Задача A. Вкусный торт

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

К Жене на празднование дня рождения пришло N человек, включая саму Женю, которой подарили огромный вкусный торт в форме выпуклого многоугольника. Она решила разделить его честно между всеми гостями и, конечно, собой. Честно — это значит, что каждому человеку достанется ровно один кусок, причем всем достанутся куски одинаковой площади. При этом каждый кусок должен иметь связную форму простого многоугольника без самопересечений и самокасаний — ведь подавать два раздельных куска одному человеку некрасиво. Торт, к сожалению, скоро испортится, поэтому его необходимо весь сразу съесть.

Женя попросила у вас помощи в нарезке торта. Напишите программу, которая разделит торт на N кусков равной площади.

Формат входного файла

Первая строка входного файла содержит два целых числа N и M ($1 \le N \le 15$, $3 \le M \le 100$) — соответственно, количество людей на дне рождения и количество вершин в выпуклом многоугольнике, форму которого имеет торт.

Следующие M строк содержат по паре целых чисел x_i, y_i, не превышающих по модулю 1000 — координаты i-ой вершины выпуклого многоугольника в порядке обхода против часовой стрелки. Никакие три последовательные вершины не лежат на одной прямой.

Формат выходного файла

В выходной файл выведите выведите N описаний кусков, на которые надо разделить торт в следующем формате.

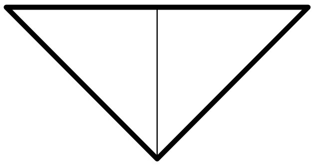
Сначала k_i — число вершин простого многоугольника, форму которого имеет i-ый кусок. При этом необходимо, чтобы k_i не превышало 1000. После этого следуют k_i пар чисел — кординаты вершин многоугольника в порядка любого обхода. Выводите координаты с максимальной точностью.

Примеры

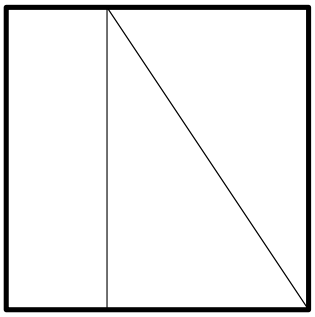
cake.in	cake.out
2 3 0 0 2 2 -2 2	3 0 0 2 2 0 2 3 0 0 -2 2 0 2
3 4 -2 -2 2 -2 2 2 -2 2	4 -2 -2 -0.666667 -2 -0.666667 2 -2 2 3 -0.666667 -2 2 -2 -0.666667 2 3 2 -2 -0.666667 2 2 2

Примечание

Так выглядит разрезание торта в первом примере:



А так во втором примере:



Задача B. Электронные часы

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

Однажды юный часовщик Максим нашел поломанные электронные часы с дисплеем, который показывает количество часов, минут и секунд. Максиму не составило труда починить их, и он решил на часах выставить определенное время.

Однако, найденные часы обладали интересной особенностью — у них была всего одна кнопка. Данная кнопка при срабатывании прибавляла к количеству минут на циферблате часов одну минуту, не изменяя количества часов и секунд. Поясним, что если количество минут при срабатывании кнопки на циферблате было равно «59», то при срабатывании кнопки количество минут на циферблате станет равно «00», а количество часов не изменится. Например, при срабатывании кнопки во время «21:59:23» время на циферблате станет равно «21:00:23».

Несмотря на починку, срабатывала кнопка не мгновенно, а по истечению l секунд после нажатия. Например, если l равно пяти, а Максим нажал кнопку в «21:59:23», то через пять секунд на циферблате часов будет время «21:00:28». А если на часах было «21:59:55», то через пять секунд будет время «22:01:00». В течении данных l секунд нажатие кнопки не влияет на состояние часов.

Максим желает за наименьшее время выставить на циферблате часов определенное время. Напишите программу, которая определит, за какое количество секунд юный часовщик сможет достичь желаемого.

Формат входного файла

Первая строка входного файла содержит время на часах в начальный момент в формате «ЧЧ:ММ:СС». Вторая строка содержит в том же формате желаемое Максимом время на часах. Третья строка содержит целое число l $(1 \leq l \leq 10)$.

