Problem A. Arithmetic Rectangle (Division 1 Only!)

Input file:	Standard input
Output file:	Standard output
Time limit:	7 seconds
Memory limit:	256 mebibytes

В каждой из клеток шахматной доски $n \times m$ записано одно целое число. Назовём *арифметическим прямоугольником* прямоугольник, расположенный на полях доски, для которого числа в каждой строке и каждом столбце составляют арифметическую прогрессию. По заданному расположению чисел на доске определите, из какого наибольшего количества клеток может состоять арифметический прямоугольник.

В примере на иллюстрации наибольший арифметический прямоугольник состоит из девяти клеток.

5	3	5	7
2	4	4	4
3	5	3	1
6	3	2	4

Input

В первой строке входного файла задано целое число t ($1 \le t \le 10\,000$) — количество тестовых примеров.

Описание каждого тестового примера начинается со строки, содержащей два целых числа n и m ($1 \le n, m \le 3\,000$) — размерность доски. В каждой из последующих n строк заданы m целых чисел из диапазона $[0, 10^9]$ — числа, размещённые на доске. Гарантируется, что объём входного файла не превышает 20 MiB.

Output

Для каждого тестового примера в отдельной строке выведите одно число — количество клеток в наибольшем арифметическом прямоугольнике, который может быть построен для доски из данного тестового примера.

Examples

Standard input	Standard output
2 4 4 5 3 5 7 2 4 4 4 3 5 3 1 6 3 2 4 2 3 0 1 2 1 2 3	9 6

Problem B. Bytean Road Race (Division 1 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	6 seconds
Memory limit:	256 mebibytes

В центре Байттауна ежегодно проводится Большая Байттаунская Гонка.

Улицы Байттауна идут в одном из двух направлений: или с севера на юг, или с запада на восток. Трасса гонки проложена по частям некоторых улиц следующим образом: на карте отмечается n перекрёстков и m отрезков между двумя соседними перекрёстками (как вертикальных, так и горизонтальных). Никакие два различных отрезка не пересекаются вне перекрёстков. Отмеченные перекрёстки пронумерованы числами от 1 до n. Гонка начинается на перекрёстке с номером 1 и заканчивается на перекрёстке с номером n. Каждый гонщик может выбирать маршрут самостоятельно, однако гонщикам разрешено двигаться только на юг и на восток и исключительно по отрезкам, отмеченным на карте. Отрезки и перекрёстки отмечены таким образом, чтобы, следуя правилам гонки, участник мог попасть в любое отмеченное место (как на перекрёсток, так и во внутреннюю точку отмеченного отрезка), и чтобы из любого отмеченного места, участник мог попасть на финиш гонки.

Организаторы планируют разместить рекламные баннеры фирм-спонсоров гонки так, чтобы ни один гонщик не мог видеть баннер одного и того же спонсора дважды. Поэтому для некоторых пар перекрёстков требуется установить, существует ли возможный маршрут участника гонки, проходящий через оба этих перекрёстка.

Input

Первая строка входного файла содержит три целых числа n, m и k ($2 \le n \le 100\,000$, $1 \le m \le 200\,000$, $1 \le k \le 300\,000$ — количество отмеченных перекрёстков, количество отмеченных отрезков и количество пар пересечений, для которых требуется проверить существование описанного в условии задачи маршрута.

Последующие n строк описывают расположение перекрёстков. В i-й из этих строк заданы координаты i-го перекрёстка, заданные двумя целыми числами x_i, y_i ($-10^9 \le x_i, y_i \le 10^9$). При этом $x_1 \le x_n$ and $y_1 \ge y_n$.

Никакие два перекрёстка не совпадают. Оси координат направлены так, что ось OX направлена на восток, а ось OY — на север.

Каждая из последующих m строк содержит описание одного отрезка. i-я из этих строк содержит пару целых чисел a_i, b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$) — номера перекрёстков, соединяемых этим отрезком. Все отрезки являются или вертикальными, или горизонтальными и могут пересекаться только в начале или в конце (которые, по построению, являются перекрёстками).

