Задача A. Законопроект

Имя входного файла:	`bill.in`
Имя выходного файла:	`bill.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

В связи с изменением политической обстановки в мире и во избежание наступления экономического кризиса в стране, президент Флатландии на очередном заседании кабинета министров поставил перед правительством первоочередную задачу о подготовке пакета реформ. Сразу же после заседания был созван специальный комитет, который подготовил k различных законопроектов, каждый из которых описывал один из возможных путей проведения реформ. Перед парламентом Флатландии (который состоит из n депутатов) встала нелегкая задача выбора одного из этих законопроектов.

Согласно конституции Флатландии, все обсуждения законопроектов в парламенте проходят в одном чтении. После того, как депутаты ознакомятся с проектом закона, проводится голосование. Каждый из депутатов может проголосовать либо за принятие закона, либо против. При этом, закон принимается, если за него проголосовало как минимум l депутатов.

Известно, что законопроекты, подготовленные специальным комитетом, будут обсуждаться на заседаниях парламента последовательно, начиная с проекта номер 1 и заканчивая проектом номер k. При этом, если в результате обсуждения законопроекта номер i этот законопроект будет принят, то оставшиеся законопроекты (с номерами от $i + 1$ до k) не будут обсуждаться вообще. Если же законопроект с номером i будет отклонен, то парламент перейдет к обсуждению следующего законопроекта (с номером $i + 1$). Если в результате последовательного голосования парламент отвергнет все проекты, то процесс проведения реформ отложится на неопределенное время (в этом случае страну, скорее всего, ожидает глубокий кризис).

Известно также, что каждый депутат Флатландии кратко ознакомился с каждым законопроектом еще до первого заседания парламента и составил для себя список всех проектов в порядке своих предпочтений. Обозначим номер j-го законопроекта в списке предпочтений i-го депутата, как $a_{i,j}$. Тогда это будет означать, что i-й депутат во время голосований будет пытаться добиться принятия проекта с номером $a_{i,1}$, а если это не возможно, то проекта с номером $a_{i,2}$; если и этого никак не добиться, то i-й депутат будет пытаться протолкнуть закон номер $a_{i,3}$ и так далее. При этом никакой депутат не хочет допустить кризиса в стране, и, поэтому, в первую очередь будет пытаться предотвратить тот случай, когда все законопроекты будут отвергнуты парламентом.

Все депутаты флатландского парламента знают не только свои предпочтения, но и предпочтения всех своих коллег. К тому же они достаточно умны и будут действовать оптимально (то есть будут пытаться добиться принятия закона, как можно более предпочтительного с их точки зрения).

Президент Флатландии попросил Вас, как своего главного советника, предсказать результаты работы парламента (а именно то, какой из вариантов закона о реформах будет в итоге принят).

Формат входного файла

В первой строке входного файла записаны три целых числа: n ($1 \le n \le 1000$), l ($1 \le l \le n$) и k ($1 \le k \le 1000$). Далее следует n строк, каждая из которых содержит k чисел (j-е число в i-й строке равно $a_{i,j}$).

Формат выходного файла

В единственной строке выходного файла выведите номер законопроекта, который будет принят парламентом, или слово “`Crisis`” (без кавычек), если все они будут отклонены.

Примеры

bill.in	bill.out
3 2 4 3 4 1 2 2 3 4 1 4 1 2 3	1

Задача B. Рисование

Имя входного файла:	`drawing.in`
Имя выходного файла:	`drawing.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

При обучении школьников младших классов программированию часто используется язык «Лого», позволяющий рисовать на экране картинки хвостом виртуальной черепашки.

Рассмотрим упрощенную версию этого языка, в которой разрешается подавать черепашке следующие команды: переместиться вперед на некоторое количество сантиметров, рисуя за собой линию, повернуть налево на 90 градусов и повернуть направо на 90 градусов.

Рассмотрим замкнутую фигуру, нарисованную таким образом. Требуется проверить, верно ли, что любой вертикальный и любой горизонтальный отрезок, соединяющий две точки фигуры, полностью содержится внутри этой фигуры.

Формат входного файла

Первая строка входного файла содержит число n — количество команд черепашки, использованных для рисования фигуры ($7 \le n \le 40\,000$). Следующие n строк содержат команды.

Команды задаются следующим образом:

- “`f` x” — переместиться вперед на x сантиметров (x — натуральное число);
- “`l`” — повернуть налево;
- “`r`” — повернуть направо.

