Задача A. Конечные автоматы

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

Однажды известный профессор обнаружил описания k конечных автоматов. По его мнению, *нетривиальность* конечного автомата, имеющего n состояний и m переходов, можно описать целым числом $d = 19m + (n+239) \cdot (n+366)/2$. Чем больше d, тем больший интерес для науки представляет изучение его свойств.

Помогите профессору вычислить нетривиальность имеющихся у него автоматов.

Формат входного файла

Первая строка входного файла содержит целое число k ($1 \le k \le 10000$) — количество конечных автоматов. Слудующие k строк содержат по два целых числа n_i ($1 \le n_i \le 1000$) и m_i ($0 \le m_i \le 26n_i^2$) — число состояний и переходов i-го автомата.

Формат выходного файла

Выходной файл должен состоять из k строк. На i-ой строке выходного файла выведите одно число — нетривиальность i-го автомата.

Примеры

automata.in	**automata.out**
4 2 0 13 20 5 23 18 6	44344 48134 45699 49458
2 15 20 1000 26000	48767 1340237

Задача B. Постулат Бертрана

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

Постулат Бертрана (теорема Бертрана-Чебышёва, теорема Чебышёва) гласит, что для любого $n \geq 2$ найдётся простое число p в интервале $n < p < 2n$. Такая гипотеза была выдвинута в 1845 году французским математиком Джозефом Бертраном (проверившим её до $n = 3000000$) и доказана в 1850 Пафнутием Чебышёвым. Раманужан в 1920 году нашёл более простое доказательство, а Эрдёш в 1932 - ещё более простое.

Ваша задача состоит в том, чтобы решить несколько более общую задачу — а именно по числу n найти количество простых чисел p из интервала $n < p < 2n$.

Напомним, что число называется *простым*, если оно делится только само на себя и на единицу.

Формат входного файла

Входной файл содержит целое число n ($2 \leq n \leq 3000$).

Формат выходного файла

В выходной файл выведите одно число — ответ на задачу.

Примеры

bertrand.in	**bertrand.out**
2	1
239	39
3000	353

Задача C. Эксперимент

Имя входного файла:	`exp.in`
Имя выходного файла:	`exp.out`
Ограничение по времени:	5 секунд
Ограничение по памяти:	64 мегабайта

Ученые в одной сверхсекретной лаборатории проводят эксперименты над сверхсекретными газами *Аккермана*. Перед началом эксперимента у них есть n пронумерованных колб объема, в которых находятся разные газы. Ученые соединяют колбы трубками, и содержащиеся в них газы приходят в контакт. Известно, что если в контакт приходят какие-то два газа, то тот из них, который изначально находился в колбе с большим номером, неимоверно быстро расширяется, уничтожает другой газ и занимает все освобождающееся пространство.

По заданной последовательность действий, которую проводят ученые, вы должны предсказать результат эксперимента.

Формат входного файла

Первая строка входного файла содержит количество колб n и количество действий m ($0 < n \le 10$, $0 \le m \le 15$). Во второй строке записаны n целых чисел v_i ($0 < v_i \le 10$) — объемы колб. Каждая из последующих m строк содержит два различных целых числа от 1 до n — номера колб, соединяемых на соответствующем шаге.

Формат выходного файла

Для каждого из сохранившихся газов выведите в выходной файл его номер и занимаемый им объем в отдельной строке. Информацию выводите в порядке увеличения номеров газов.

Примеры

exp.in	exp.out
5 3 10 10 10 10 10 1 2 2 3 4 5	3 30 5 20

Задача D. Пересылка файлов

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

В постиндустриальную эпоху основной ценностью является информация. Поэтому особо важен контроль над каналами передачи информации. В одной стране все каналы связи контролируются государством.

Перед ИТ-отделом одной достаточно крупной фирмы, занимающейся консалтингом в области инновационных технологий, была поставлена задача распространить некий файл по филиалам этой фирмы, находящимся в различных городах страны.

