Задача A. Цикл

Входной файл:	`cicle.in`
Выходной файл:	`cicle.out`
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

Дан граф. Определить, есть ли в нем цикл отрицательного веса, и если да, то вывести его.

Описание входного файла

Во входном файле в первой строке число N ($1 \le N \le 100$) — количество вершин графа. В следующих N строках находится по N чисел — матрица смежности графа. Все веса ребер не превышают по модулю 10 000. Если ребра нет, то соответствующее число равно 100 000.

Описание выходного файла

В первой строке выходного файла выведите YES, если цикл существует, или NO в противном случае. При его наличии выведите во второй строке количество вершин в искомом цикле (считая одинаковые первую и последнюю) и в третьей строке — вершины, входящие в этот цикл, в порядке обхода.

Пример

cicle.in	cicle.out
2 0 -1 -1 0	YES 3 1 2 1

Задача B. Выпуклая оболочка

Входной файл:	`convex.in`
Выходной файл:	`ccnvex.out`
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

На плоскости даны N точек. Вам требуется построить выпуклую оболочку данного множества точек. Если N нечетно, то оболочку нужно выводить в порядке обхода по часовой стрелке, иначе — против часовой стрелки.

Описание входного файла

Первая строка содержит количество точек N, $1 \le N \le 20\,000$. Каждая из последующих N строк содержит два целых числа — координаты x_i и y_i. Все числа по модулю не превосходят 10^4.

Описание выходного файла

Выходной файл должен иметь тот же формат, что и входной и, должен содержать выпуклую оболочку. Количество точек в выходном файле должно быть минимально возможным.

Пример

convex.in	ccnvex.out
4 0 0 3 4 3 1 6 0	3 6 0 3 4 0 0

Задача C. Заправки

Входной файл:	`filling.in`
Выходной файл:	`filling.out`
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

В стране N городов, некоторые из которых соединены между собой дорогами. Для того, чтобы проехать по одной дороге, требуется один бак бензина. В каждом городе бак бензина имеет разную стоимость. Вам требуется добраться из первого города в N-ый, потратив как можно меньшее количество денег.

При этом есть еще канистра для бензина, куда входит столько же бензина, сколько входит в бак. В каждом городе можно заправить бак, залить бензин в канистру, залить и туда и туда, или же перелить бензин из канистры в бак.

Описание входного файла

Во входном файле записано сначала число N ($1 \le N \le 100$), затем идет N чисел, i-ое из которых задает стоимость бензина в i-ом городе (все это целые числа из диапазона от 0 до 100). Затем идет число M — количество дорог в стране, далее идет описание самих дорог. Каждая дорога задается двумя числами — номерами городов, которые она соединяет. Все дороги двухсторонние (то есть по ним можно ездить как в одну, так и в другую сторону), между двумя городами всегда существует не более одной дороги, не существует дорог, ведущих из города в себя.

Описание выходного файла

В выходной файл выведите одно число — суммарную стоимость маршрута или -1, если добраться невозможно.

Пример

filling.in	filling.out
4 1 10 2 15 4 1 2 1 3 4 2 4 3	2
4 1 10 2 15 0	-1

Задача D. Флойд-существование

Входной файл:	`floyd.in`
Выходной файл:	`floyd.out`
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

Дан ориентированный взвешенный граф. По его матрице смежности нужно для каждой пары вершин определить, существует ли кратчайший путь между ними или нет.

Комментарий: Кратчайший путь может не существовать по двум причинам:

- Нет ни одного пути
- Есть пути сколь угодно маленького веса

Описание входного файла

В первой строке входного файла записано единственное число: N ($1 \le N \le 100$) — количество вершин

графа. В следующих N строках по N чисел — матрица смежности графа (j-ое число в i-ой строке соответствует весу ребра из вершины i в вершину j): число 0 обозначает отсутствие ребра, а любое другое число — наличие ребра соответствующего веса. Все числа по модулю не превышают 100.

Описание выходного файла

Выведите N строк по N чисел. j-ое число в i-ой строке должно соответствовать кратчайшему пути из вершины i в вершину j. Число должно быть равно 0, если пути не существует, 1 — если существует кратчайший путь, и 2 — если пути существуют, но бывают пути сколь угодно маленького веса.