Формат выходного файла

Выведите в выходной файл единственное целое число — количество секунд, которое понадобится Максиму для того, чтобы перевести часы.

Примеры

clock.in	clock.out
21:59:23 21:59:23 10	0
21:59:23 21:59:29 1	6
21:59:23 21:00:28 5	5
23:59:23 00:00:28 5	65
21:59:23 21:00:28 6	365

Задача C. Безумие

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

Сереже очень захотелось сделать что-нибудь безумное. Например, сделать так, что бы на стене дома напротив высветилось какое-нибудь хорошее слово.

Он нарисовал на клетчатом листочке размером $a \times b$ карту того, в каких окна должен гореть свет, а в каких нет. Сережа будет считать, что на доме написано слово, если внутри некоторого прямоугольника картина включенных и выключенных окон будет совпадать с этим рисунком, при этом не важно горит свет или нет в остальных окнах.

Для этого ему необходимо в некоторых окнах свет включить, а в некоторых выключить. Однако, чтобы включить или выключить свет в окне, Сереже надо договориться с хозяином квартиры. После этого он сможет произвольным образом включать и выключать свет в любых окнах этой квартиры.

У него есть план дома, и он знает какой квартире принадлежит какое окно. Ему удалось выяснить, что каждой квартире принадлежит несколько соседних окон на одном этаже.

С каким минимальным количеством хозяев надо договориться?

Формат входного файла

В первой строке входного файла два целых числа n и m ($1 \leq n, m \leq 50$).

Далее следуют n строк по m чисел в каждой — какой квартире принадлежит соответствующее окно. В $i+1$-ой строке входного файла на j-ой позиции стоит номер квартиры, которой принадлежит j-ое слева окно на $n - i + 2$-ом этаже дома. Номера квартир — целые положительные числа, не превосходящие 5000.

Потом следуют n строк по m символов в каждой — «#» означает, что свет в соответствующем окне горит, а «.», что нет.

В следующей строке находятся два целых числа a и b ($1 \leq a \leq n, 1 \leq b \leq m$). Затем следуют a строк по b символов в каждой — «#» означает, что соответствующее окно должно светиться в Сережином слове, а «.», что не должно.

Формат выходного файла

В выходной файл выведите единственное число — минимальное число квартир, с хозяевами которых должен договориться Сережа.

Примеры

crazy.in

```
5 26
33 33 33 33 34 34 35 35 35 35 36 36 36 37 37 37 37 38 38 38 39 39 39 40 40 40
25 25 25 25 26 26 27 27 27 27 28 28 28 29 29 29 29 30 30 30 31 31 31 32 32 32
17 17 17 17 18 18 19 19 19 19 20 20 20 21 21 21 21 22 22 22 23 23 23 24 24 24
9  9  9  9 10 10 11 11 11 11 12 12 12 13 13 13 13 14 14 14 15 15 15 16 16 16
1  1  1  1  2  2  3  3  3  3  4  4  4  5  5  5  5  6  6  6  7  7  7  8  8  8
#.#....###..#.#####..#....
.##.####..#.#.##.#..##.###
....####..#....#....#.#..#
..#...#.##...#.#..#####...
...#....#.###.#.#......##.
5 25
#.#.#.#...#.###...#..###.
.###..#..##.#..#..#.#.#.#
..#...#.#.#.###..#.#.###.
.###..##..#.#...#####.#..
#.#.#.#...#.#...#...#.#..
```

crazy.out

```
35
```

Задача D. Код Сайлонов

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

На звездном крейсере «Галактика» перехватили сообщение Сайлонов. Сообщение представляет собой матрицу A размером $n \times m$, заполненную целыми числами. Естественно, сообщение было закодировано. После допроса одной из сайлонских шестерок было выяснено как производится кодирование матрицы. Берется некоторая матрица C размером два на два. Каждый элемент матрицы C — целое число, не превосходящее по модулю единицы, при этом элемент C_{22} не равен нулю. Если производится кодирование матрицы B, то результатом кодирования будет матрица A, имеющая те же размеры, что и B. При этом $A_{ij} = B_{i-1,j-1} \cdot C_{11} + B_{i-1,j} \cdot C_{12} + B_{i,j-1} \cdot C_{21} + B_{ij} \cdot C_{22}$. Если один из индексов выходит за пределы матрицы, то считается, что значение элемента равно нулю.