Известно, что последовательность команд начинается с команды перемещения, не выполняется подряд две команды поворота или две команды перемещения, последняя команда также является командой перемещения.

Гарантируется, что ломаную можно поместить в некоторый прямоугольник, длины сторон которого не превосходят 10^9.

Формат выходного файла

В выходной файл выведите «`TRUE`», если нарисованная фигура удовлетворяет описанному выше свойству и «`FALSE`», если не удовлетворяет.

Примеры

drawing.in	drawing.out
11 f 1 r f 2 r f 2 r f 1 r f 1 l f 1	TRUE

Задача C. Экспедиция на Морс

Имя входного файла:	`mors.in`
Имя выходного файла:	`mors.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Для исследования плоского космоса была организована сверхсекретная научная экспедиция. Специально для нее был сконструирован космический корабль *Z-239*, а в качестве первой планеты для исследования была выбрана *Зумла*.

На *Z-239* установлено n двигателей на антиматерии, каждый из которых характеризуется вектором силы тяги, которую он создает. Для перемещения корабля необходимо включить некоторые двигатели, при этом общая сила тяги будет равна векторной сумме тяг включенных двигателей.

При старте корабля его можно ориентировать произвольным образом, после чего включить некоторые (возможно, все) двигатели.

Очередной целью экспедиции была выбрана планета *Морс*. Разумеется, руководитель экспедиции хочет добраться туда как можно быстрее. Чем больше модуль суммарной силы тяги, тем быстрее полетит корабль. Какие двигатели нужно включить, чтобы добраться на Морс как можно быстрее?

Формат входного файла

Первая строка содержит натуральное число n — количество двигателей ($1 \le n \le 30\,000$). В $i+1$-ой строке записана пара чисел — модуль силы тяги i-ого двигателя, и угол в градусах между вектором его силы тяги и направлением на Полярную звезду перед поворотом корабля (вещественное число от 0 до 360 градусов). Модуль силы тяги каждого двигателя не превосходит $10\,000$.

Формат выходного файла

В первой строке выходного файла выведите количество l двигателей, которые надо включить. Во строке должны быть записаны l различных чисел, разделенных пробелами — номера включаемых двигателей в любом порядке.

Примеры

mors.in	mors.out
3 5.00 0.00 3.00 90.00 4.00 270.00	2 1 3

Задача D. Число

Имя входного файла:	`number.in`
Имя выходного файла:	`number.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Это предложение содержит двенадцать слов, двадцать шесть слогов и семьдесят три буквы.

Вы только что начали читать предложение, чтение которого вы уже заканчиваете.

Скажем, что число в b-ичное системе счисления *описывает себя*, если оно — b-значное, его первая цифра равна числу единиц в нем, вторая цифра равна числу двоек, и т.д., $b-1$-ая цифра равна числу цифр $b-1$, а последняя цифра равна числу нулей. Примером такого числа для $b=5$ является 12002.

Задано число b. Требуется найти число в системе счисления по основанию b, которое описывает себя.

Формат входного файла

Входной файл содержит единственное целое число b ($2 \le b \le 1000$).

Формат выходного файла

Выведите в выходной файл любое описывающее себя число в системе счисления с основанием b, или -1, если такого числа не существует. Каждую цифру числа выводите на отдельной строке в десятичной записи.

Примеры

number.in	number.out
5	1 2 0 0 2
2	-1

Задача E. Физика

Имя входного файла:	`physics.in`
Имя выходного файла:	`physics.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

В трехмерном пространстве с прямоугольной декартовой системой координат находятся n абсолютно упругих шаров. Для каждого из них известны: масса m_i, радиус r_i, координаты x_i, y_i, z_i и вектор скорости $\mathbf{v_i} = (v_{ix}, v_{iy}, v_{iz})$ в начальный момент времени. Необходимо рассчитать их координаты и скорости по прошествии T секунд от начального момента времени. Шары взаимодействуют только при соударениях, других взаимодействий между ними нет.

Учтите, что шары могут сталкиваться, и их столкновение описывается законами сохранения энергии и импульса, то есть сохраняется величина $\sum\limits_{i=1}^{n} m_i \cdot (v_{ix}^2 + v_{iy}^2 + v_{iz}^2)$, а также вектор $\sum\limits_{i=1}^{n} m_i \cdot \mathbf{v_i}$. Заметим, что при столкновении шаров их скорости могут измениться только на вектор, параллельный прямой, соединяющей их центры в момент столкновения.