Пример

floyd.in	floyd.out
5 0 1 2 0 0 1 0 3 0 0 2 3 0 0 0 0 0 0 4 -1 0 0 0 -1 0	1 1 1 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 2 2 0 0 0 2 2

Задача E. Форматирование таблицы

Входной файл:	formatting.in
Выходной файл:	formatting.out
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

Вам предлагается отформатировать таблицу, данную во входном файле.

Описание входного файла

Первая строка входного файла содержит несколько букв l,c,r. Их количество равно количеству столбцов в таблице. Каждая буква задает расположение текста в соответствующем столбце (l значит, что текст сдвинут до упора влево, c — что текст расположен по середине, r — что текст сдвинут вправо). Далее следуют не менее двух и не более 100 строк данных, каждая из которых задает соответствующую строку таблицы. Каждая строка содержит несколько записей, разделенных амперсантом (‘&’) Количество записей в каждой строке равно количеству столбцов. Каждая запись должна располагаться в соответствующем столбце. Первая строка данных задает заголовок таблицы, а остальные — тело таблицы. Знак ‘&’ не содержится в ячейках таблицы.

Ограничения:

- длина любой строки входного файла не превышает 250 символов
- в таблице не более 100 строк
- в таблице не более 100 столбцов

Описание выходного файла

В выходной файл выведите таблицу в соответствии с форматом, приведенным в примере. Примечания:

- самоя длинная запись должна быть отделена от правого и левого краев ячейки ровно одним пробелом.
- если запись не может быть расположена точно посередине, то все дополнительные пробелы появляются справа от нее.

Пример

formatting.in

```
lr
EXPRESSION&RESULT
2+2&4
2+2*2&6
```

formatting.out

```
+------------+--------+
| EXPRESSION | RESULT |
+------------+--------+
| 2+2        |      4 |
| 2+2*2      |      6 |
+------------+--------+
```

formatting.in

```
lcr
LEFT&CENTER&RIGHT
1&2&3
the&the&the
longest&longest&longest
kitten&kitten&kitten
blitz&blitz&blitz
```

formatting.out

```
+---------+---------+---------+
| LEFT    | CENTER  |   RIGHT |
+---------+---------+---------+
| 1       |    2    |       3 |
| the     |   the   |     the |
| longest | longest | longest |
| kitten  | kitten  |  kitten |
| blitz   |  blitz  |   blitz |
+---------+---------+---------+
```

Задача F. Поколение комбинаторов

Входной файл:	geneartion.in
Выходной файл:	geneartion.out
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

Сочетанием из n элементов по k называется убывающая последовательность из k чисел из диапазона от 1 до n.

Сгенерируйте все сочетания из n элементов по k в анти-лексикографическом порядке.

Описание входного файла

Во входном файле содержатся два целых числа n и k. $1 \le k \le n \le 15$.

Описание выходного файла

В выходной файл выведите $\binom{n}{k}$ строк — все сочетания из n элементов по k в анти-лексикографическом порядке. Лишние пробелы и переводы строк не допустимы.

Пример

geneartion.in	geneartion.out
3 2	3 2 3 1 2 1

Задача G. Лабиринт знаний

Входной файл:	maze.in
Выходной файл:	maze.out
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

Участникам сборов подарили билеты на аттракцион "Лабиринт знаний". Лабиринт представляет собой N комнат, занумерованных от 1 до N, между некоторыми из которых есть двери. Когда человек проходит через дверь, показатель его знаний изменяется на определенную величину, фиксированную для данной двери. Вход в лабиринт находится в комнате 1, выход — в комнате N. Каждый участник сборов проходит лабиринт ровно один раз и наибрает некоторое количество знаний (при входе в лабиринт этот показатель равен нулю). Ваша задача — показать наилучший результат.

Описание входного файла

Первая строка входного файла содержит целые числа N ($1 \le N \le 2\,000$) — количество комнат и M ($1 \le M \le 10\,000$) — количество дверей. В каждой из следующих M строк содержится описание двери — номера комнат, из которой она ведет и в которую она ведет (через дверь в лабиринте можно ходить только в одну сторону), а также целое число, которое прибавляется к количеству знаний при прохождении через дверь (это число по модулю не превышает 10 000). Двери могут вести из комнаты в нее саму, между двумя комнатами может быть более одной двери.