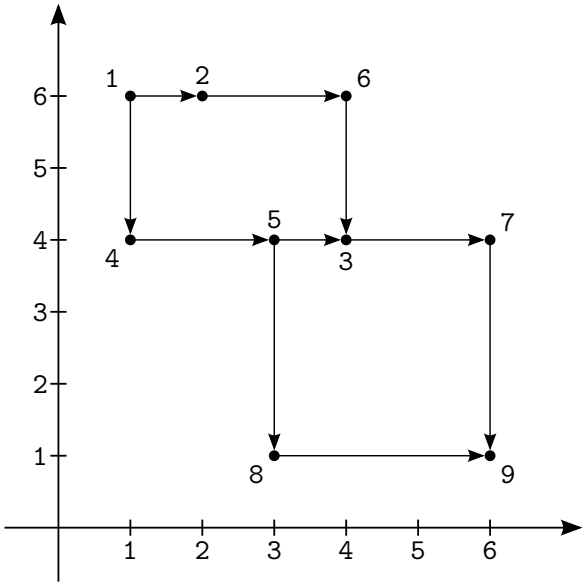
Последующие k строк содержат описания пар перекрёстков, для которых надо проверить существование корректной траектории участника гонки, проходящей через оба этих перекрёстка. i-я из этих строк содержит два целых числа p_i, q_i ($1 \le p_i, q_i \le n$, $p_i \ne q_i$) — номера соответствующих перекрёстков.

Output

Для каждого тестового примера выведите "`TAK`", если возможен маршрут участника гонки, проходящий через оба перекрёстка p_i и q_i (в любом порядке). Иначе выведите текст "`NIE`".

Examples

Standard input	Standard output
9 10 4 1 6 2 6 4 4 1 4 3 4 4 6 6 4 3 1 6 1 1 2 4 1 2 6 3 6 5 4 5 3 5 8 3 7 7 9 9 8 4 8 2 5 8 7 7 6	TAK NIE NIE TAK



Problem C. Will It Stop?

Input file:	Standard input
Output file:	Standard output
Time limit:	2 seconds
Memory limit:	256 mebibytes

Прогуливаясь в районе библиотеки Варшавского Университета, некий турист наткнулся на граффити: кусок программы, под которым был написан вопрос "Она остановится?"

Задача заинтересовала туриста, и он решил подумать над ней по возвращении домой. К сожалению, когда он переписывал кусок кода, он сделал ошибку и написал следующее:

while $n > 1$ **do**
  if $n \bmod 2 = 0$ **then**
    $n := n/2$
  else
    $n := 3 \cdot n + 3$

Сейчас он пытается понять, для каких начальных значений переменной n программа, которую он написал, остановится. Считать, что значение n в процессе выполнения программы может принимать любые значения без угрозы переполнения.

Input

В первой и единственной строке входного файла задано одно целое число n ($2 \le n \le 10^{14}$).

Output

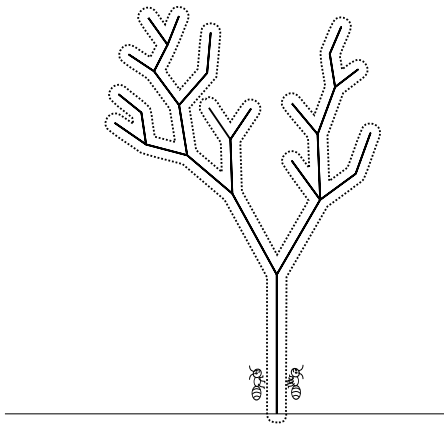
В первой и единственной строке выходного файла выведите "TAK", если программа когда-нибудь остановится для заданного значения n, и "NIE" в противном случае.

Examples

Standard input	Standard output
4	TAK

Problem D. Ants (Division 1 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	30 seconds
Memory limit:	24 mebibytes



Муравьи, равно как и составители задач по программированию, любят деревья. Нам дано дерево с двумя муравьями, ползущими по нему — Левым муравьём и Правым муравьём так, как это указано на рисунке, расположенном выше (муравьи двигаются вдоль пути, обозначеного пунктирной линией).

Муравьи начинают своё путешествие снизу ствола по двум его противоположным сторонам. Левому муравью требуются 2 секунды для того, чтобы пройти по одному ребру дерева при движении от корня (вверх), и 1 секунда для того, чтобы пройти по одному ребру дерева при движении к корню (вниз).

Правый муравей двигается как вверх, так и вниз в два раза быстрее. Когда два муравья встречаются, они оба разворачиваются и начинают двигаться в противоположном направлении. Если какой-нибудь муравей спустится на землю, он немедленно начинает путешествие с противоположной стороны ствола.

Ваша задача — вычислить момент, когда муравьи встретятся во второй раз. При решении задачи размерами муравьёв пренебречь.

Input

В первой строке входного файла задано целое число t ($1 \le t \le 1000$) — количество тестовых примеров.

Каждый тестовый пример состоит из двух строк. Первая строка содержит чётное число n ($2 \le n \le 10^8$), обозначающее количество рёбер дерева.

