

Problem A. Balance

Input file: *standard input* или *balance.in*
Output file: *standard output* или *balance.out*
Time limit: 4 seconds
Memory limit: 256 Mebibytes

В столице Бйитландии расположен офис крупной ИТ-компании. Ежегодно компания проводит набор студентов для стажировки. На собеседовании сотрудники компании задают кандидатам несложные задачи типа следующей.

Определим *сбалансированность* целочисленной последовательности как разность между её максимальным и минимальным элементами. Рассмотрим целочисленную последовательность A длины N .

Требуется найти наименьшее целое число k ($1 \leq k \leq N - 1$), для которого максимум из значений сбалансированности последовательностей B ($B_i = A_i$ для всех $1 \leq i \leq k$) и C ($C_i = A_{i+k}$ для всех $1 \leq i \leq N - k$) будет минимальным.

Input

Входной файл состоит из не более, чем 50 тестовых примеров. Каждый тестовый пример расположен на одной строке и начинается с целого числа N ($2 \leq N \leq 4 \cdot 10^5$). Далее следуют N целых неотрицательных чисел, каждое из которых не превосходит $2 \cdot 10^6$. Входной файл заканчивается тестовым примером с $N = 1$, который обрабатывать не следует. Гарантируется, что объём входного файла не превышает 50 мегабайт.

Output

Для каждого тестового примера выведите одно целое число — требуемое значение k .

Example

<i>standard input</i> или <i>balance.in</i>	<i>standard output</i> или <i>balance.out</i>
4 2 3 5 4	2
8 6 8 8 9 2 5 11 4	5
1	

Problem B. Binary key

Input file: *standard input* или `binary.in`
Output file: *standard output* или `binary.out`
Time limit: 2 seconds
Memory limit: 256 mebibytes

Одним из направлений работы крупной байтландской IT-компании являются задачи, связанные с поиском данных.

При оптимизации поиска файлов с графической информацией возникла следующая задача. Пусть дана строка A , состоящая из нулей и единиц, называемая «ключом». Требуется найти, сколько существует различных строк заданной длины, состоящих из нулей и единиц и содержащих A в качестве подстроки.

Input

В первой строке входного файла заданы два целых числа L , L_A — соответственно длина строки, которая должна содержать ключ и длина самого ключа ($1 \leq L_A \leq L \leq 2000$). Во второй строке задан ключ — L_A символов '0' или '1'.

Output

Выведите количество различных двоичных строк длины L , содержащих ключ, по модулю 10^4 .

Examples

<i>standard input</i> или <code>binary.in</code>	<i>standard output</i> или <code>binary.out</code>
4 3 101	4
20 2 11	865

Problem C. Bind it!

Input file: *standard input* или `bindit.in`
Output file: *standard output* или `bindit.out`
Time limit: 2 seconds
Memory limit: 256 mebibytes

Получившая популярность среди сотрудников крупной IT-компании головоломка «Bind it!» устроена следующим образом. На плоскости отмечены 12 различных точек с целыми неотрицательными координатами, не превышающими 6. Одна из этих точек помечена как стартовая (символ ‘s’, остальные обозначены некоторыми строчными буквами из диапазона ‘a’ до ‘z’, причём разные точки обозначены разными буквами). Требуется последовательно соединить все точки, начиная со стартовой, 11 отрезками таким образом, чтобы получившаяся ломаная не имела самопересечений, а также звеньев, параллельных осям координат или прямым $y = x$ и $y = -x$.

Чтобы минимизировать время, которое сотрудники тратят на решение головоломки, руководство компании поручило вам написать программу, которая по заданной конфигурации точек находит решение или сообщает, что головоломка решения не имеет.

Input

Входной файл состоит из не более, чем 50 тестовых примеров. Каждый тестовый пример задаёт одну головоломку и представляет собой 7 строк длиной по 7 символов каждая. i -й символ в j -й строке равен ‘.’, если точка с координатами $(i - 1, j - 1)$ не отмечена, ‘s’, если точка отмечена как стартовая, или символ от ‘a’ до ‘z’ (исключая ‘s’) для остальных точек. Гарантируется, что в каждом тестовом примере будет ровно один символ ‘s’ и ровно 11 попарно различных строчных букв. Тестовые примеры разделены пустой строкой. Последний тестовый пример состоит из одного символа ‘0’ (нуля), который обрабатывать не требуется.

Output

Для каждого тестового примера в отдельной строке выведите ‘impossible!’, если головоломка не имеет решения, или строку из 12 символов, начинающуюся с символа ‘s’, за которым следуют в некотором порядке 11 различных строчных букв (обозначений остальных точек, присутствующих в примере) — последовательность соединения точек. Если решений несколько, выведите лексикографически наименьшую строку.

Example

<i>standard input</i> или bindit.in	<i>standard output</i> или bindit.out
q..rj..z.. .d..... .e....t .b...u. kc.....s.. .s..a..b.c .d.e.f.h..g .ij.k.. 0	suctzbjkreqd impossible!

Problem D. Bonus

Input file: *standard input* или *bonus.in*
Output file: *standard output* или *bonus.out*
Time limit: 2 seconds
Memory limit: 256 mebibytes

В одной крупной IT-компании настала пора распределения премий. Всего в штате компании N сотрудников, включая Генерального Директора. Размер премиального фонда равен N^K байтландских тугриков. На общем собрании Генеральный Директор заявил, что он принял решение выписать себе премию в 1 (один) байтландский тугрик, а оставшиеся деньги распределить поровну между остальными сотрудниками.

