Задача A. Задача из нового ЕГЭ

Ограничение по времени:	1 second
Ограничение по памяти:	256 megabytes

У исполнителя две команды:

1. прибавь A
2. прибавь B

Первая из них увеличивает число на экране на A, вторая — на B. Программа для этого исполнителя — это последовательность команд. Сколько различных чисел можно получить из числа 1 с помощью программы, которая содержит ровно C команд?

Формат входных данных

Вводятся три целых числа A, B, C ($-1000 \le A \le 1000$, $-1000 \le B \le 1000$, $1 \le C \le 1000$)

Формат выходных данных

Выведите количество различных чисел, которые можно получить из числа 1 с помощью программы ровно из C команд.

Примеры

Входные данные	Выходные данные
3 -2 5	6
0 0 10	1

Система оценки

Тесты в этой задаче состоят из трех групп:

0. Тесты 1 и 2 . Тесты из условия. Оцениваются 0 баллов.
1. Тесты 3 – 18. Тесты с ограничением $C \le 16$. Эти тесты оцениваются в 30 баллов, при этом баллы ставятся только за прохождение всех тестов группы.
2. Тесты 19 – 44. Эти тесты оцениваются в 70 баллов, при этом баллы ставятся только за прохождение всех тестов группы.

Задача B. Лексикографически наименьшее троичное

Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Числа от A до B включительно записали в троичной системе счисления без ведущих нулей. Каждое число записали в отдельной строке. Затем полученные строки расположили в лексикографическом порядке. Определите, какое число окажется на первом месте.

Числа A и B задаются в десятичной системе счисления, ответ также нужно вывести в десятичной системе счисления.

Например, пусть $A = 2$ и $B = 12$. Тогда:

Числа в десятичной системе	Числа в троичной системе	Троичные числа в лексикографическом порядке
2	2	10
3	10	100
4	11	101
5	12	102
6	20	11
7	21	110
8	22	12
9	100	2
10	101	20
11	102	21
12	110	22

Таким образом, в этом примере ответом будет являться число 3 (в троичной системе счисления записывающееся как 10).

Формат входных данных

Заданы натуральные числа A и B ($1 \le A \le B \le 10^{15}$).

Формат выходных данных

Выведите ответ задачи.

Примеры

Входные данные	Выходные данные
2 12	3

Система оценки

Тесты в этой задаче состоят из трех групп:

0. Тест 1. Тест из условия. Оценивается 0 баллов

1. Тесты 2 – 14. Тесты с ограничением $B - A \le 10^6$. Эти тесты оцениваются в 30 баллов, при этом баллы ставятся только за прохождение всех тестов группы.

2. Тесты 15 – 32. Эти тесты оцениваются в 70 баллов, при этом баллы ставятся только за прохождение всех тестов группы.

Задача C. Задача из старого ЕГЭ

Ограничение по времени:	2 seconds
Ограничение по памяти:	256 megabytes

Строки (цепочки цифр) создаются по следующему правилу.

Первая строка состоит из одного символа — цифры «1». Каждая из последующих цепочек создается такими действиями: в очередную строку записывается сначала номер строки (без ведущих нулей), а затем два раза приписывается предыдущая строка.

Вот первые 4 строки, созданные по этому правилу:

1. 1
2. 211
3. 3211211
4. 432112113211211

А, например, 11-я строка будет начинаться так:

11. 1110987...

По заданным N и K определите, какая цифра будет стоять в N-ой строке на K-ом месте, либо что длина строки меньше K.

Формат входных данных

Вводятся два натуральных числа N и K ($1 \le N \le 100000$, $1 \le K \le 10^{15}$).

Формат выходных данных

Определите, какая цифра стоит в N-ой строке на K-ом месте и выведите ее. Если длина N-ой строки меньше K, выведите -1 (минус один).

Примеры

Входные данные	Выходные данные
3 2	2
11 4	0
2 25	-1

Система оценки

Тесты в этой задаче состоят из пяти групп:

0. Тесты 1 – 3. Тесты из условия. Оцениваются 0 баллов.
1. Тесты 4 – 21. Тесты с ограничениями $N < 10$, $K < 100$, K не превосходит длину N-ой строки. Эти тесты оцениваются в 20 баллов, при этом баллы ставятся только за прохождение всех тестов группы.
2. Тесты 22 – 47. Тесты с ограничениями $N < 100$, K не превосходит длину N-ой строки. Эти тесты оцениваются в 20 баллов, при этом баллы ставятся только за прохождение всех тестов группы.
3. Тесты 48 – 67. Тесты с ограничениями $N < 1000$. Эти тесты оцениваются в 30 баллов, при этом баллы ставятся только за прохождение всех тестов группы.
4. Тесты 68 – 91. Эти тесты оцениваются в 20 баллов, при этом баллы ставятся только за прохождение всех тестов группы.