Во второй строке задано само дерево. Дерево задаётся строкой, состоящей из $\frac{n}{2}$ символов, обозначающих $2n$ — битовое двоичное число, записанное в шестнадцатеричной системе счисления (используя цифры и строчные буквы от `a` до `f`).

Это число описывает путь Левого муравья по всему дереву в предположении, что Правый муравей не движется. Последовательные биты этого числа (начиная с самого левого) обозначают направление движения по соответствующему ребру: 1, если движение идёт от корня, и 0, если движение идёт к корню. При этом дерево имеет ствол, то есть непосредственно из корня выходит ровно одно ребро.

Размер входного файда не превышает 50 MiB, что существенно превосходит объём памяти, выделяемый Вашей программе.

Output

Для каждого тестового примера в отдельной строке выведите момент второго разворота муравьёв (в секундах), заданный в виде несократимой дроби `p/q` (без пробелов вокруг знака '/'), где p и q — целые положительные числа (если ответ целый, то, очевидно, $q = 1$).

Examples

Standard input	Standard output
1 28 fb1da30d1b7230	282/5

Приведённые в примере данные соответствуют иллюстрации и кодируются в следующую последовательность бит:

```
1111 1011 0001 1101 1010 0011 0000 1101 0001 1011 0111 0010 0011 0000
```

Problem E. Gophers

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	13 seconds
Memory limit:	256 mebibytes

Адвокат Дик Дастардли решил поглумиться над байтландскими сусликами. Эти маленькие зверюшки живут в норах. Часто норы строятся таким образом, чтобы все они лежали на одной прямой. Дик нашёл подобную прямую с n норами, расположенными вдоль неё. Занумеруем эти норы с запада на восток последовательными натуральными числами от 1 до n с запада на восток.

Он планирует использовать m имеющихся у него CD-плееров, поставить на каждом альбом группы «Байтлз» и распределить плееры вдоль прямой. Музыка из CD-плеера мешает спать всем сусликам, находящимся в норах, расположенных не далее l метров от плеера.

Далее Дик собирается время от времени передвигать свои плееры следующим образом: в i-й день Дик берёт плеер, расположенный на расстоянии p_i метров от норы с номером 1 и переносит его на расстояние r_i метров от этой норы.

Для каждого передвижения плееров определите количество нор, в которых при новой расстановке плееров суслики не смогут спать.

Input

В первойт строке входного файла заданы четыре целых числа n, m, d и l ($2 \le n, m \le 500\,000$, $1 \le d \le 500\,000$, $1 \le l \le 10^9$) — количество нор, количество CD-плееров, количество дней, в которые Дик передвигает плееры и максимальное расстояние, находясь на котором, CD-плеер будет мешать суслику спать.

Вторая строка входного файла содержит $n - 1$ целых чисел $x_2, x_3, \ldots, x_n$ ($0 < x_2 < x_3 < \ldots < x_n \le 10^9$) — расстояния от нор с номерами $2, 3, \ldots, n$ до норы с первым номером.

Третья строке содержит m целых чисел $z_1, z_2, \ldots, z_m$ ($0 \le z_1 < z_2 < \ldots < z_m \le 10^9$) — расстояния от последовательных плееров до норы с номером 1. Все плееры расположены к востоку от этой норы.

Далее следуют d строк. i-я из них содержит два целых числа p_i и r_i ($0 \le p_i, r_i \le 10^9$, $p_i \ne r_i$), обозначающих, что в начале i-го дня Дик собирается передвинуть плеер, расположенный на расстоянии p_i метров к востоку от норы с номером 1 в точку, расположенную на расстоянии r_i метров к востоку от этой норы. Гарантируется, что перед каждой такой операцией на расстоянии p_i метров к востоку от норы с номером 1 есть CD-плеер, а на расстоянии r_i — нет.

Output

Ваша программа должна вывести $d+1$ строк. i-я строка (для $i = 1, 2, \ldots, d$) должна содержать одно целое число — количество нор, в которых суслики не смогут заснуть в ночь *перед* i-м передвижением плеера. В $d + 1$-й строке выведите количество нор, в которых суслики не смогут заснуть после последнего заданного передвижения плеера.

Examples

Standard input	Standard output
5 3 4 1 2 5 6 11 2 4 8 2 1 4 10 8 6 1 8	2 3 3 5 3

Problem F. Laundry

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	8 seconds
Memory limit:	256 mebibytes

В прачечную поступили заказы от n футбольных клубов. i-й клуб отправил на стирку $2 \cdot d_i$ гетр и d_i футболок. К сожалению, в связи с отказом техники выстиранное бельё пришлось сушить на верёвке... Чтобы не перепутать заказы, было принято решение различать выстиранные вещи по цвету. Известно, что:

- каждая гетра прикрепляется к верёвке с помощью одной прищепки,
- каждая футболка прикрепляется к верёвке с помощью трёх прищепок,
- гетры от одного клуба должны быть прикреплены прищепками одинакового цвета,
- футболки одного клуба должны быть прикреплены прищепками одинакового цвета,
- Никакие две вещи, принадлежащие различным клубам, не могут быть прикреплены прищепками одного и того же цвета.
- Учитывая вышесказанное, требуется обойтись прищепками минимального количества цветов.

Известно количество прищепок каждого цвета. Выясните, можно ли развесить выстиранные вещи указанным в задаче образом, и, если да — какое минимальное количество цветов прищепок при этом будет использовано?

Input

Первая строка входного файла содержит два целых числа n и k ($2 \le n, k \le 1\,000\,000$) — количество клубов и количество цветов прищепок, которые имеются в прачечной.

Вторая строка содержит n целых чисел $d_1, d_2, \ldots, d_n$ — количество футболок и пар гетр в каждом заказе. ($1 \le d_i \le 1\,000\,000$).

Третья строка содержит k целых чисел $l_1, l_2, \ldots, l_k$. i-е число обозначает количество прищепок i-го цвета в прачечной ($1 \le l_i \le 4\,000\,000$).

Output

Выведите одно число — минимальное количество цветов прищепок, которые будут использованы.

Если развесить выстиранные вещи указанным в задаче способом невозможно, выведите строку "`NIE`".

Examples

Standard input	Standard output
2 4 3 4 20 10 8 10	3
3 8 5 4 3 14 14 14 14 14 14 14 14	NIE

Explanation for the first example: Первому клубу требуется 6 прищепок для гетр и 9 — для футболок. Второму — соответственно 8 и 12. Для заказа от второго клуба можно использовать только прищепки первого цвета, для заказа от первого, например, прищепки второго и четвёртого цвета.

Problem G. Bits Generator (Division 1 Only!)

Input file:	Standard input
Output file:	Standard output
Time limit:	5 seconds
Memory limit:	256 mebibytes

Задан генератор случайных чисел, работающий следующим образом. В момент, когда компьютер включается, случайным образом выбирается целое число между 0 и $m-1$, которое используется как random seed. Для представления этого числа используем переменную z. Для генерации случайного бита вызывается следующая функция, перевычисляющая seed, который потом используется для генерации.

$z := \lfloor (z \cdot a + c)/k \rfloor \bmod m$
if $z < \lfloor m/2 \rfloor$ **then**
  return 0
else
  return 1

Здесь a, c, k — некоторые константы. Вызвав эту функцию n раз, получаем последовательность битов $b_1, b_2, \ldots, b_n$. Сколько различных значений seed могут генерировать эту последовательность?

Input

В первой строке входного файла заданы четыре целых числа a, c, k, m и n ($0 \le a, c < m$, $1 \le k < m$, $2 \le m \le 1\,000\,000$, $1 \le n \le 100\,000$). Вторая строка содержит слово, состоящее из n символов "0" и "1"; i-й символ этой строки задаёт бит b_i.

Output

Выведите одно целое число — количество целых чисел в диапазоне между 0 и $m-1$, которые могли быть начальным значением seed для генератора.

Examples

Standard input	Standard output
3 6 2 9 2 10	4

Explanation of the example: Начальными значениями seed могли быть числа 1, 2, 7 или 8.

Problem H. Afternoon Tea

Input file:	Standard input
Output file:	Standard output
Time limit:	2 seconds
Memory limit:	256 mebibytes

Британским учёным удалось найти регламент, согласно которому проходило описанное Льюисом Кэрроллом Безумное чаепитие.

Сначала чашка наполняется наполовину чаем и наполовину молоком. Далее процесс определяется так называемым *кодом церемонии* — n-буквенным словом, состоящим из букв 'H' и 'M'.

Далее для $i = 1, 2, \ldots, n$, делается следующее действие: если i-я букав кода церемонии равна H, пьющий выпивает чашку до половины и доливает чаю так, чтобы чашка снова была полной. Если i-я буква церемонии равна M, то пьющий выпивает чашку до половины и доливает молока так, чтобы чашка снова была полной.

После прохождения всех n шагов получившаяся смесь выливается — считается, что чай уже остыл и пить его нельзя.

