

Problem A. David's Magic

Input file: *standard input* или `davidsmagic.in`
Output file: *standard output* или `davidsmagic.out`
Time limit: 2 second
Memory limit: 64 mebibytes

В XIX веке, когда информация среди населения распространялась существенно медленнее, чем сейчас, средний уровень математических знаний был невысок. Поэтому бродячие фокусники часто демонстрировали числовые фокусы как пример «чтения мыслей на расстоянии».

Один из самых известных байтландских магов того времени Давид Праймфилд демонстрировал такой трюк: он просил одного из зрителей задумать целое число N , просуммировать все числа от 1 до N включительно, кроме одного числа $1 \leq K \leq N$. После чего зритель сообщал получившуюся сумму, а Давид с первой попытки называл N и K .

Напишите программу, которая повторяет фокус Давида.

Input

Входной файл состоит из не более, чем 10^5 тестовых примеров. Каждый тестовый пример расположен на отдельной строке и содержит одно целое число S — сообщённую зрителем сумму ($1 \leq S \leq 2^{56}$). Входной файл завершается строкой с $S = 0$, которую обрабатывать не нужно.

Output

Для каждого тестового примера в отдельной строке выведите два целых числа — количество суммируемых зрителем чисел N и пропущенное число K .

Example

| <i>standard input</i> или <code>davidsmagic.in</code> | <i>standard output</i> или <code>davidsmagic.out</code> |
|---|---|
| 2 | 2 1 |
| 4 | 3 2 |
| 54 | 10 1 |
| 608 | 35 22 |
| 2012 | 63 4 |
| 20120608 | 6344 5732 |
| 0 | |

Problem B. Destination Points

Input file: *standard input* или *destination.in*
Output file: *standard output* или *destination.out*
Time limit: 2 seconds
Memory limit: 64 mebibytes

Во время одной из войн, в которых армия Байтландии участвовала в XIX веке, командующему байтландским экспедиционным корпусом сообщили, что разведчикам удалось добыть важный документ — копию приказа о подготовке армии противника к наступлению.

В приказе содержались указания о выдвижении каждого из M полков противника к новым местам дислокации. Формат каждого указания — “ i -й полк, дислоцированный в населённом пункте H_i , должен прибыть на новое место дислокации через t_i минут после часа X ”.

Командующему известно, что все пункты новой дислокации попарно различны, что войска противника прекрасно вымуштрованы и двигаются на марше с фиксированной скоростью без привалов, при этом тыловые службы для передислокации всегда выбирают кратчайший маршрут, проходящий по существующим дорогам. Также для каждой дороги известно время, за которое полк противника проходит её целиком.

На занятой противником территории оказалось N населённых пунктов, некоторые из этих пунктов соединены дорогами с двусторонним движением так, что любые два населённых пункта соединены напрямую не более, чем одной дорогой и между двумя любыми населёнными пунктами существует путь по сети дорог.

Командующий интересуется, может ли он по имеющимся данным однозначно определить, в каком населённом пункте окажется каждый полк после передислокации?

Input

В первой строке входного файла задано одно целое число T ($1 \leq T \leq 15$) — количество тестовых примеров. В первой строке каждого тестового примера заданы три целых числа M — количество полков противника, упоминаемых в приказе ($1 \leq M \leq 20$), N — количество населённых пунктов на территории, занятой противником, и R — количество дорог между этими пунктами ($2 \leq N \leq 100$, $N - 1 \leq R \leq 500$). Полки пронумерованы числами от 1 до M , населённые пункты — числами от 1 до N . В каждой из последующих R строк заданы дороги, по одной на строку. Каждая дорога задаётся тремя целыми числами a_i , b_i и l_i — соответственно населёнными пунктами, которые соединяет эта дорога, и временем прохождения дороги ($1 \leq a_i \neq b_i \leq N$, $1 \leq l_i \leq 1000$). В последующих M строках заданы указания полкам. j -я строка содержит два целых числа H_j и t_j — соответственно номер города, в котором j -й полк находится изначально, и время, которое полк должен затратить на преодоление кратчайшего пути к месту новой дислокации ($1 \leq H_j \leq N$, $0 \leq t_j \leq 2000$).

Output

Для каждого тестового примера в отдельной строке выведите M целых чисел — места новой дислокации каждого из полков в порядке их нумерации (перечисления во входном файле). В случае, если однозначно вычислить финальную дислокацию невозможно, выведите -1 .