Описание выходного файла

В выходной файл выведите "`:)`" — если можно получить неограниченно большой запас знаний, "`:(`" — если лабиринт пройти нельзя, и максимальное количество набранных знаний в противном случае.

Пример

maze.in	maze.out
2 2 1 2 5 1 2 -5	5

Задача H. В поисках паросочетания

Входной файл:	pairs.in
Выходной файл:	pairs.out
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

Ваша задача — найти максимальное паросочетание в двудольном графе.

Описание входного файла

В первой строке входного файла содержатся два целых числа n и m ($1 \le n, m \le 250$) — количество вершин в долях A и B соответственно.

Далее идут n строк с описанием ребер. i-ая вершина из A описывается в $i+1$-ой строке. Каждая строка содержит номера вершин из B, соседних с i-ой вершиной из A. Вершины в долях нумеруются независимо. Каждая строка заканчивается числом 0.

Описание выходного файла

Первая строка выходного файла должна содержать целое число l — количество ребер в максимальном паросочетании. Далее должны идти l строк по два числа u_j, v_j — ребра из паросочетания.

Пример

pairs.in	pairs.out
2 2 1 2 0 2 0	2 1 1 2 2

Задача I. Скобки

Входной файл:	parentheses.in
Выходной файл:	parenthesis.out
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

Правильной скобочной последовательностью называется такая последовательность круглых и квадратных скобок, которая могла бы встречаться в каком-нибудь арифметическом выражении. Например, последовательности `()[]` и `([])` являются правильными скобочными последовательностями, а последовательности `(][)` и `)()(` — нет. Считается, что круглые и квадратные скобки равноправны, но круглой открывающей скобке обязательно должна соответствовать круглая закрывающая, а квадратной — квадратная.

Скобки считаются упорядоченными в таком порядке: '`(`', '`)`', '`[`', '`]`'. Например, '`)`' больше чем '`(`', но меньше чем '`[`', а максимальной из всех скобок считается '`]`'.

Говорят, что одна скобочная последовательность *лексикографически меньше* другой, если существует такое число k, что первые k скобок в них совпадают, а $k+1$-ая скобка в первой меньше, чем во второй.

Программист Вася выписал на бумажке все правильные скобочные последовательности длины $2n$ таким образом, что каждая написанная скобочная последовательность лексикографически меньше следующей за ней. Он очень обрадовался, что теперь может по *лексикографическому номеру* скобочной последовательности найти саму последовательность (последовательности, записанные на бумажке, нумеруются, начиная с 1). Однако бумажка потерялась... А повторять эту огромную работу Васе лень... Помогите бедному Васе написать программу, которая бы выдавала правильную скобочную последовательность по ее лексикографическому номеру!

Описание входного файла

Во входном файле задано число n ($1 \le n \le 20$) и произвольное натуральное число A — лексикографический

номер искомой последовательности.

Описание выходного файла

В первой строке выходного файла должна содержаться искомая скобочная последовательность.

Пример

parentheses.in	parenthesis.out
2 1	(())
2 3	()[]

Задача J. Праздничная иллюминация

Входной файл:	party.in
Выходной файл:	party.out
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

Для того, чтобы устроить праздничную иллюминацию по поводу проведения зимних сборов, у нас есть N разноцветных лампочек, занумерованных от 1 до N. Лампочки управляются четырьмя кнопками.

Кнопка 1. При нажатии на эту кнопку, все лампочки меняют состояние. Те, которые были включены, выключаются, а те, которые выключены — включаются.

Кнопка 2. При нажатии на эту кнопку меняют свое состояние лампочки с нечетными номерами.

Кнопка 3. При нажатии на эту кнопку меняют свое состояние лампочки с четными номерами.

Кнопка 4. При нажатии на эту кнопку меняют свое состояние все лампочки с номерами $3K + 1 (K \geq 0)$, т.е. 1,4,7...

Есть счетчик C, который записывает суммарное количество нажатий на кнопки. В начале праздника все лампочки включены, и счетчик C сброшен в 0.