Задача D. Труба

Ограничение по времени:	2 seconds
Ограничение по памяти:	256 megabytes

Трубопровод длиной L километров строят из труб длиной A километров каждая (последняя труба может иметь меньшую длину). Каждые B километров устанавливается подпорка для крепления трубопровода (то есть подпорки ставятся в точках B, $2B$, $3B$, $\ldots$; в начале и в конце трубопровода подпорки не ставятся).

Определите, сколько раз на протяжении трубопровода точка установки подпорки **в точности совпадает** с местом стыка труб.

В первом из приведенных ниже примеров стык труб и подпорка окажутся в одной точке в точках 3, 6, 9, во втором примере — в точках 4.02 и 8.04.

Формат входных данных

В первой строке записано число L. Во второй строке — число A. В третьей строке — число B. Строки не содержат никаких символов, кроме цифр и десятичной точки. После третьей строки идет символ перевода на новую строку.

Все три числа положительные, не обязательно целые. В записи каждого из чисел используется не более 9 десятичных цифр, и, быть может, десятичная точка. Числа L, A, B заданы **точно**.

Формат выходных данных

Выведите одно целое число — ответ задачи

Примеры

Входные данные	Выходные данные
10 1 3	3
10.5 1.005 .004	2

Система оценки

Тесты в этой задаче состоят из трех групп:

0. Тесты 1 и 2. Тесты из условия. Оцениваются 0 баллов
1. Тесты 3 – 32. В каждом из этих тестов все числа целые. Каждый тест оценивается в 1 балл, тесты оцениваются независимо друг от друга.
2. Тесты 33 – 102. Эти тесты оцениваются в 70 баллов, при этом баллы ставятся только за прохождение всех тестов группы.

Задача E. Количество кратчайших путей

Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

В городе есть N площадей, соединенных дорогами. Известна длина каждой дороги.

Посчитайте количество способов добраться с площади A до площади B так, чтобы пройденный путь был минимален.

Формат входных данных

Задано число N — количество площадей в городе и M — количество улиц. Далее идет описание улиц. Каждая улица задается тремя числами: номерами площадей, которые она соединяет, и длиной. Никакая улица не соединяет площадь с самой собой. Одни и те же площади могут быть соединены разными улицами (в том числе, эти улицы могут иметь разную длину). По каждой улице можно ездить как в прямом, так и в обратном направлении. Далее заданы числа A и B.

Ограничения: $1 \le N \le 1000$, $1 \le M \le 100000$, $1 \le A \le N$, $1 \le B \le N$. Длины улиц выражаются натуральными числами, не превышающими 100000.

Формат выходных данных

Выведите количество кратчайших путей между площадями A и B. Если с площади A нельзя добраться до площади B, выведите 0.

Примеры

Входные данные	Выходные данные
4 5 1 4 2 1 2 2 3 4 1 2 4 2 1 3 1 1 4	2
5 4 1 2 5 4 5 4 1 3 4 2 3 2 2 5	0

Система оценки

Тесты в этой задаче состоят из трех групп:

0. Тесты 1 и 2. Тесты из условия. Оценивается 0 баллов
1. Тесты 3 – 18. Тесты с ограничением $N \le 500$, $M \le 20000$. Эти тесты оцениваются в 30 баллов, при этом баллы ставятся только за прохождение всех тестов группы.
2. Тесты 19 – 25. Эти тесты оцениваются в 30 баллов, при этом баллы ставятся только за прохождение всех тестов группы.
3. Тесты 26 – 30. Каждый из этих тестов оценивается в 8 баллов. Баллы за эти тесты ставятся независимо, но только при условии, что проходят все тесты первой или второй группы.

Задача F. Фибоначчиева последовательность

Ограничение по времени: 1 second
Ограничение по памяти: 256 megabytes

Последовательность чисел a_1, a_2, … является фибоначчиевой, то есть для любого $i \geq 3$ верно, что

$$a_i = a_{i-1} + a_{i-2}.$$

Заданы два члена этой последовательности с номерами i и j. Найдите k-ый член последовательности.