Адмирал Адама просит Вас расшифровать сообщение Сайлонов, ведь от этого зависит судьба остатков человечества!

Формат входного файла

В первой строке входного файла два целых числа n и m ($1 \le n, m \le 20$). Далее следуют n строк по m чисел в каждой — закодированная матрица A. В $i+1$-ой строке входного файла на j-ой позиции стоит число A_{ij}. Далее следует строка, содержащая четыре числа C_{11}, C_{12}, C_{21} и C_{22} — элементы матрицы C.

Все числа во входном файле целые и не превышают 1000 по абсолютному значению. $|C_{ij}| \le 1$ для любых i и j и $C_{22} \ne 0$.

Формат выходного файла

В выходной файл выведите n строк по m чисел в каждой — результат расшифровки матрицы A при кодировании матрицей C.

Примеры

cylon.in	cylon.out
3 3 1 2 3 4 4 4 7 4 4 -1 0 0 1	1 2 3 4 5 6 7 8 9
1 6 -4 -12 -23 -31 -39 -65 1 1 -1 -1	4 8 15 16 23 42

Задача E. Профессиональный декоратор заборов

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

Вася работает подмастерьем в известной студии. Недавно ему поручили помогать молодому, но подающему большие надежды художественному декоратору заборов и изгородей Витезславу Смолокурову. Миссия эта очень ответственная, и от ее выполнения зависит Васино будущее.

Стиль Смолокурова очень необычен, а его работы пользуются большим спросом. Процесс работы разделен на два этапа. На первом этапе Вася делает заготовку — длинный забор, который состоит из набора цветных вертикальных планок. На втором этапе Витезслав приступает к работе.

Для того, чтобы придать забору более спокойный и гармоничный вид, он несколько раз производит следующую операцию: выбирает некоторый цвет и отрезок, после чего перекрашивает этот отрезок забора в выбранный цвет. По своей творческой натуре, Смолокуров может в корне менять концепцию узора по несколько раз за час, поэтому иногда он перекрашивает одну и ту же планку несколько раз. Кроме того, Витезслав не хочет, чтобы какой-то узор повторялся слишком часто. Для того, чтобы избежать этого, он иногда проверяет, не совпадает ли один отрезок забора с другим.

Несложно догадаться, что и перекрашивание и проверки осуществляет Вася. Работа эта не самая простая, поэтому Вася просит ему помочь хотя бы с проверками на совпадение.

Формат входного файла

Первая строка входного файла содержит одно целое число n — количество планок в заборе ($1 \le n \le 100\,000$). Вторая строка содержит n целых чисел, разделенных пробелами — цвета соответствующих планок.

Третья строка входного файла содержит одно целое число m — количество сравнений и перекрашиваний ($1 \le m \le 100\,000$). Следующие m строк содержат описания заданий, который Вася получает от Витезслава: четыре целых числа q, l, r и k.

В случае перекрашивания $q = 0$. Эта запись означает перекрашивание всех планок с l по r включительно на цвет k ($1 \le l \le r \le n$). В запросе на сравнение $q = 1$. Эта запись означает сравнение кусков забора длины k начиная с позиций l и r соответственно ($1 \le l, r \le n-k+1$, $k > 0$).

Все числа во входном файле положительные и не превышают $100\,000$.

Формат выходного файла

Выведите одну строку: для каждого запроса на сравнение выведите «+» в случае совпадения соответствующих кусков забора и «-» в противном случае.

