подавать ему на вход строки и смотреть, что он выдаёт. Определите, какая из двух компаний создала этот чёрный ящик.

В этой задаче два примера и ещё 50 тестов. В каждом тесте свойства чёрного ящика зафиксированы: два решения, подающие на вход одинаковый список строк, получают одинаковые ответы. В первом примере чёрный ящик создала компания Shuffler Inc. Во втором примере чёрный ящик создала компания Amogus Ltd. с многочленом $p = x^{24} + x^4 + x^3 + x + 1$.

Протокол взаимодействия

Если вы хотите подать на вход строку, выведите её, выведите перевод строки и очистите буфер вывода. Каждая строка должна состоять из маленьких английских букв и иметь длину от 3 до 100 символов.

В ответ вы получите целое число от 0 до $2^{24} - 1$ в десятичной записи, за которым следует перевод строки.

Если вы определили, какая из двух компаний создала чёрный ящик, выведите «Shuffler» или «Amogus» соответственно, после чего завершите работу программы.

Подать на вход строку можно не более 100 000 раз.

Примеры

Пустые строки добавлены в условие, чтобы показать порядок ввода и вывода. При реальном взаимодействии пустых строк нет.

<i>стандартный ввод</i>	<i>стандартный вывод</i>
10356807	abc
11014054	zzzzz
10356807	abc
	Shuffler
12489722	abc
10456448	zzzzz
12489722	abc
12710928	somestring
	Amogus

Замечание

Поле $\mathbb{F}_2$ состоит из двух элементов: 0 и 1. Умножение действует как обычно, а сложение производится по модулю 2. Любой многочлен над $\mathbb{F}_2$ можно записать как $x^{\alpha_1} + x^{\alpha_2} + \dots + x^{\alpha_k}$, потому что коэффициенты могут быть только нулями и единицами. Степенью $\deg(p)$ многочлена p называется максимальный показатель степени x в нём: например, многочлен $x^3 + x^2 + 1$ имеет степень 3. Будем считать, что $\deg(0) = -\infty$.

Неприводимым называется многочлен, который делится нацело ровно на два многочлена: единичный и себя. Например, многочлен $x^3 + x^2 + 1$ делится только на себя и на многочлен $x^0 = 1$, поэтому он неприводимый. С другой стороны, $(x + 1) \cdot (x + 1) = x^2 + 2x + 1 = x^2 + 1$ (так как 2 превращается в 0 в $\mathbb{F}_2$), поэтому многочлен $x^2 + 1$ неприводимым не является.

Частное и остаток от деления многочлена t на многочлен p — это многочлены q и r , для которых выполняется равенство $t = q \cdot p + r$, причём $\deg(r) < \deg(p)$. Например, для $t = x^3 + x^2 + 1$ и $p = x^2 + 1$ получаем $q = x + 1$ и $r = x$, поскольку в $\mathbb{F}_2$ верно равенство $x^3 + x^2 + 1 = (x + 1) \cdot (x^2 + 1) + x$.

ASCII-коды маленьких английских букв: «a» $\rightarrow 97 = 0110\,0001_2$, «b» $\rightarrow 98 = 0110\,0010_2$, «c» $\rightarrow 99 = 0110\,0011_2$, ..., «z» $\rightarrow 122 = 0111\,1010_2$. Например, строка «abc» записывается в двоичном виде как $0110\,0001\,0110\,0010\,0110\,0011_2$, и после дописывания в конец 24 нулей превращается в многочлен $(x^{46} + x^{45} + x^{40}) + (x^{38} + x^{37} + x^{33}) + (x^{30} + x^{29} + x^{25} + x^{24})$, где скобками выделены части, которые получаются из разных букв.

Задача J. Чёрный рынок

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

На протяжении n дней Вольфред приходит на чёрный рынок к Рону, который продаёт m уникальных амулетов.

У Вольфреда есть кошелек, который изначально пуст. Каждый день Рон устанавливает новые цены на оставшиеся амулеты, а Вольфред добавляет в свой кошелек (где могло что-то остаться с предыдущих дней) a монет из банка. После этого Вольфред идёт на рынок и может купить любой (в том числе пустой) набор амулетов, на который у него хватает монет.