Во втором тесте фибоначчиева последовательность: $3, -1, 2, 1, 3, 4, ...$

Формат входных данных

Заданы числа i, j, k, a_i, a_j. Ограничения: i, j, k — натуральные числа, не превышающие 10^6, i и j всегда различны. a_i и a_j — целые числа, не превышающие по модулю $2 \cdot 10^9$.

Формат выходных данных

Выведите a_k. Гарантируется, что входные данные таковы, что все члены последовательности — целые числа, и a_k не превышает по модулю $2 \cdot 10^9$.

Примеры

Входные данные	Выходные данные
1 2 6 1 1	8
3 6 4 2 4	1

Система оценки

Тесты в этой задаче состоят из четырех групп:

0. Тесты 1 и 2. Тесты из условия. Оценивается 0 баллов

1. Тесты 3 – 12. Тесты с ограничениями $j = i \pm 1$, $1 \leq i, j, k \leq 30$, а a_i, a_j, не превышают 10^6. Эти тесты оцениваются в 30 баллов, при этом баллы ставятся только за прохождение всех тестов группы.

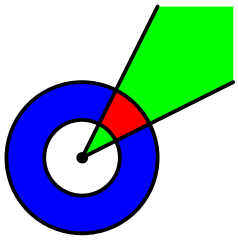
2. Тесты 13 – 22. Тесты для случаев, когда i, j, k не превышают 30, и a_i, a_j не превышают по модулю 10^6. Эти тесты оцениваются в 40 баллов, при этом баллы ставятся только за прохождение всех тестов группы.

3. Тесты 23 – 32. Тесты off-line группы (то есть решения будут тестироваться на этих тестах после окончания тура). Каждый тест оценивается в 3 балла.

Задача G. Полярные прямоугольники

Ограничение по времени: 1 second
Ограничение по памяти: 256 megabytes

Вася недавно изучил полярную систему координат. А именно, он изучил понятие полярного прямоугольника. Пусть задана стандартная декартова плоскость. Если на ней нарисовать две окружности с центром в начале координат, то область, находящаяся между ними, называется *кольцом* (на рисунке обозначена синим). Если на ней нарисовать два луча, то область, заметаемая первым лучом при движении ко второму, называется *углом* (т.е. область между этими двумя лучами, на рисунке обозначена зеленым). *Полярным прямоугольником* называется пересечение некоторого угла с некоторым кольцом (на рисунке обозначено красным).



Задано несколько полярных прямоугольников. Найдите площадь их пересечения. Помните, что пересечение полярных прямоугольников может состоять из нескольких частей!

Формат входных данных

В первой строке вводится целое число N — количество прямоугольников ($1 \le N \le 100\,000$). Далее в N строках содержится описание прямоугольников. Каждый прямоугольник описывается четверкой действительных чисел r_1, r_2, φ_1, φ_2, где r_1, r_2 обозначают радиусы окружностей, образующих кольцо ($r_1 < r_2$), а φ_1, φ_2 обозначают углы, образованные первым и вторым лучами с осью абсцисс, заданные в радианах. При этом заметается область от первого луча до второго в направлении против часовой стрелки (т.е. возрастания углов), даже в случае, когда $\varphi_1 > \varphi_2$. Все числа заданы максимум с шестью знаками после десятичной точки. Углы лежат в полуинтервале $[0, 2\pi)$, а радиусы не превосходят 10^6. Гарантируется, что $\varphi_1 \ne \varphi_2$.

Формат выходных данных

Выведите единственное число — площадь искомого пересечения. Ответ будет считаться правильным, если его абсолютная или относительная погрешность не будет превышать 10^{-6}.

Примеры

Входные данные	Выходные данные
2 1 3 0 3 2 4 1.5 4.5	3.7500000000
2 1 2 0 3 1 2 2 1	3.0000000000

Система оценки

Тесты в этой задаче состоят из четырех групп:

0. Тесты 1 и 2. Тестами из условия. Оцениваются 0 баллов.

1. Тесты 3 – 31. Тесты, в которых $N = 2$. Эти тесты оцениваются в 30 баллов, при этом баллы ставятся только при прохождении всех тестов группы.