Example

| <i>standard input</i> или destination.in | <i>standard output</i> или destination.out |
|--|--|
| 2 | 2 4 |
| 2 4 4 | -1 |
| 2 1 10 | |
| 4 1 10 | |
| 3 2 8 | |
| 3 4 6 | |
| 1 10 | |
| 2 14 | |
| 1 3 2 | |
| 2 1 6 | |
| 1 3 6 | |
| 1 6 | |

Problem C. Diplomatic mail

Input file: *standard input* или *diplomacy.in*
Output file: *standard output* или *diplomacy.out*
Time limit: 2 seconds
Memory limit: 64 mebibytes

В XIX веке, когда ещё не было изобретено компьютеров и даже механических шифровальных машин, шифровка сообщений, передаваемых по дипломатическим каналам, была довольно сложным делом. Байтландские дипломаты использовали следующий шифр (оставшийся невзломанным до начала XX века).

- Сообщение, состоящее только лишь из прописных букв латинского алфавита, преобразуется в последовательность целых чисел: каждая буква заменяется на свой порядковый номер, начиная с нулевого ('A' — 0, 'B' — 1, 'C' — 2, ..., 'Z' — 25).
- Полученный таким образом вектор целых чисел домножается справа на секретную матрицу M (секретная матрица M — это квадратная матрица, состоящая из целых положительных чисел и обладающая следующим свойством: $M_{i,j} = M_{i-1,j-1}$, где $M_{i,j}$ — элемент, стоящий в матрице в i -ой строке на j -ом месте).
- Каждый элемент вектора, полученного в результате умножения, заменяется на остаток от деления этого элемента на 26, после чего проводится обратное преобразование — числа заменяются на соответствующие буквы латинского алфавита по аналогии с первым шагом. Полученная строка и является зашифрованным сообщением.

Недавно известный байтландский режиссёр решил экранизировать остросюжетный роман, рассказывающий о дипломатических играх конца XIX века. Для сохранения деталей необходимо правильно шифровать передаваемые сообщения. Ваша задача — написать программу, которая бы по входящему сообщению выдавала зашифрованный текст.

Input

В первой строке входного файла содержится сообщение, которое требуется зашифровать — непустая строка, состоящая из заглавных латинских букв. Длина строки N не превосходит $5 \cdot 10^4$.

Начиная со второй строки во входном файле заданы $2N - 1$ целых чисел K_i , разделённых пробелами или переводами строк. Эти числа описывают шифрующую матрицу M следующим образом: Пусть K_j — j -ое из чисел K_i ($1 \leq j \leq 2 \cdot N - 1$, $0 \leq K_j \leq 25$). Тогда, если считать, что нумерация строк и столбцов шифрующей матрицы ведётся с нуля и $M_{0,0}$ — элемент в верхнем левом углу, $M_{N-i,0} = K_i$ при $1 \leq i \leq N$ и $M_{0,i-N} = K_i$ при $N + 1 \leq i \leq 2N - 1$.

Остальные элементы матрицы M определяются её свойством. Гарантируется, что длина любой строки входных данных не превышает 10^5 .

Output

Выведите строку, состоящую только лишь из заглавных букв латинского алфавита, которая представляет собой зашифрованное сообщение.

Examples

| <i>standard input</i> или diplomacy.in | <i>standard output</i> или diplomacy.out |
|--|--|
| BBB 1 2 3 2 1 | GHG |
| TCM 1 2 3 2 1 | VQH |

Problem D. Divide and... sell

Input file: *standard input* или *divide.in*
Output file: *standard output* или *divide.out*
Time limit: 2 seconds
Memory limit: 64 mebibytes

Промышленная революция XIX века в Байтландии, как и в остальном мире, привела к разорению части крупных землевладельцев. Не сумев адаптироваться к новым экономическим условиям, они были вынуждены распродавать земли, когда-то дарованные их предкам королём.

Дарованные королём поместья представляли собой выпуклые N -угольники, при этом чем более знатым был дворянин, тем больше было значение N . Не случайно королевские сады назывались «Круглыми землями»... Так что продавать, согласно законам и обычаям байтландского дворянства, можно было только треугольные участки земли. При этом строительство забора, разделяющего участки, было обязанностью продавца.

Так что часто возникала следующая задача: задан участок в виде выпуклого N -угольника. Требуется построить внутри забор таким образом, чтобы участок оказался разбит на $N - 2$ треугольных участков, а длина забора была бы минимальна.