Определите минимальное целое a , при котором Вольфред может за эти n дней купить у Рона все m амулетов, и постройте для этого a план покупок.

Формат входных данных

В первой строке заданы два целых числа n и m ($1 \leq n \leq 10\,000$; $1 \leq m \leq 6$) — число дней и амулетов соответственно.

В следующих n строках заданы цены амулетов. В i -й из этих строк содержатся m целых чисел $p_{i,1}, p_{i,2}, \dots, p_{i,m}$ ($1 \leq p_{i,j} \leq 10\,000$), где $p_{i,j}$ — цена j -го амулета в i -й день.

Формат выходных данных

В первой строке выведите минимально возможное целое число a .

Во второй строке выведите план покупок в виде m целых чисел $d_1, d_2, \dots, d_m$ ($1 \leq d_i \leq n$): номера дней, когда следует купить первый, второй, $\dots$, m -й амулеты. Если возможных планов несколько, выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
3 2 3 5 8 13 1 2	1 3 3
2 2 4 100 1 7	4 2 2
2 6 750 4200 1050 2100 1050 900 683 4578 987 2394 914 936	5057 2 1 2 2 2 2

Задача К. Дерево из прямоугольников

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Вам дана прямоугольная таблица, каждая клетка которой либо пуста, либо занята. Гарантируется, что хотя бы одна клетка пуста. Назовём *комнатой* прямоугольник, целиком состоящий из пустых клеток. Можно ли разбить все пустые клетки на комнаты таким образом, чтобы граф из комнат образовывал дерево?

Говоря формальнее, *разбиение* — это такой набор комнат, что каждая пустая клетка таблицы принадлежит *ровно одной* из них. Две различные комнаты разбиения считаются *соседними*, если они касаются по стороне: можно взять по клетке из этих двух комнат так, чтобы эти две клетки были соседними по стороне. Такое отношение естественным образом задаёт неориентированный граф без петель и кратных рёбер, в котором вершинами будут комнаты, а рёбра проводятся между соседними комнатами.

Восстановите искомое разбиение или сообщите, что его не существует.

Формат входных данных

В первой строке даны два целых числа n и m ($1 \leq n, m \leq 1000$) — число строк и столбцов в таблице, соответственно.

В следующих n строках задана таблица: по m символов в каждой строке, каждый из которых равен либо «#», либо «.», что означает пустую и занятую клетку, соответственно.

Формат выходных данных

Если искомое разбиение существует, то выведите в отдельной строке «Yes». В n следующих строках выведите полученное вами разбиение на прямоугольные комнаты в следующем формате: изначально занятые клетки должны всё ещё быть обозначены символом «.», а комнаты разбиения должны быть заполнены символами «А» и «В» таким образом, что в одной комнате все символы одинаковы, а соседним комнатам соответствуют различные символы. Граф соседства данных комнат должен быть деревом. Можно доказать, что если разбиение на комнаты действительно образует дерево, то его можно задать в описанном выше формате. Если возможных ответов несколько, то разрешается вывести любой из них. Для лучшего понимания формата вывода советуем внимательно изучить первый пример.

Если искомого разбиения не существует, то выведите «No».

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 6 # . ### ##### ### . # .### . # .### . .	Yes B . BAA BAABAA BAA . . B .BBB . B .BBB . .
2 2 # . . #	No
1 1 #	Yes A

Замечание

В ответе на первый пример задано следующее разбиение пустых клеток на комнаты (клеткам из одной комнаты сопоставлены одинаковые цифры, а клеткам из разных — разные):

1 . . 455
122455
122 . . 6
.333 . 6
.333 . .

В графе комнат соседними будут комнаты 1 и 2, 2 и 3, 2 и 4, 4 и 5, а также 5 и 6. Легко понять, что такой граф на шести вершинах — дерево.

Во втором примере пустые клетки не образуют связное множество, из-за чего любое разбиение их на комнаты также задаст несвязный граф.