2. Тесты 32 – 51. Тесты с ограничением $N \le 5000$. Эти тесты оцениваются в 30 баллов, при этом баллы ставятся только при прохождении всех тестов первой и второй групп.

3. Тесты 52 – 61. Эти тесты оцениваются в 40 баллов, баллы ставятся только при прохождении всех тестов первой, второй и третьей групп.

Задача H. Тетрис

Ограничение по времени:	2 seconds
Ограничение по памяти:	256 megabytes

Недавно программист Антон узнал о существовании игры "Тетрис++". Игра очень похожа на обычный, всем привычный тетрис, но есть несколько отличий.

Правила игры "Тетрис++":

Игровое поле представляет собой клетчатое поле высотой N и шириной M квадратиков, на дне которого могут находиться квадратики разных цветов. В начале игры из левого верхнего угла начинает падать фигурка, имеющая вид вертикального прямоугольника размера 3x1, каждый из квадратиков которого окрашен в некоторый цвет.

В момент появления фигурки на доске игрок может моментально сдвинуть ее на произвольное целое количество клеток вправо и сдвинуть клеточки фигурки циклически вверх произвольное количество раз. Дальше фигурка падает вниз до тех пор, пока не "встает"на лежащий квадратик или на дно игрового поля.

После падения фигурки начинается самое интересное: всевозможные последовательности, состоящие не менее чем из 3 квадратиков одного цвета, лежащих подряд на одной вертикали, горизонтали или диагонали, одновременно мгновенно исчезают. Все цветные квадратики, расположенные выше исчезнувших, мгновенно и одновременно падают вниз. Если после падения образуются новые последовательности из 3 и более квадратиков одного цвета, то они снова одновременно исчезают, и так далее.

Цель игры оставить на доске минимально возможное количество цветных квадратиков.

Помогите Антону определить, на сколько позиций нужно переместить падающую фигурку вправо и сколько раз циклически её сдвинуть вверх, чтобы в конце концов на поле осталось как можно меньше квадратиков.

Формат входных данных

В первой строке расположены числа N и M, разделённые пробелом ($4 \le N \le 26$, $1 \le M \le 15$). В каждой из следующих N строк расположены по M чисел, разделённых пробелами, описывающих начальное состояние игрового поля (сверху вниз). Числа от 1 до 9 при этом обозначают цвет соответствующего квадратика, а число 0 — отсутствие цветного квадратика в данном месте игрового поля. В последних трёх строках расположены три числа — цвета (от 1 до 9) квадратиков (сверху вниз) падающей фигурки.

Гарантируется, что верхние 4 ряда игрового поля не содержат цветных квадратиков.

Формат выходных данных

Требуется вывести 3 числа, отделенных друг от друга пробелами. Первое число — количество единиц, на которое требуется передвинуть фигурку вправо (от 0 до $M - 1$). Второе число — требуемое количество циклических сдвигов (от 0 до 2). Третье — суммарное количество исчезнувших в результате падения этой фигурки квадратиков. Если оптимальных решений больше одного, должно быть выведено решение с минимальным числом перемещений фигурки вправо, а если таких решений несколько, решение с минимальным числом перемещений фигурки и минимальным сдвигом.

Примеры

Входные данные	Выходные данные
6 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 1 1 1	1 0 6
8 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 2 3 1 0 0 0 4 2 3 3 0 5 4 5 4 5 1 3 2 1	0 1 10

Система оценки

Тесты в этой задаче состоят из трех групп:

0. Тесты 1 и 2. Тесты из условия. Оцениваются 0 баллов.
1. Тесты 3 – 24. Тесты с дополнительным условием, что все квадратики одного цвета. Эти тесты оцениваются в 30 баллов, при этом баллы ставятся только за прохождение всех тестов группы.
2. Тесты 25 – 97. Эти тесты оцениваются в 70 баллов, при этом баллы ставятся только за прохождение всех тестов группы.

Задача I. Уравнение

Ограничение по времени: 1 second
Ограничение по памяти: 256 megabytes

Программист Вова давно интересуется различными задачами со строками. В этой задаче будем считать, что строка — произвольная последовательность ASCII-символов с кодами от 33 до 126.

Недавно Вова узнал об операциях со строками:

- сложение строк, обозначается символом "+". Например, `Doctor+Who=DoctorWho`.
- умножение строк, обозначается символом "*". