

Задача А. Join

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 64 мегабайта

В этом году ЛКШ проводится на новой базе. Её территория — прямоугольник $n \times m$, на котором расположены квадратные домики (один домик занимает квадратик 1×1).

Так как база совсем новая и между домиками ещё не проложены тропинки, оргкомитет ЛКШ решил проложить их своими силами. Так как ресурсы ограничены, было решено соединять только соседние домики (соседними считаются домики, граничащие по вертикали или горизонтали), причём так, чтобы от любого домика до любого можно было дойти единственным способом.

Эту задачу поручили Тоше — младшему преподавателю параллели A' и он уже готов с энтузиазмом приступить к её выполнению. Но сначала Антон хочет узнать, сколькими способами может это сделать. Конечно, такую простую задачу он может решить и сам, но уже поздно, а Тоша очень устал.

Помогите Тоше, чтобы он успел выспаться перед завтрашними парами.

Формат входных данных

В первой строке даны два целых числа n и m ($1 \leq n, m \leq 9$) — размеры базы. В следующих n строках содержатся по m символов, описывающих карту. Символ "." обозначает домик, а "*" — полянку для игры во фрисби.

Гарантируется, что на базе есть хотя бы один домик.

Формат выходных данных

Выведите одно целое число — число способов соединить домики по модулю 10^9 .

Примеры

стандартный ввод	стандартный вывод
2 2	4
2 2 *. .*	0

Задача В. Задобрите Инти

Имя входного файла: `makeintihappy.in`
Имя выходного файла: `makeintihappy.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Как вы помните, племя Майя вымерло, племя Йюля вынуждено было распасться и покинуть обжитые территории, и вот теперь племени Авгуйя тоже угрожает опасность. Задоблив бога дождя Тлалока, они напрочь забыли про бога солнца Инти, который будучи обделённым вниманием грозиться ниспослать испепеляющую жару, если жители племени не порадуют его божественный взор новой коллекцией симпатичных узоров, выложенных из огромных гранитных плит, покрашенных в чёрный и белый цвета. Напомним, что симпатичным узором называется прямоугольник шириной W и высотой H ни какой квадрат 2 на 2 которого не покрашен в один цвет.

В данной задаче таскать гранитные плиты Вам не придётся, однако потребуется определить K -й лексикографически симпатичный узор. Один узор будем считать лексикографически меньше другого, если при просмотре узоров по столбцам сверху вниз найдётся такая позиция, что все плиты рассмотренные ранее совпадают попарно по цвету, а плита на данной позиции у первого узора белая, а у второго — чёрная.

Формат входных данных

Во входном файле заданы числа W , H и K . ($1 \leq W \leq 1000$, $1 \leq H \leq 10$, $1 \leq K \leq 10^{18}$)

Формат выходных данных

Выведите требуемый симпатичный узор, обозначая белые плиты символом $'w'$, а чёрные — символом $'b'$. Если не существует узора с таким номером — выведите "Impossible" (без кавычек).

Примеры

<code>makeintihappy.in</code>	<code>makeintihappy.out</code>
5 1 17	bwww
7 4 1234567	wbwbwb wbbwwb wwwbbwb wbwbwb

Задача С. Песочные часы

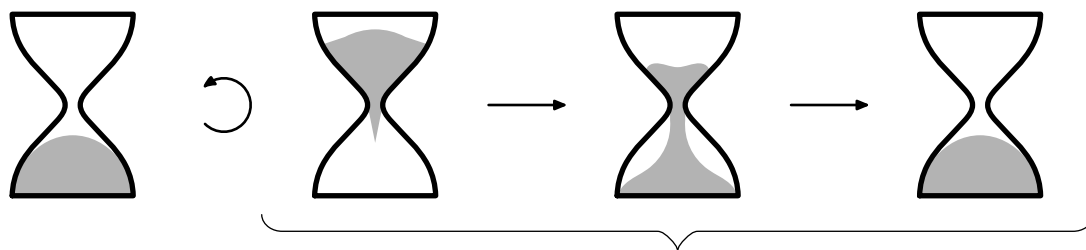
Имя входного файла:	clocks.in
Имя выходного файла:	clocks.out
Ограничение по времени:	1 секунда
Ограничение по памяти:	4 мегабайта

Обратите внимание на нестандартное ограничение по памяти в этой задаче.