Примеры

decorator.in	decorator.out
7 1 2 1 3 1 2 1 3 0 4 5 2 1 3 1 2 1 3 1 3	+-
2 1 2 5 1 1 2 1 0 2 2 1 1 1 2 1 0 1 2 3 1 1 1 2	-++

Задача F. Химия

Имя входного файла:	`h2o.in`
Имя выходного файла:	`h2o.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Ваша задача — вычислить массу молекулы вещества, заданного химической формулой. Формула представляет собой перечисление элементов входящих в молекулу вещества. Для сокращения записи формулы используется обозначение En, которое означает, что элемент E надо повторить n раз. Масса молекулы вещества — это сумма весов всех его элементов.

Название элемента	Обозначение	Масса
Водород	H	1
Углерод	C	12
Азот	N	14
Кислород	O	16

Формат входного файла

В единственной строке входного файла записана формула. Ее длина не превышает 20 символов. Формула состоит из букв «CHNO» и цифр от 1 до 9. Цифра всегда идет после буквы.

Формат выходного файла

В выходной файл выведите одно число — массу молекулы.

Примеры

h2o.in	h2o.out
H2O	18
C3H5NO3NO3NO3	227

Задача G. Полурегистронезависимый поиск

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

Недавно в компании, в которой работает Вася, проводили инвентаризацию. В ходе нее было обнаружено множество предметов, которые не нужны в каждодневной работе компании, но могли бы оказаться полезными в некоторых ситуациях. Например: искусственная ель, хоккейная шайба, воздушные шары и многое другое. Все это убрали в кладовку — пыльное и темное помещение. Естественно, никому не хочется каждый раз залезать туда в поисках предмета, которого там может и не оказаться. Для решения этой проблемы шеф отдал приказ составить программу, которая будет по названию предмета определять, есть он в кладовке, или нет.

Разработку и реализацию этой программы поручили Васе. Конечно же, ему предоставили список вещей, находящихся в кладовой. Сперва Вася решил, что задача слишком простая для него, и даже немного обиделся. Однако, вскоре он выяснил неприятную особенность: для упрощения записи, были приняты следующие соглашения: использовать регистр для того, чтобы подчеркнуть габариты предмета. К примеру, было обнаружено множество надувных шаров. В зависимости от размера их записывали как «шар», «Шар», «ШАР» или даже «шаР». Естественно предположить, что человек, который ищет «ШАР» может искать именно большие шарики. С другой стороны, некоторые названия не подвергались таким преобразованиям. Поэтому, если кто-то ищет «Диван», то имеет смысл давать положительный ответ, даже если в списке значится лишь «диван».

После нескольких часов раздумий, Вася решил поступить следующим образом: если какое-то название записано везде одинаково, то программа должна давать положительный ответ, даже если регистры некоторых букв в запросе отличаются. В противном случае нужно проверять, есть ли в списке в точности такое название с точно такими же регистрами всех букв. К несчастью, у Васи случилась беда — сломался компьютер, и он просит вас помочь ему в написании этой программы.

Формат входного файла

Первая строка входного файла содержит одно целое число n ($1 \le n \le 1000$) — количество предметов в списке. Следующие n строк содержат по одному названию предмета из него. Далее следует целое число m — количество запросов ($0 \le m \le 1000$). Каждая из следующим m строк содержит одно слово — запрос. Все названия предметов во входном файле состоят из строчных и прописных латинских букв, длина строк не превосходит 20-ти символов.

Формат выходного файла

Выходной файл должен содержать одну строку: для каждого запроса выведите «+», если ответ на него положительный и «-» в противном случае.

Примеры

halfcase.in	halfcase.out
2 Ok ok 4 Ok ok OK oK	++--
2 no no 2 NO No	++

Задача H. Производство бензина

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

Для оснащения нового цеха по производству бензина компания «Нанонефть» объявила конкурс. На него подали заявки n поставщиков соответствующих производственных линий. Для каждой заявки заданы три числа:

- A_i — стоимость производственной линии;
- B_i — затраты на производство одной тонны бензина на этой линии;
- C_i — цена, по которой произведенную на этой линии тонну бензина готовы купить клиенты.

Точкой окупаемости называется то количество бензина, которое требуется произвести на линии, чтобы его суммарная цена была равна сумме стоимости линии и затрат на его производство.