Каналы передачи информации в этой стране, как уже говорилось, контролируются государством, поэтому за передачу по ним информации приходится платить деньги. Ситуация также осложняется тем, что каналы однонаправленные, то есть информацию по ним можно передавать только в одном направлении.

Пусть, для удобства, города пронумерованы натуральными числами от Главный офис находится в городе номер 1, таким образом, необходимо найти такой набор каналов связи, по которым можно доставить файл от города номер 1 до любого другого, а среди всех таких наборов выбрать имеющий наименьшую суммарную стоимость.

Задан список каналов связи, которыми может воспользоваться фирма. Напишите программу находящую требуемый набор каналов связи.

Формат входного файла

Первая строка входного файла содержит числа n и m — количество городов и количество каналов связи соответственно ($1 \leq n, m \leq 22$). Последующие m содержат описания каналов связи. Каждое описание содержит три целых числа: u, v и c — соответственно номера городов, соединенных каналом и стоимость пересылки файла по этому каналу ($1 \leq u, v \leq n$, $0 \leq \ \leq 1000$). Ни один из каналов не соединяет город с самим собой, но между двумя городов может быть больше одного канала.

Формат выходного файла

В первой строке выходного файла выведите стоимость пересылки файла и размера набора каналов, обеспечивающего такую стоимость. Во второй строке выведите номера каналов, составляющих такой набор. Каналы нумеруются от 1 до m в том порядке, в котором они перечислены во входном файле.

Примеры

file.in	file.out
2 2 1 2 3 1 2 4	3 1 1
3 3 1 2 5 1 3 10 3 2 4	14 2 2 3

Задача E. Сжимающий оператор

Имя входного файла:	`operator.in`
Имя выходного файла:	`operator.out`
Ограничение по времени:	3 секунды
Ограничение по памяти:	64 мегабайта

Оператором A, действующим из множества X в множество Y (или просто оператором из X в Y) называется правило, согласно которому каждому элементу x множества X сопоставляется элемент $y = Ax$ из множества Y. Пусть X и Y — множества точек на плоскости. Оператор A из X в Y называется *сжимающим с коэффициентом* q, где q — вещественное число из полуинтервала $[0, 1)$, если для любого $x \in X$ выполнено $A||x|| \le q||x||$ (здесь $||x||$ — *норма* точки x — расстояние от x до начала координат). Проще говоря, оператор называется сжимающим с коэффициентом q если он сопоставляет каждой точке точку, которая не менее, чем в q раз ближе к началу координат.

Для заданного оператора A требуется проверить, является ли он сжимающим с коэффициентом q.

Формат входного файла

Первая строка входного файла содержит количество точек n $(1 \le n \le 100)$ и число q $(0 \le q < 1)$, заданное не более чем с 3 знаками после десятичной точки. Следующие n строк содержат по 4 целых числа, по модулю не превосходящих 1000, разделенные пробелами — координаты точки множества X и сопоставленной ей точки из множества Y.

Формат выходного файла

В выходной файл выведите одно слово: «`Yes`» если оператор является сжимающим с коэффициентом q и «`No`» в противном случае.

Примеры

operator.in	**operator.out**

`2 0.5`
`0 10 5 0`
`10 0 0 1`
	`Yes`

`2 0.1`
`0 10 5 0`
`10 0 0 1`
	`No`

`2 0.9`
`0 0 0 0`
`10 0 0 1`
	`Yes`

Задача F. Проценты

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

Списки ингредиентов на упаковках иногда сопровождаются их процентным содержанием, чаще всего округленным до целого числа процентов. Чтобы такой список выглядел правдоподобным, в сумме указанные числа давать 100%.

Однако и здесь есть определенные тонкости. Нетрудно убедиться, что, если округлять все дробные числа процентов по математическим правилам, то результирующая сумма может отличаться от нужной. Поэтому никто не сможет усомниться в вашей честности, если вы произведете округление так, как сочтете нужным. Осталось только найти лучший вариант.