По заданному коду церемонии выясните, чего в процессе чаепития было выпито больше — чаю или молока.

Input

В первой строке входного файла задано целое число n ($1 \le n \le 100\,000$). Вторая строка содержит слово длиной n, состоящее из букв 'H' и 'M' — соответствующий код церемонии.

Output

Если в процессе чаепития было выпито больше чаю, чем молока, выведите "H", если было выпито больше молока, чем чаю, выведите "M", если же чаю и молока было выпито поровну, выведите "HM".

Examples

Standard input	Standard output
5 HMHHM	H

Explanation of the example: Во время чаепития было выпито $1\frac{37}{64}$ чашек чаю и $\frac{59}{64}$ чашек молока.

Problem I. Intelligence Quotient (Division 1 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	10 seconds
Memory limit:	256 mebibytes

В Байтландском университете n студентов учатся на механико-математическом факультете и m — на математико-механическом. При этом все студенты одного факультета знакомы друг с другом, также существуют некоторые пары студентов механико-математического и математико-механического факультета, которые знакомы друг с другом.

Для каждого студента также известен его IQ. Ректор университета открыл новый исследовательский центр при университете и планирует привлечь к работе центра студентов, руководствуясь следующими правилами: любые два студента, приглашённые работать в исследовательском центре, должны быть знакомы друг с другом и суммарный IQ всех студентов должен быть максимален.

Ваша задача — написать программу, строящую список студентов для включения в приказ ректора в соответствии с перечисленными выше требованиями.

Input

Первая строка входного файла содержит три целых числа n, m и k ($1 \le n, m \le 400$, $0 \le k \le n \cdot m$) — соответственно количество студентов на механико-математическом факультете, количество студентов на математико-механическом факультете и количество пар студентов разных факультетов, знакомых друг с другом, соответственно.

В каждой из последующих k строк задана одна пара знакомых: i-я строка содержит два целых числа a_i и b_i ($1 \le a_i \le n$, $1 \le b_i \le m$) — номера личных дел студентов механико-математического и математико-механического факультетов соответственно, знакомых друг с другом. Личные дела студентов каждого факультета нумеруются последовательными натуральными числами, начиная с единицы.

Следующая строка содержит n целых чисел в диапазоре $[1, 10^9]$ — IQ студентов механико-математического факультета, перечисленных по возрастанию номеров их личных дел.

Последняя строка содержит m целых чисел в диапазоре $[1, 10^9]$ — IQ студентов математико-механического факультета, перечисленных по возрастанию номеров их личных дел.

Output

В первой строке выходного файла должно быть одно целое число — максимальный суммарный IQ, которого удалось достичь.

Вторая строка должна содержать одно целое число — количество выбранных студентов механико-математического факультета. В третьей строке в произвольном порядке должны быть перечислены номера личных дел студентов механико-математического факультета, выбранные для работы в центре. Если ни одного такого студента нет, выведите пустую строку.

В четвёртой и пятой строке в аналогичном формате выведите информацию по студентам математико-механического факультета.

В случае, если возможны несколько оптимальных решений, выведите любое.

Examples

Standard input	Standard output
3 2 3 1 1 2 1 2 2 1 3 1 1 2	6 1 2 2 1 2

Problem J. Cave (Division 1 Only!)

Input file:	Standard input
Output file:	Standard output
Time limit:	6 seconds
Memory limit:	256 mebibytes

Спелеологи приступили к исследованию недавно открытого комплекса пещер. Комплекс состоит из n пещер, соединённых узкими «коридорами». При этом между любыми двумя пещерами существует единственный путь, идущий по «коридорам».

Участники экспедиции приняли решение разделиться на группы так, чтобы каждая группа исследовала одинаковое количество пещер и при этом одна пещера была исследована ровно одной группой. Также для того, чтобы не создавать затруднений при движении по коридорам, члены каждой группы должны иметь возможность перемещаться между выбранными ими для исследования пещерами, не проходя через пещеры, исследуемые другими группами.

Какова численность групп, которые могут быть сформированы участники экспедиции?

Input

Первая строка входного файла содержит одно целое число n ($2 \le n \le 3\,000\,000$), обозначающее количество пещер в комплексе. Пещеры занумерованы последовательными целыми числами от 1 до n.

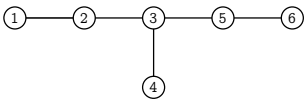
В последующих $n-1$ строках описаны коридоры, соединяющие пещеры. В i-й из этих строк задано целое число a_i ($1 \le a_i \le i$), задающее проход, соединяющий пещеры с номерами $i+1$ и a_i.