Формат входного файла

Первая строка входного файла содержит натуральное число n $(1 \le n \le 50)$ — количество шаров. Следующие n строк входного файла содержат описание начального состояния шаров, $(i+1)$-ая строка содержит 8 разделенных пробелами вещественных чисел, не более чем с тремя знаками после запятой: m_i, r_i, x_i, y_i, z_i, v_{ix}, v_{iy}, v_{iz}.

Последняя строка входного файла содержит целое число T $(0 \le T \le 100)$ — время, состояние системы по прошествии которого надо рассчитать. Все проекции скоростей заданы в метрах в секунду, все радиусы и координаты центров — в метрах, все массы — в килограммах, а время T задано в секундах. Гарантируется, что входные данные таковы, что в каждом столкновении участвуют ровно 2 шара. Все числа во входном файле не превосходят 100 по абсолютной величине. Начальное положение шаров таково, что они не касаются друг друга и не пересекаются. Массы и радиусы всех шаров строго положительны.

Формат выходного файла

В выходной файл выведите n строк. На i-ой строке выведите 6 вещественных чисел с точностью не меньше чем 3 знака после десятичной точки: x-координату центра i-ого шара, y-координату центра i-ого шара, z-координату центра i-ого шара, проекцию его скорости на ось Ox, проекцию его скорости на ось Oy, проекцию его скорости на ось Oz по прошествии T секунд. Координаты выводите в метрах, проекции скоростей — в метрах в секунду.

Примеры

physics.in
```
3
1.00 1.00 0.00 0.00 0.00 1.00 1.00 1.00
2.00 2.00 10.00 10.00 10.00 -1.00 -1.00 -1.00
3.00 3.00 20.00 20.00 20.00 -0.5 -0.5 -0.5
100
```

physics.out
```
-155.6427 -155.6427 -155.6427 -1.6667 -1.6667 -1.6667
-47.0284 -47.0284 -47.0284 -0.6667 -0.6667 -0.6667
26.5665 26.5665 26.5665 0.1667 0.1667 0.1667
```

physics.in
2 1.00 1.00 0 0 0 0 0 0 1.00 1.00 10 0 0 -1 0 0 9
physics.out
-1.0000 0.0000 0.0000 -1.0000 0.0000 0.0000 2.0000 0.0000 0.0000 0.0000 0.0000 0.0000

Задача F. Транспортная реформа

Имя входного файла:	`reform.in`
Имя выходного файла:	`reform.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Далекое королевство состоит из n городов, соединенных m дорогами, при этом из каждого города можно доехать в каждый по дорогам (по каждой из дорог можно ездить в обоих направлениях). Некоторые дороги находятся в хорошем состоянии, другие же требуют ремонта.

Содержание каждой из дорог обходится казне в 239239 дублонов в год. В связи с новой транспортной реформой, направленной на уменьшение ненужных расходов, было решено оставить на государственном содержании как можно меньше дорог, но по-прежнему должна остаться возможность из каждого города добраться в любой другой по государственным дорогам. Заметим, что при этом на государственном содержании останется ровно $n - 1$ дорога.

При этом, возможно, в число оставленных дорог войдут и требующие ремонта. Сэкономленные в связи с этой реформой деньги планируется направить на ремонт дорог, оставленных на государственном содержании.

В королевстве есть две конкурирующие фирмы, занимающиеся починкой дорог. Поскольку король не желает быть обвиненным в поддержке одной из этих фирм, он хочет иметь возможность распределить дороги, оставленные на государственном содержании и требующие ремонта, поровну между ними. В связи с этим требуется оставить на государственном содержании четное число дорог, требующих ремонта.

Требуется составить соответствующий желанию короля список дорог, которые следует оставить на государственном содержании.

Формат входного файла

Первая строка содержит натуральные числа n ($2 \le n \le 30\,000$) и m ($1 \le m \le 100\,000$) — число городов и дорог в королевстве соответственно. Далее идут m строк — описание дорог. Дорога описывается тремя числами. Первые два из них — номера городов, которые она соединяет, третье равно 1, если дорога не требует ремонта, и 2, если требует. Города пронумерованы целыми числами от 1 до n. Ни одна дорога не соединяет город с самим собой, между любыми двумя городами не более одной дороги.

