

Problem A. Money Exchange

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

Кроме основной денежной единицы Байтландии — байтландского тугрика — в стране используются единицы меньшего номинала. При этом исторически существует N номиналов монет, начиная с однотугриковой, выбранных так, что каждая большая монета номинирована в монетах следующего номинала (то есть, например, монета в 1 тугрик номинирована как 7 центов, в 1 цент — как 3 гроша, в 1 грош — как 6 сантимов).

Известный биржевой игрок Джордж прибыл в Байтландию и собирается провести ряд валютнообменных операций. Так как курсы валют в различных банках обозначены с использованием различных номиналов, то «на лету» понять, какой из курсов является наиболее высоким (и выгодным для продажи), а какой — наиболее низким (и выгодным для покупки), Джордж не может.

Поэтому он заказал Вам программу, которая по списку курсов американского доллара в различных банках определяет разницу между наиболее высоким и наиболее низким курсом, выраженную в монетах наименьшего номинала.

Input

В первой строке входного файла задано целое число T ($1 \leq T \leq 20$) — количество тестовых примеров. В первой строке каждого тестового примера записаны два целых числа — количество различных номиналов монет N ($2 \leq N \leq 7$) и количество B банков, работающих с долларом ($2 \leq B \leq 10$). В следующей строке заданы $N - 1$ целых положительных чисел, не превосходящих 10. i -е число обозначает номинал i -й монеты, выраженный в $i + 1$ -ых монетах. Далее следует B строк, каждая из которых содержит N целых неотрицательных чисел b_j , не превосходящих 10. j -е из этих чисел обозначает количество j -х в порядке уменьшения номинала монет, входящих в курс американского доллара для данного банка (то есть за доллар дают b_1 тугриков плюс b_2 следующих за тугриком монет... плюс b_N монет минимального номинала).

Output

Для каждого тестового примера выведите в отдельной строке одно целое число — разницу между максимальным и минимальным курсом доллара, выраженную в монетах минимального номинала.

Example

<i>standard input</i> или <i>change.in</i>	<i>standard output</i> или <i>change.out</i>
2	16
2 3	49
3	
0 10	
1 0	
6 1	
3 4	
2 6	
2 2 2	
4 0 0	
1 10 1	
7 0 3	

Problem B. Coin game

Input file: *standard input* или *coingame.in*
Output file: *standard output* или *coingame.out*
Time limit: 3 seconds
Memory limit: 64 mebibytes

Как вы знаете, в Байтландии используется N монет различного достоинства. Поэтому там распространена следующая игра: первоначально на прямой отмечаются N точек («позиций»), после чего выставляются N монет различного достоинства так, что центр каждой монеты совпадает с одной из отмеченных точек, и никакие две монеты не лежат друг на друге. Расстояния между точками достаточны для того, чтобы монеты не соприкасались.

В процессе игры монеты можно класть друг на друга так, чтобы монета с меньшим номиналом всегда лежала на монете с большим номиналом. Если на какой-то позиции лежит несколько монет, то верхняя монета называется «доступной».

Ход заключается в переносе некоторой «доступной» монеты в соседнюю позицию; ход разрешён в случае, если позиция либо свободна, либо занята одной или несколькими монетами с большим номиналом, чем переносимая монета. В этом случае монета кладётся поверх уже занимающих эту позицию монет.

Требуется найти наименьшее количество ходов, за которое можно расположить монеты слева направо по возрастанию, действуя по правилам игры.

Input

В первой строке входного файла задано одно целое число N ($2 \leq N \leq 7$) — количество монет. В следующей строке заданы N попарно различных целых чисел — начальная расстановка монет, перечисленных слева направо. Монеты занумерованы целыми числами от 1 до N по возрастанию номиналов.

Output

Если расположить монеты слева направо по возрастанию невозможно, выведите -1 . В противном случае выведите одно целое число — минимальное количество ходов, за которое можно расположить монеты слева направо по возрастанию.

Example

<i>standard input</i> или <i>coingame.in</i>	<i>standard output</i> или <i>coingame.out</i>
3 3 1 2	16
2 2 1	-1

Problem C. Collect A Boxes

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

Сложная валютная система Байтландии приводит к тому, что на складах монетного двора постоянно хранятся ящики с мелочью. Для обслуживания складов Государственный Банк Байтландии заказал у крупной ИТ-компании новейшую автоматизированную систему обслуживания, составным элементом которой является робот-кладовщик.

Склад представляет собой прямоугольное помещение $M \times N$ без внутренних стен, разделённое на клетки размером 1×1 . В некоторых клетках размещены ящики, причём в каждой клетке может быть не более одного ящика.

В одной из не занятых ящиками клеток склада находится лифт. В момент времени 0 из лифта выходит робот, задача которого — загрузить все ящики в лифт. Робот может двигаться вправо, влево, вверх или вниз (если не мешают стены склада). В момент, когда робот оказывается на клетке с ящиком, он может или взять ящик, или продолжить движение. Одновременно робот может нести только один ящик. При этом в случае, если робот уже несёт ящик, он не помещается в клетку, в которой расположен другой ящик (и соответственно, не может проходить через такие клетки).

Как только робот с ящиком оказывается в клетке с лифтом, он немедленно ставит ящик в лифт. Лифт является достаточно просторным, чтобы вместить и робота, и все ящики. Ставить ящик где-либо, кроме лифта, робот не может.

На одно перемещение между клетками робот затрачивает одну секунду, временем на погрузку или выгрузку ящиков можно пренебречь. За какое минимальное время робот сумеет загрузить все ящики в лифт?