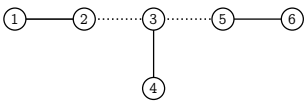
Output

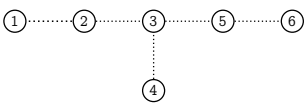
Вывадите в возрастающем порядке все целые неотрицательные числа k, такие, что пещера может быть поделена между k группами указанным выше способом.

Examples

Standard input	Standard output
6 1 2 3 3 5	1 3 6







Problem K. Cross Spider

Input file:	Standard input
Output file:	Standard output
Time limit:	2 seconds
Memory limit:	256 mebibytes

Недавно обнаруженный в Байтландии паук *Araneida baitoida* обладает следующей замечательной особенностью - он может мгновенно построить сколь угодно большую паутину при условии, что она вся лежит в одной плоскости.

Поэтому паук, вместо того, чтобы ждать, пока муха попадётся в построенную паутину, может, зная текущее положение мухи, построить паутину, которая поймает эту муху (соответствующая плоскость пройдёт через соответствующую точку).

Паук заметил n мух, летающих в саду. По координатам этих мух выясните, сможет ли паук, построив паутину прямо сейчас, поймать сразу n мух.

Input

Первая строка входного файла содержит целое число n ($1 \le n \le 100\,000$). Последующие n строк задают положение мух. i-я муха задана тремя целыми числами x_i, y_i, z_i ($-1\,000\,000 \le x_i, y_i, z_i \le 1\,000\,000$) — координатами i-й мухи в трёхмерном Евклидовом пространстве. При этом положение никаких двух мух не совпадает.

Output

Выведите "TAK", если паук сможет поймать всех мух в данный момент, соорудив одну паутину. Иначе выведите "NIE".

Examples

Standard input	Standard output
4 0 0 0 -1 0 -100 100 0 231 5 0 15	TAK
4 0 1 0 -1 0 -100 100 0 231 5 0 15	NIE

Problem L. Choose The Best Computer (Division 2 Only!)

Input file:	Standard input
Output file:	Standard output
Time limit:	2 seconds
Memory limit:	256 mebibytes

Известный тестер компьютеров Том вычислил, что популярность компьютера у покупателей вычисляется по формуле

$$2 \cdot R + 3 \cdot S + D,$$

где

- RAM (в GiB), обозначенна как R;
- частота процессора (в мегагерцах) обозначена как S;
- объём жёсткого диска (в гигабайтах), обозначенный как D.

Ваша задача — по заданному списку компьютеров вывести названия двух наиболее популярных компьютеров из этого списка: сначала название самого популярного, затем — название второго по популярности.

Input

Первая строка входного файла содержит целое число n ($1 \le n \le 10^4$) — количество компьютеров в списке. В каждой из последующих n строк записана спецификация одного компьютера. Спецификация задана в следующем формате: сначала идёт название компьютера — строка, составленная из менее, чем 20 заглавных латинских букв, затем — объём оперативной памяти (целое число R; $1 \le R \le 128$), частота процессора (целое число S; $1 \le S \le 4000$), объём жёсткого диска (целое число D; $1 \le D \le 3000$), разделённые пробелом

Output

Выведите названия двух наиболее популярных компьютеров из списка, по одному имени на строку, отсортированные по убыванию популярности. В случае, если формула даёт одинаковое значение, более популярными считается компьютер, чьё название лексикографически младше (например, "IBM" популярнее "LENOVO").

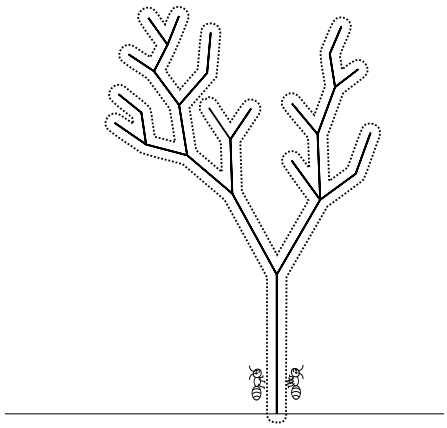
Если дан только один компьютер, то выведите его название в единственной строке выходного файла.

Example

Standard input	Standard output
4 ABC 13 22 1 DEF 10 20 30 GHI 11 2 2 JKL 20 20 20	JKL DEF

Problem M. Ants (Division 2 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	10 seconds
Memory limit:	256 mebibytes



Муравьи, равно как и составители задач по программированию, любят деревья. Нам дано дерево с двумя муравьями, ползущими по нему — Левым муравьём и Правым муравьём так, как это указано на рисунке, расположенном выше (муравьи двигаются вдоль пути, обозначеного пунктирной линией).