Формат выходного файла

В выходной файл выведите $n - 1$ строку — номера государственных дорог (дороги нумеруются с 1 в том порядке, в котором они идут во входном файле).

Если оставить на государственном содержании дороги указанным образом невозможно — выведите -1.

Примеры

reform.in	reform.out
4 4 1 2 1 2 3 2 3 4 1 4 1 2	2 3 4

Задача G. Небоскреб

Имя входного файла:	`sky.in`
Имя выходного файла:	`sky.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Спасатели разрабатывают новую систему действий при возникновении аварийных ситуациях на небоскребах. В этих случаях важно определять, какие части здания стоят устойчиво, а какие — нет.

На данный момент используется следующая двумерная модель. Строение описывается как прямоугольник, составленный из одинаковых блоков квадратной формы. Предполагается, что при аварии некоторые из блоков полностью разрушаются, а остальные остаются неповрежденными. Будем называть сегментом множество блоков, таких что из любого можно дойти до любого другого, если разрешается переходить из блока в блок, имеющий с ним общую сторону.

Считается, что сегмент стоит устойчиво, если один из его блоков соприкасается с нижней стороной прямоугольника. Сегменты из блоков, которые стоят неустойчиво, проваливаются вниз до тех пор, пока какой-либо из блоков сегмента не будет соприкасаться с нижней стороной прямоугольника или с блоком устойчивого сегмента. После этого сегмент так же считаются стоящим устойчиво.

По данным о том, какие блоки сохранились, требуется определить положение верхнего блока в каждом вертикальном столбце.

Формат входного файла

Первая строка входного файла содержит одно целое число m ($1 \le m \le 10\,000$). За ней следуют m строк, описывающих вертикальные столбцы. Описание производится в таком порядке: сначала записано число a_i оставшихся целыми фрагментов этого вертикального столбца, за которым следуют $2a_i$ чисел $l_i^{(1)}, r_i^{(1)}, l_i^{(2)}, r_i^{(2)}, \ldots, l_i^{(a_i)}, r_i^{(a_i)}$, задающие нижние и верхние границы уцелевших фрагментов. При этом $1 \le l_i^{(1)} \le r_i^{(1)}$, $l_i^{(2)} \le r_i^{(2)}$, $\ldots$, $l_i^{(a_i)} \le r_i^{(a_i)} \le 10^6$, для всех допустимых i и j выполнено $r_i^{(j)} < l_i^{(j+1)} - 1$.

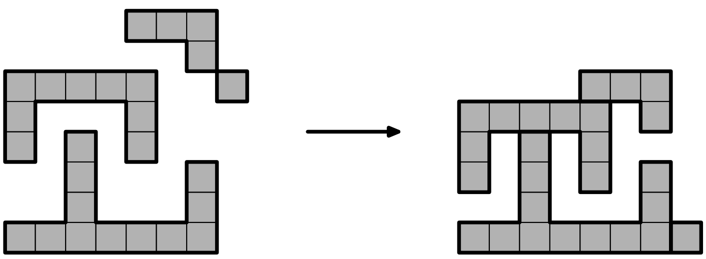
Сумма всех a_i не превосходит $100\,000$.

Формат выходного файла

В выходной файл выведите m чисел — высоты, на которых находится самый верхний блок в соответствующем вертикальном столбце.

Примеры

sky.in	sky.out
8 2 1 1 4 6 2 1 1 6 6 2 1 4 6 6 2 1 1 6 6 3 1 1 4 6 8 8 2 1 1 8 8 2 1 3 7 8 1 6 6	5 5 5 5 6 6 6 1



Задача H. Башни

Имя входного файла:	towers.in
Имя выходного файла:	towers.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Для того, чтобы защититься от некоторых соседей, король решил построить стену, имеющую форму отрезка. С некоторыми соседями король находится в хороших отношениях, а некоторым готовится объявить войну. Король решил не загораживаться от друзей очень высокой стеной. Однако, стена, отделяющая его от врагов, должна быть достаточно высокой. Было решено, что для наблюдения за прилежащей территорией нужно построить башни. При этом, на участках между башнями высота стен должна изменяться равномерно.

После того, как стена и башни были построены, король заметил, что башни могут быть использованы для наблюдения за состоянием других башен. Однако, некоторые башни оказались очень высокими и загородили другие.