Input

Первая строка входного файла содержит одно целое число — количество тестовых примеров T ($1 \leq T \leq 100$). Далее задаются тестовые примеры. В первой строке каждого тестового примера заданы два целых числа M и N — размеры склада ($1 \leq M, N \leq 500$). Каждая из последующих M строк содержит по N символов — эти строки описывают карту склада. i -й символ в j -й строке равен 'L', если в соответствующей клетке находится лифт, 'B', если в ней находится ящик, и '.', если клетка пуста. Гарантируется, что в каждом тестовом примере ровно одна клетка занята под лифт.

Output

Для каждого тестового примера выведите одно целое число — минимальное время, за которое робот сможет загрузить все ящики в лифт.

Example

<i>standard input</i> или <i>collect.in</i>	<i>standard output</i> или <i>collect.out</i>
2	18
3 3	0
BVB	
.L.	
B.B	
1 2	
L.	

Problem D. Conference

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

Благодаря сохранившейся в течение сотен лет валютной системе Байтландия весьма популярна среди нумизматов. Редкие экземпляры монет различных номиналов всё ещё находятся в обращении, так что шанс «поймать» раритет довольно велик.

На конференцию нумизматов, организованную в столице Байтландии, было приглашено N участников. Некоторые из них знакомы друг с другом, причём если участник A знает участника B , то участник B знает участника A . После конференции оказалось, что ситуация изменилась следующим образом: если до конференции нумизмат A_1 был знаком с нумизматом A_2 , нумизмат A_2 — с нумизматом A_3 , нумизмат A_3 — с нумизматом A_4 и так далее до A_n , то после конференции все нумизматы A_i стали знакомы друг с другом.

Вам задан список пар участников, которые были знакомы друг с другом до конференции. Требуется отвечать на запросы типа «знакомы ли участники X и Y друг с другом после конференции»?

Input

В первой строке входного файла заданы три целых числа: N — количество участников конференции ($1 \leq N \leq 10^5$), M — количество пар нумизматов, знакомых друг с другом до начала конференции ($1 \leq M \leq 2 \cdot 10^5$), и Q — количество запросов ($1 \leq Q \leq 10^5$). Участники нумеруются последовательными целыми числами от 1 до N .

В последующих M строках заданы пары (a_i, b_i) номеров участников, знакомых друг с другом на момент начала конференции ($1 \leq a_i, b_i \leq N$, $a_i \neq b_i$). В последующих Q строках заданы запросы. Каждый запрос состоит из двух номеров участников x_j и y_j ($1 \leq x_j, y_j \leq N$, $x_j \neq y_j$).

Output

Для каждого запроса выведите в отдельной строке “Yes”, если нумизматы, указанные в запросе, знакомы друг с другом на момент окончания конференции, и “No” в противном случае.

Example

<i>standard input</i> или <i>conference.in</i>	<i>standard output</i> или <i>conference.out</i>
5 3 2	No
2 1	Yes
3 1	
3 2	
4 3	
2 1	

Problem E. Create A Progression

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

Наличие неудобной в использовании валютной системы не устраивает многих жителей Байтландии, так что недавно была организована «Партия финансовых реформ», целью которой стало изменение валютной системы на более удобную. Партия предлагает вместо существующих ввести монеты K номиналов, которые образовывали бы арифметическую прогрессию.

Один из членов партии подготовил вариант законопроекта. Однако при наборе текста в него вкралась ошибка и некоторые номиналы «выпали» из списка. Требуется по отсортированному по возрастанию списку номиналов выяснить, какое минимальное количество номиналов надо добавить, чтобы получившаяся последовательность образовывала арифметическую прогрессию.

Input

В первой строке входного файла задано целое число M — количество номиналов, перечисленных в тексте законопроекта ($1 \leq M \leq 100$). Во второй строке через пробел записаны M номиналов — целые числа K_i ($1 \leq i \leq M$, $1 \leq K_i \leq 1000$, $K_i < K_{i+1}$).

Output

Выведите единственное целое число — минимальное количество номиналов монет, которые нужно добавить в перечисленную в законопроекте последовательность так, чтобы все числа последовательности стали образовывать арифметическую прогрессию.

Examples

<i>standard input</i> или <i>create.in</i>	<i>standard output</i> или <i>create.out</i>
1 3	0
3 2 3 5	1

Problem F. Cryptosystem

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

По заказу Государственного Банка Байтландии в рамках внедрения системы электронных платежей специалисты по защите информации одной крупной IT-компании разработали усовершенствованную криптографическую систему. В процессе проверки её устойчивости возникла следующая задача: задано целое положительное число N , требуется найти все целые положительные числа M , такие, что сумма всех попарно различных делителей числа M равна N .

Input

В первой строке входного файла задано одно целое число T ($1 \leq T \leq 100$) — количество тестовых примеров. Каждый тестовый пример расположен в отдельной строке и состоит из одного целого числа N ($0 < N \leq 10^9$) — заданной суммы делителей.

Output

Для каждого тестового примера выведите в отдельной строке -1 , если данное число не является суммой всех попарно различных делителей ни для какого целого положительного числа, в противном случае выведите в порядке возрастания все целые положительные числа, для которых данное число является суммой всех попарно различных делителей. Числа в одной строке разделяйте пробелами.

Гарантируется, что объём выходного файла не будет превышать 1 мегабайта.

Example

<i>standard input</i> или <i>crypto.in</i>	<i>standard output</i> или <i>crypto.out</i>
5	2
3	-1
5	6 11
12	28 39
56	60 78 92 123 143 167
168	