Муравьи начинают своё путешествие снизу ствола по двум его противоположным сторонам. Левому муравью требуются 2 секунды для того, чтобы пройти по одному ребру дерева при движении от корня (вверх), и 1 секунда для того, чтобы пройти по одному ребру дерева при движении к корню (вниз).

Правый муравей двигается как вверх, так и вниз в два раза быстрее. Когда два муравья встречаются, они оба разворачиваются и начинают двигаться в противоположном направлении. Если какой-нибудь муравей спустится на землю, он немедленно начинает путешествие с противоположной стороны ствола.

Ваша задача — вычислить момент, когда муравьи встретятся во второй раз. При решении задачи размерами муравьёв пренебречь.

Input

В первой строке входного файла задано целое число t ($1 \le t \le 1000$) — количество тестовых примеров.

Каждый тестовый пример состоит из двух строк. Первая строка содержит чётное число n ($2 \le n \le 10^8$), обозначающее количество рёбер дерева.

Во второй строке задано само дерево. Дерево задаётся строкой, состоящей из $\frac{n}{2}$ символов, обозначающих $2n$ — битовое двоичное число, записанное в шестнадцатеричной системе счисления (используя цифры и строчные буквы от `a` до `f`).

Это число описывает путь Левого муравья по всему дереву в предположении, что Правый муравей не движется. Последовательные биты этого числа (начиная с самого левого) обозначают направление движения по соответствующему ребру: 1, если движение идёт от корня, и 0, если движение идёт к корню. При этом дерево имеет ствол, то есть непосредственно из корня выходит ровно одно ребро.

Output

Для каждого тестового примера в отдельной строке выведите момент второго разворота муравьёв (в секундах), заданный в виде несократимой дроби `p/q` (без пробелов вокруг знака '/'), где p и q — целые положительные числа (если ответ целый, то, очевидно, $q = 1$).

Examples

Standard input	Standard output
1 28 fb1da30d1b7230	282/5

Приведённые в примере данные соответствуют иллюстрации и кодируются в следующую последовательность бит:

```
1111 1011 0001 1101 1010 0011 0000 1101 0001 1011 0111 0010 0011 0000
```

Problem N. Huffman encoding (Division 2 Only!)

Input file:	Standard input
Output file:	Standard output
Time limit:	2 seconds
Memory limit:	256 mebibytes

Концепция кодов Хаффмана базируется на том, что каждому символу ставится в соответствии последовательность нулей и единиц таким образом, что последовательность, кодирующая некий символ, не может быть префиксом для кода никакого другого символа.

Легко заметить, что для создания такого рода последовательности можно просто поместить символы в листья двоичного дерева, обозначить левое ребро как 0, а правое как 1. Тогда путь от корня дерева до данного листа и будет соответствующим кодом.

Например, если построить систему кодов для набора символов {A, B, C, D, E}, то получается такой вариант: A кодируется как 00, B как 01, C как 10, D как 110 и E как 111.

Преимуществом данного способа кодирования является, в частности, то, что любая последовательность таких кодов может быть однозначно декодирована.

Ваша задача — по заданному коду (набору символов и ассоциированных с каждым символом последовательностей нулей и единиц) и заданной закодированной строке восстановить исходную.

Input

Первая строка входного файа содержит одно целое число k ($1 \le k \le 20$) — количество используемых символов. Каждая из последующих k содержит один символ (строчную или прописную латинскую букву), а также его код — непустую последовательность из не более, чем 10 нулей и единиц. Гарантируется, что коды символов удовлетворяют условию задачи.

$k + 2$-я строка содержит непустую последовательность нулей и единиц длиной не более 250 символов, которую надо декодировать. Гарантируется, что последовательность может быть корректно декодирована.

Output

Выведите получившуюся в результате декодирования последовательность символов.

Example

Standard input	Standard output
5 A 00 B 01 C 10 D 110 E 111 00000101111	AABBE

Problem O. Cave (Division 2 Only!)

Input file:	Standard input
Output file:	Standard output
Time limit:	20 seconds
Memory limit:	256 mebibytes

Спелеологи приступили к исследованию недавно открытого комплекса пещер. Комплекс состоит из n пещер, соединённых узкими «коридорами». При этом между любыми двумя пещерами существует единственный путь, идущий по «коридорам».