Согласно байтландскому законодательству, при выдаче денежных сумм в ведомости фиксируется не только выданная сумма, но и «контрольная запись» — остаток этой суммы от деления на зависящее от месяца целое число M (это делается для того, чтобы усложнить правку документов задним числом).

Вам требуется по заданным N , K и M найти «контрольную запись», которая будет зафиксирована в ведомости для каждого из сотрудников компании (исключая Генерального Директора).

Input

В первой и единственной строке входного файла заданы три целых числа N , K и M ($2 \leq N, K, M \leq 10^9$).

Output

Выведите одно число — требуемую «контрольную запись».

Examples

<i>standard input</i> или <i>bonus.in</i>	<i>standard output</i> или <i>bonus.out</i>
2 2 666	3
10 3 37	0

Problem E. Boulevards

Input file: *standard input* или *boulevards.in*
Output file: *standard output* или *boulevards.out*
Time limit: 2 seconds
Memory limit: 256 mebibytes

В столице Байтландии N перекрёстков, соединённых M бульварами с двусторонним движением так, чтобы между любыми двумя перекрёстками можно было добраться по сети бульваров. Время на движение по каждому бульвару в обе стороны одинаково. Бульвары пересекаются только в начальных и конечных точках; никакие два перекрёстка не соединены непосредственно более, чем одним бульваром.

После окончания рабочего дня двое программистов — сотрудников крупной IT-компании, работающих над одним и тем же проектом, направились домой по бульварам. Каждый из них планирует минимизировать время в пути. С учётом этого они хотят выбрать маршрут таким образом, чтобы как можно большую часть маршрута пройти вместе — обсуждение перспектив развития проекта оказалось весьма увлекательным.

Вам задана карта столицы Байтландии, время, за которое можно пройти каждый бульвар с начала до конца, а также номера перекрёстков, на которых находятся соответственно офис IT-компании, дом первого программиста и дом второго программиста. При этом никакие два из трёх вышечисленных объектов не находятся на одном и том же перекрёстке. Требуется найти наибольшее время, в течение которого программисты смогут идти вместе.

Input

В первой строке выходного файла задано целое число T ($1 \leq T \leq 100$) — количество тестовых примеров. Далее задаются тестовые примеры. Первая строка тестового примера содержит два целых числа N и M — количество перекрёстков и бульваров соответственно ($3 \leq N \leq 2000, N - 1 \leq M \leq \min(N(N - 1)/2, 10^4)$). Перекрёстки занумерованы различными целыми числами от 0 до $N - 1$. Во второй строке заданы номера трёх перекрёстков, на которых расположены соответственно офис, дом первого программиста и дом второго программиста — три попарно различных целых неотрицательных числа, меньших N . Далее идут M строк, задающих бульвары. Каждая из этих строк содержит по три целых числа a_i, b_i, c_i — соответственно номера перекрёстков, которые соединяет бульвар, и время, за которое программисты пройдут этот бульвар ($0 \leq a_i, b_i < N, 0 < c_i \leq 1000$).

Output

Для каждого тестового примера выведите наибольшее время, которое программисты пройдут вместе при условии, что для каждого программиста путь от офиса до дома будет оптимальным по затраченному времени.

Example

<i>standard input</i> или boulevards.in	<i>standard output</i> или boulevards.out
2	100
4 5	0
1 3 0	
1 2 100	
2 3 50	
0 2 40	
1 3 500	
0 1 500	
4 5	
1 3 0	
1 2 100	
2 3 50	
0 2 40	
1 3 10	
0 1 10	

Problem F. Boxes

Input file: *standard input* или `boxes1.in`
Output file: *standard output* или `boxes1.out`
Time limit: 2 seconds
Memory limit: 256 mebibytes

На склад в офисе крупной IT-компании привезли некоторое количество компьютеров. При разгрузке коробки с компьютерами ставили в N расположенных в ряд «блоков» из поставленных одна на другую коробок.

Однако после разгрузки сотрудники службы безопасности сделали замечание, что расположение коробок не соответствует нормам техники безопасности, гласящим, что соседние «блоки» должны отличаться по высоте не более, чем на одну коробку (в том числе если с одной из стороны блока не стоит ни одной коробки, то высота этого блока не может быть больше 1).

Руководство компании решило установить некоторое количество компьютеров на рабочие места сотрудников так, чтобы расположение оставшихся коробок соответствовало нормам техники безопасности. При этом очередной компьютер можно брать только сверху некоторого блока.

Учитывая, что замена компьютера у сотрудника — процесс трудоёмкий и отнимающий рабочее время, руководство хочет минимизировать количество забираемых со склада компьютеров. Ваша задача — вычислить это количество.

Input

В первой строке входного файла задано одно целое число T ($1 \leq T \leq 25$) — количество тестовых примеров. Каждый тестовый пример расположен на одной строке, в которой сначала идёт целое число N ($2 < N \leq 10^5$) — количество блоков, далее следуют N целых неотрицательных чисел, меньших 100 — изначальное количество коробок в каждом блоке. При этом первое и последнее число всегда равны 0.

Output

Для каждого тестового примера выведите одно целое число — минимальное количество компьютеров, которое придётся убрать со склада, чтобы оставшиеся коробки стояли в соответствии с требованиями безопасности.

Example

<i>standard input</i> или <code>boxes1.in</code>	<i>standard output</i> или <code>boxes1.out</code>
1 8 0 3 4 5 7 2 2 0	11