По значению счетчика C и конечному состоянию некоторых лампочек необходимо найти все возможные конечные конфигурации лампочек.

Описание входного файла

Первая строка входного файла содержит число N ($1 \leq N \leq 100$). Во второй находится значение счетчика C ($1 \leq C \leq 10\,000$). В третьей строке записаны номера лампочек, которые включены. Строка заканчивается числом -1. В четвертой строке в таком же формате заданы лампочки, которые выключены.

Описание выходного файла

В выходной файл выведите всевозможные конечные состояния лампочек, по одному в строке. Каждая строка содержит N символов — 0, если соответствующая лампочка выключена, и 1 в противном случае. Конфигурации нужно выводить в лексикографическом порядке.

Пример

party.in	party.out
10 1 -1 7 -1	0000000000 0101010101 0110110110

Задача K. Прямоугольники

Входной файл:	rect.in
Выходной файл:	rect.out
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

Вася очень любит головоломки, в которых необходимо посчитать количество прямоугольников на большой картинке с горизонтальными и вертикальными отрезками. Сегодня он нарисовал прямоугольник $m \times n$ на прямоугольной решетке на своем ноутбуке, несколько таких отрезков внутри, и предложил своему брату Саше решить ее. Но Саша не хочет проводить время, считая прямоугольники, которых так много, поэтому он решил использовать свой компьютер для этого. Он попросил Вас написать программу, которая найдет количество прямоугольников на прямоугольной решетке с несколькими вертикальными и горизонтальными отрезками на ней.

Описание входного файла

Входной файл состоит из нескольких наборов тестов. Каждый тест начинается с со значений m и n. В следующей строке находится единственное число k — количество нарисованных отрезков. Далее идут описания отрезков. Каждый отрезок описывается четырьмя целыми числами x_1 y_1 x_2 y_2, где либо $x_1 = x_2$, либо $y_1 = y_2$. Все x из диапазона от 0 до m, а y — от 0 до n. Значения n и m не превосходят 100.

Входной файл заканчивается строкой, состоящей из двух 0.

Саша довольно неаккуратный. Он мог поместить один и тот же отрезок несколько раз во входной файл. Так же он мог разбить отрезок на несколько, возможно перекрывающихся, частей.

Описание выходного файла

Для каждого теста из входного набора необходимо вывести количество прямоугольников на решетке, в соответствии с форматом, приведенным в примере.

Пример

rect.in
2 2 2 0 1 2 1 1 1 1 2 2 2 1 0 1 1 1 0 0
rect.out
Grid 1: The number of rectangles is 5. Grid 2: The number of rectangles is 1.

Задача L. Корень из перестановки

Входной файл:	root.in
Выходной файл:	root.out
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

Девочка Маша написала на доске последовательность из N различных натуральных чисел из диапазона от 1 до N. Затем к доске подошел хулиган Вася и написал свою последовательность: если на i-м месте в Машиной последовательности стоит число j, а на j — число k, то на i-ое место в своей последовательности Вася пишет число k. Затем он безжалостно стер Машину последовательность. Помогите бедной девушке восстановить утерянную последовательность!

Описание входного файла

В первой строке входного файла записано число N ($1 \le N \le 1\,000$). Во второй строке идут N чисел — последовательность, написанная Васей.

Описание выходного файла

В выходной файл выведите последовательность, написанную Машей. Если существует несколько вариантов — выведите любой.

Пример

root.in	root.out
3 2 3 1	3 1 2

Задача M. Сортировка

Входной файл:	sort.in
Выходной файл:	sort.out
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

Мария Ивановна, учитель средней школы 5 села Уборкино заполняет, классный журнал. Но она неожиданно столкнулась с проблемой: список имен, который у нее есть, не отсортирован. Помогите ей! Список необходимо отсортировать в алфавитном порядке по фамилиям. Люди с одинаковой фамилией должны идти в том же порядке, в котором они идут в исходном списке.

Описание входного файла

Первая строка входного файла содержит натуральное чи сло N ($1 \le N \le 20\,000$) — количество человек в классе. Далее идут N строк, содержащих по два слова, записанных через пробел: фамилия и имя ученика. В записи фамилии и имени встречаются только буквы латинского алфавита, причем первая буква всегда большая, а остальные — маленькие. Длина фамилии не менее 1 и не более 20. Длина имени не менее 1 и не более 20.