В третьем примере получается граф из одной вершины и нуля рёбер, то есть дерево на одной вершине.

Задача L. Коробки по кругу

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

По кругу стоят n коробок, пронумерованных от 1 до n по часовой стрелке (после коробки с номером n следует коробка с номером 1). Изначально в i -й коробке находится a_i конфет.

За конфетами пришли m человек. Каждый из них должен выбрать ℓ коробок, идущих последовательно по кругу, и взять ровно по одной конфете из каждой выбранной коробки. Разным людям разрешено выбирать одинаковые или пересекающиеся наборы коробок (брать непересекающиеся наборы тоже разрешено). Из коробки нельзя взять больше конфет, чем в ней находится.

Могут ли все m человек выбрать коробки так, чтобы в конце все коробки оказались пустыми?

Формат входных данных

В первой строке заданы три целых числа n , m и ℓ ($1 \leq \ell \leq n \leq 10^5$; $0 \leq m \leq 10^9$) — число коробок, число людей и число конфет, которые должен взять каждый человек.

Во второй строке заданы n целых чисел $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^9$) — число конфет в коробках.

Формат выходных данных

Выведите «YES», если все люди могут забрать конфеты описанным способом так, чтобы коробки опустели. Иначе выведите «NO». Каждую букву можно выводить в любом регистре.

Примеры

стандартный ввод	стандартный вывод
4 3 2 2 3 1 0	YES
4 1 2 1 0 1 0	NO
3 10 2 10 0 10	YES

Замечание

В первом примере люди могут выбрать пары коробок $(1, 2)$, $(1, 2)$ и $(2, 3)$.

Во втором примере выбор из пар $(1, 2)$, $(2, 3)$, $(3, 4)$ и $(4, 1)$. Чтобы забрать конфеты из первой коробки, необходимо выбрать пару $(1, 2)$ или $(4, 1)$, однако во второй и четвертой коробках конфет нет.

Задача М. Героями не рождаются

Ограничение по времени: 4 секунды
Ограничение по памяти: 1024 мегабайта

Настоящий герой должен иметь ровно p единиц силы. Вы начинаете с чистого персонажа с начальной силой 0 и хотите превратить его в настоящего героя. Для этого можно выполнять следующие операции:

- Вы можете натренировать персонажа, увеличивая его силу на единицу. Это стоит одну монету.
- Вы можете растренировать персонажа, уменьшая его силу на единицу. Как ни странно, это тоже стоит одну монету.
- Вы можете применить специальное усиление. Таких усилений n , и s -е из них умножает силу персонажа на s . Применение s -го усиления стоит a_s монет.

Вы можете выполнять эти операции сколько угодно раз и в любом порядке. В частности, для фиксированного s вы можете применить s -е усиление несколько раз, и каждый раз сила будет умножаться на s .

Какое минимальное число монет нужно потратить, чтобы сделать персонажа настоящим героем — то есть добиться того, чтобы персонаж имел ровно p единиц силы?

Формат входных данных

В первой строке содержатся два целых числа n и p ($1 \leq n \leq 10^5$; $1 \leq p \leq 10^8$) — число различных усилений и требуемая сила.

Во второй строке содержатся n целых чисел, описывающих стоимости усилений $a_1, \dots, a_n$ ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите одно целое число: минимальное число монет, которое нужно потратить, чтобы сделать силу персонажа в точности равной p .

Примеры

стандартный ввод	стандартный вывод
3 35 1 2 3	11
5 25 10 10 10 3 10	10
4 17 11 12 13 14	17

Замечание

В первом примере есть несколько оптимальных способов достичь силы 35. Например, можно действовать следующим образом:

$$0 \xrightarrow{+1} 1 \xrightarrow{+1} 2 \xrightarrow{\times 3} 6 \xrightarrow{\times 3} 18 \xrightarrow{\times 2} 36 \xrightarrow{-1} 35.$$

В третьем примере все усиления слишком дорогие, и в оптимальном ответе они не используются.

Задача N. Крестики-ролики

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Пока мы писали правила крестиков-ноликов, по клавиатуре прошёлся кот, и теперь мы играем в крестики-ролики.