Вам, как исполняющему обязанности менеджера «Нанонефти» предстоит сделать выбор оптимальной заявки. А именно, необходимо выбрать один вариант оснащения цеха, при котором точке окупаемости соответствует наименьшее количество бензина.

Формат входного файла

В первой строке входного файла содержится число n — количество заявок ($1 \le n \le 10^5$). В следующих n строках заданы по три целых числа A_i, B_i, C_i ($1 \le A_i, B_i, C_i \le 10^9$, $B_i < C_i$).

Формат выходного файла

В выходной файл выведите номер заявки, при выборе которой точке окупаемости соответствует наименьшее количество бензина. При существовании нескольких оптимальных заявок следует вывести номер наименьшей из них.

Примеры

petrol.in	petrol.out
2 1 2 3 2 1 3	1
3 1 2 4 3 1 4 2 2 4	1

Задача I. Доказательство

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

Аня и Валя поступили в институт. Первым предметом у них в расписании была дискретная математика. На первом занятии они изучали N различных эквивалентных определений дерева и доказывали их эквивалентность. За каждое доказательство у доски того факта, что одно определение дерева следует из другого, студент получал конфетку.

Но чтобы все выступления у доски были содержательными, преподаватель поставил условие — нельзя доказывать очевидные факты, которые являются логическими следствиями уже доказанных. То есть, если уже доказано, что из A_1 следует A_2, из A_2 следует A_3, $\ldots$, из A_{k-1} следует A_k, то нельзя доказывать, что из A_1 следует A_k. Студенты в группе, где учатся Аня и Валя, очень дружные, и поэтому они решили распределить доказательства так, чтобы получить как можно больше конфеток.

Выясните, какое максимальное количество конфеток могут получить студенты, и как они должны действовать для этого.

Формат входного файла

Во входном файле содержится единственное число N ($2 \le N \le 100$) — количество различных определений дерева.

Формат выходного файла

В первой строке выходного файла выведите число m — максимальное число конфеток, могут получить студенты.

В каждой из следующих m строк выведите по два числа A и B, означающих, что очередной студент будет доказывать что из определения A следует определение B. Все определения пронумерованы целыми числами от 1 до N.

Примеры

proof.in	proof.out
2	2 1 2 2 1
3	5 1 3 2 3 2 1 1 2 3 1

Задача J. Super Massive Black Hole

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

Джон — потомственный производитель черных дыр. Уже не одно поколение завод его семьи исправно производит черные дыры любой массы.

Сегодня на завод Джона поступил заказ — через D дней он должен отдать заказчику n черных дыр. При этом черные дыры должны иметь массы w_i.

Из соображений безопасности каждый день Джон может производить не более одной черной дыры. К сожалению черные дыры нельзя хранить, так как они очень агрессивны и почти сразу начинают уничтожать все вокруг себя.

Но у Джона есть специальная машина времени для перемещения черных дыр (по неизвестным причинам машина способна перемещать только черные дыры). Однако использование этой машины стоит много денег, а именно, чтобы переместить черную дыру массой w на один день вперед или назад, Джону приходится заплатить w условных единиц. Естественно Джон хочет тратить как можно меньше условных единиц на перемещение черных дыр. Помогите Джону.

Формат входного файла

В первой строке входного файла два целых числа n и D ($1 \le n \le 100$, $0 \le D \le 1000$). Во второй строке n целых чисел w_i ($1 \le w_i \le 1000$) — массы черных дыр.

Формат выходного файла

В выходной файл одно число — минимальное число условных единиц, необходимое на выполнение заказа. Напоминаем, что все черные дыры необходимо отдать заказчику в день D, а начать их производить Джон может только с сегодняшнего дня. Сегодняшний день имеет номер ноль.

Примеры

smbh.in	smbh.out
3 1 1 2 3	3
3 0 100 200 500	400

Примечание

В первом примере одно из оптимальных решений выглядит следующим образом: в нулевой день производится первая черная дыра, в первый — третья и во второй — вторая. Таким образом суммарные затраты составляют $1 \cdot 1 + 3 \cdot 0 + 2 \cdot 1 = 3$.