Каждый раз, приходя из школы, Петя разогревает себе суп. Петя давно установил, что для достижения оптимальной температуры, суп надо греть в течении ровно t минут. Однажды у Пети в часах села батарейка. И тут неожиданно выяснилось, что это были единственные часы в доме.

Порывшись на чердаке, Петя нашел двое старых песочных часов — на a и на b минут соответственно. Каждые песочные часы состоят из двух половинок, одна из которых исходно заполнена песком.

Для того, чтобы использовать часы, их ставят на одно из оснований, при этом песок из верхней половины начинает постепенно пересыпаться в нижнюю.



Песок полностью пересыпается из верхней половины в нижнюю за a минут в первых часах и за b минут во вторых

Песок пересыпается равномерно и с одинаковой скоростью, вне зависимости от количества песка, оставшегося в верхней половине. В первых часах весь песок пересыпается за a минут, во вторых — за b минут.

В тот момент, когда Петя ставит суп на огонь, весь песок в каждом часах находится в нижней половине. В этот момент Петя может перевернуть какие-либо часы, либо и те и другие сразу. Далее Петя может переворачивать часы в момент, когда в одних из них заканчивается пересыпаться песок. В один из таких моментов Петя должен снять суп с плиты.

Петя хочет узнать, как ему действовать, чтобы снять суп с плиты ровно через t минут.

Формат входных данных

Во входном файле заданы целые числа a , b и t ($1 \leq a, b \leq 500$, $1 \leq t \leq 10^5$).

Формат выходных данных

Выведите последовательность инструкций для Пети. Каждая инструкция — это пара **<событие>**: **<действие>**.

События бывают трех типов:

- **Initially** — начальный момент времени, в последовательности должна быть ровно одна инструкция, помеченная этим событием, она должна быть первой;
- **When A stops** — когда заканчивается пересыпаться песок в первых часах;
- **When B stops** — когда заканчивается пересыпаться песок во вторых часах.

Действия бывают четырех типов:

- **flip A** — перевернуть первые часы;
- **flip B** — перевернуть вторые часы;
- **flip A and B** — перевернуть и те и другие часы;
- **ready** — снять суп с плиты, инструкция с таким действием должна быть ровно одна, она должна быть последней.

Если подогреть суп с использованием этих песочных часов не удастся, выведите в выходной файл одно слово — “Impossible”.

Примеры

clocks.in	clocks.out
5 7 9	Initially: flip A and B When A stops: flip A When B stops: flip A and B When A stops: ready
2 4 11	Impossible

Задача D. Диаграммы Юнга наносят ответный удар

Имя входного файла: nice3.in
Имя выходного файла: nice3.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Компания BrokenTiles планирует заняться выкладыванием во дворах у состоятельных клиентов узор из черных и белых плиток, каждая из которых имеет размер 1×1 метр. Известно, что дворы всех состоятельных людей имеют наиболее модную на сегодня форму прямоугольника $n \times m$ метров.

Однако при составлении финансового плана у директора этой организации появилось целых две серьезных проблемы: во первых, каждый новый клиент очевидно захочет, чтобы узор, выложенный у него во дворе, отличался от узоров всех остальных клиентов этой фирмы, а во вторых, этот узор должен быть экстравагантным.

Как показало исследование, узор является экстравагантным, если в нем нигде не встречается диаграмма Юнга для суммы чисел из разложения числа k на простые множители, полностью покрытого плитками одного цвета. Например для числа 20 это $5 + 2 + 2$, а для 16 это $2 + 2 + 2 + 2$.

По каким-то странным причинам клиенты хотят только узоры для $k = 4$. Для составления финансового плана директору необходимо узнать, сколько клиентов он сможет обслужить, прежде чем экстравагантные узоры данного размера закончатся. Помогите ему!

Формат входных данных

На первой строке входного файла находятся два натуральных числа n и m . $1 \leq n \leq 10^{100}$, $1 \leq m \leq 5$, $1 \leq p \leq 10000$.

Формат выходных данных

Выведите в выходной файл единственное число — количество различных экстравагантных узоров, которые можно выложить во дворе размера $n \times m$ по модулю p . Узоры, получающиеся друг из друга сдвигом, поворотом или отражением считаются различными.

Примеры

nice3.in	nice3.out
2 2 20	14
3 3 7	0

Задача Е. Калила и Димна на лесозаготовках

Имя входного файла: `lumber.in`
Имя выходного файла: `lumber.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Спонсор сегодняшней задачи — codeforces round 189. Codeforces — мечты сбываются!