Input

В первой строке входного файла задано целое положительное число T ($1 \leq T \leq 50$) — количество тестовых примеров. Далее заданы тестовые примеры. Каждый пример расположен на одной строке и начинается с целого числа N ($3 < N \leq 50$) — количества вершин многоугольника, задающего участок, за которым следуют N пар целых чисел, по модулю не превосходящих 10^4 — соответственно x - и y -координаты вершин многоугольника в порядке их обхода против часовой стрелки. Гарантируется, что все внутренние углы многоугольника строго меньше 180 градусов.

Output

Для каждого тестового примера в отдельной строке выведите одно число — минимальную длину забора с точностью 0.01.

Example

| <i>standard input</i> или <i>divide.in</i> | <i>standard output</i> или <i>divide.out</i> |
|--|--|
| 2 | 5.66 |
| 4 6 1 7 5 2 5 1 1 | 23.827 |
| 5 6 -1 4 3 -6 1 -6 -6 0 -9 | |

Problem E. Domino

Input file: *standard input* или *domino.in*
Output file: *standard output* или *domino.out*
Time limit: 2 seconds
Memory limit: 64 mebibytes

В истории Байтландии присутствует курьёзный эпизод, когда командующий байтландской армией спустил изрядную сумму при игре в преферанс, после чего издал приказ, запрещающий карточные игры. Поэтому в XIX веке среди офицеров байтландской армии получила распространение игра в домино. Были придуманы и доминошные пасьянсы.

Вот один из них: берётся N костяшек домино, на концах которых написаны некоторые числа. Задача игрока — расположить все эти доминошки в ряд друг за другом («цепочку») так, чтобы числа на концах любых двух костяшек домино, прилегающих друг к другу, совпадали. Кроме того, число, с которого начался ряд, должно совпадать с числом, на котором ряд закончился.

Рассказывают, что впоследствии этот пасьянс использовали при отборе в разведку для проверки нестандартности мышления: если у двух кандидатов выстроенные независимо цепочки совпадали, то на одно задание их отправлять было нельзя: разведчики с одинаковым мышлением увидят одно и то же...

А сколько вообще существует способов выложить из заданного набора домино подобную цепочку?

Input

В первой строке входного файла задано целое число N ($1 \leq N \leq 15$). В каждой из следующих N строк записаны по два целых числа A_i и B_i ($1 \leq A_i, B_i \leq 7$, $1 \leq i \leq N$) — числа, записанные на концах i -ой костяшки домино.

Output

Выведите единственное целое число — количество различных решений головоломки. При этом костяшки домино с одинаковым набором чисел, расположенные в одинаковом порядке, неразличимы (см. пример 2).

Examples

| <i>standard input</i> или <i>domino.in</i> | <i>standard output</i> или <i>domino.out</i> |
|--|--|
| 3 2 4 4 6 6 2 | 6 |
| 3 6 7 7 7 6 7 | 3 |

Problem F. Duel

Input file: *standard input* или `duel.in`
Output file: *standard output* или `duel.out`
Time limit: 5 seconds
Memory limit: 64 mebibytes

В XIX веке в Байтландии был распространён обычай «коллективной дуэли»: в случае, когда N человек не сошлись, например, во взглядах на одно место из блаженного Августина, они собирались на пустыре около какой-либо речки, после чего по команде секунданта одновременно производили по одному выстрелу в случайного оппонента, при этом выбор любого из $N - 1$ оппонентов был равновероятен.

В художественной литературе того времени популярным сюжетом была история про l человек, которые во время такой дуэли выстрелили друг в друга «по кругу» — первый выстрелил во второго, второй в третьего и так далее: $l - 1$ -й — в l -го, а l -й — в первого.

Литературоведы считают, что причина этому — заимствование, так как шансы на то, что подобное произойдёт, невелики. Для обоснования своей позиции они попросили Вас найти вероятность такого события для заданного l .

Input

В первой строке входного файла задано одно целое число T ($0 \leq T \leq 30$) — количество тестовых приемров. Каждый тестовый пример состоит из одной строки, в которой записаны по два целых числа N — количество дуэлянтов и l — требуемое количество тех, кто выстрелил «по кругу» ($2 \leq N \leq 10^7$, $2 \leq l \leq N$).

Output

Для каждого тестового примера выведите в отдельной строке требуемую вероятность с точностью 10^{-6} .

Example

| <i>standard input</i> или <code>duel.in</code> | <i>standard output</i> или <code>duel.out</code> |
|--|--|
| 2 | 1 |
| 2 2 | 0.6296296296296 |
| 4 2 | |