Играют двое, X и O, на изначально пустом поле 3×3 . Первым ходит X, дальше игроки ходят по очереди. В свой ход игрок выбирает одну из 9 клеток — пустую или уже занятую (знаки **можно** перезаписывать); пропускать ход нельзя. Затем подбрасывается монетка, и в выбранную клетку с равной вероятностью ставится X или O — независимо от того, кто ходил.

Как только три знака одного игрока оказываются на одной горизонтали, вертикали или диагонали, этот игрок побеждает — даже если это случилось после хода соперника. Если поле заполнилось, а победителя нет, игра продолжается.

Ваша задача — найти оптимальную стратегию для этой игры.

Формат входных данных

В первой строке задано целое число t ($1 \leq t \leq 10^4$) — число наборов входных данных.

Каждый набор входных данных занимает четыре строки. В первой строке записана заглавная английская буква «X» или «O» — игрок, который сейчас ходит. Следующие три строки описывают поле, по три символа в строке: «X», «O» или «.» (пустая клетка).

Гарантируется, что позиция может возникнуть в игре и победитель ещё не определён.

Формат выходных данных

Для каждого набора входных данных выведите четыре строки. В первой строке выведите оценку позиции: вероятность победы игрока, который сейчас ходит, при оптимальной игре. Абсолютная погрешность не должна превышать 10^{-6} . В следующих трёх строках выведите поле из входных данных, заменив клетку оптимального хода символом «#».

Можно доказать, что существует стратегия, при которой из любой позиции, как бы ни играл соперник, игра с вероятностью 1 заканчивается, а вероятность победы не меньше оценки этой позиции. Засчитывается любой ход, с которого может начинаться такая стратегия, даже если при повторении той же позиции стратегия сделает другой ход.

Пример

стандартный ввод	стандартный вывод
3	0.800000000
X	XX#
XX.	...
...	...
...	0.666666667
O	XX.
XX.	OO#
OO.	...
...	0.666666667
X	X#X
XOX	XOO
XOO	OXX
OXX	

Задача О. Учимся прыжкам

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Лео недавно купил скакалку и теперь учится через неё прыгать. Он составил список упражнений и сначала хочет освоить все из них. Он начинает в некоторой позиции в своей комнате и пытается проделать упражнения одно за другим (в некотором порядке, каждое ровно один раз). Пока что у него не получается даже самое простое, но это ненадолго! Сейчас он сидит в своём кресле и думает: как упорядочить упражнения так, чтобы он мог выполнить их подряд?

Есть одна проблема: Лео живёт в довольно тесной комнате, длина которой всего пять метров. У каждого упражнения есть заданный сдвиг a : если Лео начинает выполнять его в позиции s (то есть стена находится в s метрах позади него), то он заканчивает в позиции $s + a$. Чтобы успешно выполнить упражнения, все позиции во время прыжков должны лежать строго внутри комнаты. Другими словами, они должны быть равны 1, 2, 3 или 4.

Как переупорядочить упражнения, чтобы эти ограничения были соблюдены?

Формат входных данных

В первой строке задано целое число t ($1 \leq t \leq 1000$) — число наборов входных данных. Далее следуют описания наборов входных данных.

В первой строке каждого набора входных данных заданы два целых числа n и s ($1 \leq n \leq 2 \cdot 10^5$; $1 \leq s \leq 4$) — число упражнений, которые Лео хочет освоить, и его начальная позиция.

Во второй строке заданы n целых чисел, описывающих сдвиги упражнений $a_1, \dots, a_n$ ($-3 \leq a_i \leq 3$).

Сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных выведите «YES», если существует подходящая перестановка упражнений, и «NO» в противном случае. Каждую букву можно выводить в любом регистре.

В первом случае также выведите такую перестановку a . Если возможных ответов несколько, принимается любой из них.

Пример

стандартный ввод	стандартный вывод
3	NO
3 2	YES
1 1 1	-1 2 -3 0 3 -2 1
7 3	NO
-3 -2 -1 0 1 2 3	
2 2	
3 -2	