Для каждой башни король попросил вас выяснить, сколько других башен из нее видно.

Формат входного файла

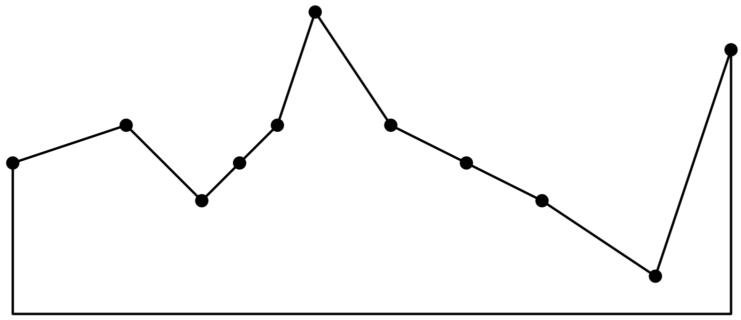
В первой строке входного файла находится n ($2 \le n \le 2000$) — количество башен стены. В следующих n строках находятся натуральные числа x_i и h_i ($0 \le x_i \le 100000$, $1 \le h_i \le 10000$) — координата и высота i-ой башни. Все x_i различны.

Формат выходного файла

Выходной файл должен содержать n строк. В i-ой строке выведите количество башен, которые видно из башни номер i.

Примеры

towers.in	towers.out
11 0 4 3 5 5 3 6 4 7 5 8 8 10 5 12 4 14 3 17 1 19 7	2 5 4 4 4 10 4 4 5 3 5



Задача I. Треугольник

Имя входного файла:	`triangle.in`
Имя выходного файла:	`triangle.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

После окончания многолетней войны короли-победители решили разделить между собой захваченную территорию. Для того, чтобы избежать лишних споров, короли решили делить территорию следующим образом.

Каждому королю был предоставлен набор, состоящий из n отрезков $l_1, \ldots, l_n$. Королям разрешается присоединять к их королевствам территорию, имеющую форму треугольника, составленного из имеющихся отрезков.

Помогите королю вашего королевства максимизировать площадь треугольника.

Формат входного файла

На первой строке задано число n ($3 \le n \le 100000$) — количество отрезков в наборе. В следующих n строках заданы длины отрезков в следующем формате.

Сначала указывается целое положительное число l длина отрезка, а затем единица измерения (`m`, `km`, `mile`, `uin`, `kairi`, `zhang`, `sen`).

Напомним, что:

$$1mile = 1609m$$
$$1km = 1000m$$
$$1uin = 33m$$
$$1kairi = 1852m$$
$$1zhang = 3m$$
$$1sen = 38m$$

Длины всех отрезков не превосходят десяти километров. Гарантируется, что из заданных отрезков можно выбрать три отрезка так, что из них можно составить треугольник.

Формат выходного файла

Выведите в выходной файл максимальную площадь (в таланг вахах) с точностью 10^{-6} и номера трех отрезков в любом порядке (если ответов несколько, выведите любой). Помните, что один таланг вах равен четырем квадратным метрам.

Примеры

triangle.in	triangle.out
7 1mile 1km 100m 20uin 1kairi 300zhang 40sen	291590.0233624191 7 1 5

Задача J. Слова

Имя входного файла:	words.in
Имя выходного файла:	words.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Для шифрования слов с ними можно производить множество различных операций. Например, интересна такая операция: первые несколько букв заданного слова приписываются к его концу в обратном порядке, после чего удаляются из начала слова. При этом слово $a_1a_2 \ldots a_k a_{k+1} \ldots a_n$ переходит в слово $a_{k+1} \ldots a_n a_k a_{k-1} \ldots a_1$ (число k выбирается в диапазоне от 0 до n).

Для двух заданных слов требуется определить, можно ли применением описанной операции преобразовать первое слово во второе.

Формат входного файла

Входной файл состоит из двух строк — на первой из них записано исходное слово, а во второй — предполагаемый результат.

Формат выходного файла

В первой строке выходного файла выведите "Yes", если преобразование возможно и "No", если нет. В случае положительного ответа во второй строке выведите k — длину перемещаемой части исходного слова k (из всех таких k выберите минимальный).

Примеры

words.in	words.out
wpwdwpw wdwpwpw	Yes 2
dWOddWd dOdWdWd	No