Участники экспедиции приняли решение разделиться на группы так, чтобы каждая группа исследовала одинаковое количество пещер и при этом одна пещера была исследована ровно одной группой. Также для того, чтобы не создавать затруднений при движении по коридорам, члены каждой группы должны иметь возможность перемещаться между выбранными ими для исследования пещерами, не проходя через пещеры, исследуемые другими группами.

Какова численность групп, которые могут быть сформированы участники экспедиции?

Input

Первая строка входного файла содержит одно целое число n ($2 \le n \le 3\,000\,000$), обозначающее количество пещер в комплексе. Пещеры занумерованы последовательными целыми числами от 1 до n.

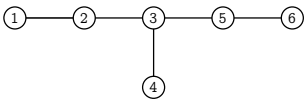
В последующих $n-1$ строках описаны коридоры, соединяющие пещеры. В i-й из этих строк задано целое число a_i ($1 \le a_i \le i$), задающее проход, соединяющий пещеры с номерами $i+1$ и a_i.

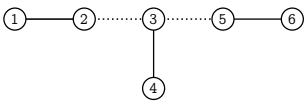
Output

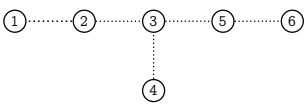
Вывадите в возрастающем порядке все целые неотрицательные числа k, такие, что пещера может быть поделена между k группами указанным выше способом.

Examples

Standard input	Standard output
6 1 2 3 3 5	1 3 6







Problem P. Sequences (Division 2 Only!)

Input file:	Standard input
Output file:	Standard output
Time limit:	7 seconds
Memory limit:	256 Mebibytes

Для заданной последовательности из n целых положительных чисел определим *простую* подпоследовательность как подпоследовательность, состоящую из не менее, чем двух идущих подряд элементов изначальной последовательности, сумма которых равна простому числу.

Например, для последовательности:

35638

существует две простые подпоследовательности длины 2 ($5 + 6 = 11$ и $3 + 8 = 11$), одна простая подпоследовательность длины 3 ($6 + 3 + 8 = 17$), и одна простая подпоследовательность длины 4 ($3 + 5 + 6 + 3 = 17$).

Для заданной последовательности найдите простую подпоследовательность наименьшей длины.

Input

Первая строка входного файла содержит одно целое число t ($1 \le t \le 20$) — количество тестовых примеров.

Каждый тестовый пример состоит из одной строки. Сначала в строке идёт целое число n $0 < n \le 10^4$ — длина последовательности. Далее n неотрицательных целых чисел, не превосходящих 10^4, задают саму последовательность.

Output

Для каждого тестового примера в отдельной строке выведите "`Shortest prime subsequence length is x:`", где x — длина простой подпоследовательности наименьшей длины, а затем (в соответствии с форматом, указанным в примере) — простую подпоследовательность наименьшей длины. Если таких подпоследовательностей несколько, выберите ту из них, которая встречается в исходной последовательности раньше. Если таких подпоследовательностей нет, выведите строку "`No prime subsequences.`".

Example

Standard input
3 5 3 5 6 3 8 5 6 4 5 4 12 21 15 17 16 32 28 22 26 30 34 29 31 20 24 18 33 35 25 27 23 19 21
Standard output
Shortest prime subsequence length is 2: 5 6 Shortest prime subsequence length is 3: 4 5 4 No prime subsequences.

Problem Q. Switch (Division 2 Only!)

Input file:	Standard input
Output file:	Standard output
Time limit:	2 seconds
Memory limit:	256 mebibytes

Вы находитесь на улице, вдоль которой в ряд выстроены K ($4 \le K \le 25$) фонарей, причём некоторые фонари включены, а некоторые — выключены. Из любых взятых подряд четырёх фонарей как минимум один выключен.

Система экономии электроэнергии, управляющая фонарями, устроена так, что если некоторое (не меньшее четырёх) количество фонарей, расположенных подряд, оказываются включёнными, то все эти фонари автоматически выключаются.

Вы можете включить любой выключенный фонарь (но не можете выключать фонари). Какое минимальное количество фонарей вам нужно включить, чтобы в результате ни один фонарь на улице не горел?

Input

В первой строке входного файла задано целое число K — количество фонарей. Каждая из последующих K строк содержит одни число: 0, если соответствующий фонарь выключен, и 1, если соответствующий фонарь включён.

Output

Выведите одно число — минимальное количество включений фонарей, после которого все фонари на улице будут выключены.

Example

Standard input	Standard output
5 1 1 0 1 1	1