Вам заданы количества всех ингредиентов, входящих в состав продукта. Для каждого ингредиента известно, положительно или отрицательно влияет на продажи его присутствие в составе. По этим данным необходимо рассчитать процентные доли каждого из ингредиентов от их суммарного количества и округлить их в нужную сторону. При этом не следует допускать, чтобы одновременно доля какого-либо вредного компонента была округлена вверх, а доля какого-либо хорошего — вниз.

Формат входного файла

В первой строке входного файла задано количество ингридиентов n ($1 \le n \le 30$). Следующие n строк описывают сами ингредиенты: знак «+» для положительно влияющих на продажи, и «−» для для отрицательно влияющих, а затем, через пробел, количество соответствующего ингредиента — целое число от 1 до 1000.

Формат выходного файла

В выходной файл выведите n целых чисел, в сумме дающих 100, по одному на строке — процентные содержания ингредиентов.

Примеры

percent.in	**percent.out**
2 - 10 + 5	66 34
3 - 10 - 10 - 10	33 34 33

Задача G. Треугольная комната

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

Во многих книгах по занимательной математике приводится такая задача. *Расставить по периметру треугольной комнаты 3 стула так, чтобы у каждой стены стояло по 2. Ее решение — поставить по стулу в каждый из углов комнаты.*

Эта задача легко обобщается. Пусть комната представляет собой треугольник ABC. Даны: общее количество стульев n, количество стульев n_{AB}, которое должно стоять у стены AB, количество стульев n_{BC}, которое должно стоять у стены BC, количество стульев n_{AC}, которое должно стоять у стены AC. Необходимо найти соответствующую расстановку стульев или установить, что ее не существует. При этом стулья можно ставить только в углы комнаты и вдоль стен, в центр комнаты стулья ставить нельзя. В любой из углов можно поставить произвольное количество стульев.

Формат входного файла

Входной файл содержит целые числа $n, n_{AB}, n_{BC}, n_{AC}$ $(0 \le n, n_{AB}, n_{BC}, n_{AC} \le 1000)$.

Формат выходного файла

В первой строке выходного файла выведите `NO`, если стулья указанным способом расставить невозможно. В противном случае выведите `YES` в первой строке выходного файла, а во второй выведите 6 целых неотрицатеьных чисел: $k_A, k_{AB}, k_B, k_{BC}, k_C, k_{AC}$ — соответственно количество стульев, которые необходимо поставить в угол A, вдоль стены AB, в угол B, вдоль стены BC, в угол C и вдоль стены AC.

Примеры

room.in	room.out
`3 2 2 2`	`YES` `1 0 1 0 1 0`
`3 3 2 2`	`NO`

Задача H. Последовательность

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

Вася написал на доске n целых чисел a_i и ушел. Пришел Петя и, увидев Васину последовательность, решил её немного изменить. Для этого он решил, что может стирать с доски лишь те числа, оба соседа которых строго больше. Более формально, Петя мог стереть i-ое число, если $a_{i-1} > a_i$ и $a_{i+1} > a_i$. Когда на доске не осталось чисел, которые мог стереть Петя, он ушел.

Пришел Вася и очень удивился увиденному. Напишите программу, которая выводит последовательность, которую увидел Вася.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \le n \le 10000$) — количество чисел. Следующая строка содержит n разделенных пробелом целых чисел a_i ($-1000000 \le a_i \le 1000000$) — записанные на доске числа.

Формат выходного файла

На первая строке выходного файла выведите целое число k — количество осташихся на доске чисел. На следующей строке выведите k чисел b_i — оставшиеся на доске числа.

Примеры

sequence.in	**sequence.out**
12 1 2 3 2 4 1 3 4 2 3 2 1	8 1 2 3 4 4 3 2 1