Калила и Димна — два шакала. Они живут в огромных джунглях. Однажды шакалы решили устроиться на завод лесозаготовки и подработать.

Управляющий завода хочет, чтобы они отправились в джунгли и срубили n деревьев высотой a_1, a_2, \dots, a_n . Для этого Калила и Димна купили цепную пилу в магазине. Каждый раз, когда они используют пилу на дереве номер i , они уменьшают высоту этого дерева на единицу. Каждый раз Калила и Димна должны заправить пилу для использования. Цена заправки зависит от того, какие деревья полностью спилены (дерево считается полностью спиленным, если его высота равна 0). Если максимальный идентификатор полностью срубленного дерева равняется i (первоначально это дерево имело высоту a_i), то цена заправки пилы равняется b_i . Если ни одно дерево не срублено полностью, то заправлять пилу запрещается. Изначально пила заправлена. Известно, что для каждого $i < j$, $a_i < a_j$ и $b_i > b_j$, а также $b_n = 0$ и $a_1 = 1$.

Калила и Димна хотят полностью срубить все деревья с минимальными затратами. Они ждут Вашей помощи! Поможете?

Формат входных данных

В первой строке записано целое число n ($1 \leq n \leq 10^5$). Во второй строке записано n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$). В третьей строке записано n целых чисел b_1, b_2, \dots, b_n ($0 \leq b_i \leq 10^9$).

Гарантируется, что $a_1 = 1$, $b_n = 0$, $a_1 < a_2 < \dots < a_n$ и $b_1 > b_2 > \dots > b_n$.

Формат выходных данных

В единственной строке должна быть записана минимальная стоимость вырубания всех деревьев.

Примеры

<code>lumber.in</code>	<code>lumber.out</code>
5 1 2 3 4 5 5 4 3 2 0	25
6 1 2 3 10 20 30 6 5 4 3 2 0	138

Задача F. Транспортировка кошек

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Zxc960115 содержит большое хозяйство. Он кормит m милых кошечек и держит у себя p кормильщиков. Через ферму проходит прямая дорога, а вдоль дороги расположено n холмов, пронумерованных от 1 до n , слева направо. Расстояние от холма i до $i-1$ равняется d_i метров. Кормильщики живут на холме 1.

Однажды кошечкам захотелось порезвиться и они разбежались. Кошка i пошла к холму h_i , дошла до него в момент времени t_i , а затем стала ждать кормильщика на холме h_i . Кормильщики должны собрать всех разбежавшихся кошек. Каждый кормильщик идет прямо от холма номер 1 до холма номер n , не останавливаясь у какого-либо холма, и собирает всех кошек, **ожидающих** на каждом холме. Кормильщики двигаются со скоростью 1 в единицу времени и достаточно сильны, чтобы собрать сколько угодно кошек.

Например, пусть имеется два холма ($d_2 = 1$) и одна кошечка, которая дошла до холма 2 ($h_1 = 2$) в момент времени 3. Тогда, если кормильщик отправится за кошками от холма 1 в момент времени 2 или 3, то он сможет забрать эту кошку. Но если он отправится от холма 1 в момент времени 1, то он не сможет этого сделать. Если кормильщик отправится за кошкой в момент времени 2, то кошка будет ждать его 0 единиц времени, если же он отправится в момент времени 3, то кошка будет ждать его 1 единицу времени.

Ваша задача — составить расписание отправки от холма 1 для кормильщиков так, чтобы общее время ожидания кошек до того как их заберут было минимальным.

Формат входных данных

В первой строке входных данных содержится три целых числа n, m, p ($2 \leq n \leq 10^5, 1 \leq m \leq 10^5, 1 \leq p \leq 100$).

Во второй строке содержится $n - 1$ положительных целых чисел d_2, d_3, \dots, d_n ($1 \leq d_i < 10^4$).

В каждой из следующих m строк содержится по два целых числа h_i и t_i ($1 \leq h_i \leq n, 0 \leq t_i \leq 10^9$).

Формат выходных данных

Выведите целое число, минимальную сумму времен ожидания всех кошек.

Примеры

стандартный ввод	стандартный вывод
4 6 2 1 3 5 1 0 2 1 4 9 1 10 2 10 3 12	3