Описание выходного файла

Выходной файл должен полностью удовлетворять формату входного файла и должен содержать тот же список, но отсортированный согласно условию.

Пример

sort.in	sort.out
4 Pupkin Vasya Ivanov Petya Iskandeev Semil Ivanov Roma	4 Iskandeev Semil Ivanov Petya Ivanov Roma Pupkin Vasya

Задача N. Подстрока

Входной файл:	substr.in
Выходной файл:	substr.out
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

Ваша задача — найти все вхождения данной строки в длинный текст.

Описание входного файла

Входной файл начинается со строки, которую нужно искать. Все символы входного файла до звездочки * — строка, котрую нужно искать. Все символы, начиная с первой звездочки и до конца файла — текст, в котором нужно искать.

Все символы с ASCII кодами 33...126 (кроме '*') допустимы и могут встречаться как в строке, так и в тексте. Все пробелы и символы перевода строки во входном файле должны игнорироваться.

Длина строки не превосходит 10 000, а длина текста не превосходит 200 000 символов.

Описание выходного файла

Выведите в выходной файл в возрастающем порядке все точки вхождения данной строки в данный текст. Точка вхождения — это количество символов от начала текста до первого символа вхождения, увеличенное на 1. Если вхождение начинается сразу после звездочки, то точка его вхождения равна 1.

Пример

substr.in	substr.out
aa* bb aaa	3 4

Задача O. Сумма

Входной файл:	sum.in
Выходной файл:	sum.out
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

Однажды Петя написал у себя в тетради последовательность 1,2,3,4,5,6,7,... Потом он просуммировал числа со 2-го по 4-ое и получил 9. Затем он просуммировал числа с 4-го по 5-ое и тоже получил 9. Его заинтересовал вопрос, а сколько существует различных отрезков с заданной суммой K. Как истинного программиста, Петю интересуют только $K = 2^N$. Помогите ему!

Описание входного файла

Входной файл содержит число N, $0 \le N \le 60$.

Описание выходного файла

В выходной файл выведите количество различных отрезков с суммой 2^N.

Пример

sum.in	sum.out
1	1

Задача P. Два коня

Входной файл:	twohorse.in
Выходной файл:	twohorse.out
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

На стандартной шахматной доске (8×8) живут 2 шахматных коня: Красный и Зеленый. Обычно они беззаботно скачут по просторам доски, пощипывая шахматную травку, но сегодня особенный день: у Зеленого коня День Рождения. Зеленый конь решил отпраздновать это событие вместе с Красным. Но для осуществления этого прекрасного плана им нужно оказаться на одной клетке. Заметим, что Красный и Зеленый шахматные кони сильно отличаются от черного с белым: они ходят не по очереди, а одновременно, и если оказываются на одной клетке, никто никого не съедает. Сколько ходов им потребуется, чтобы насладиться праздником?

Описание входного файла

Во входном файле содержатся координаты коней, записанные по стандартным шахматным правилам (т.е. двумя символами — маленькая латинская буква (от a до h) и цифра (от 1 до 8), задающие столбец и строку соответсвенно).

Описание выходного файла

Выходной файл должен содержать наименьшее необходимое количество ходов, либо -1, если кони не могут встретиться.

Пример

twohorse.in	twohorse.out
a1 a3	1

Задача Q. Окна

Входной файл:	widows.in
Выходной файл:	widows.out
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

На экране нарисовано n прямоугольных окон. Ваша задача — найти площадь области, покрытой всеми окнами.

Описание входного файла

Первая строка входного файла содержит целое число n ($1 \le n \le 30000$). Далее идут n строк по 4 целых числа x_1 y_1 x_2 y_2, где (x_1, y_1) — координаты левого верхнего угла окна, а (x_2, y_2) — правого нижнего. Все координаты по модулю не превосходят 5 000. Заметьте, что ось y на экране направлена вниз.

Описание выходного файла

В выходной файл выведите единственное целое число — площадь, покрытую всеми окнами.

Пример

widows.in	widows.out
2 0 0 3 2 1 1 4 4	2

Рисунок к примеру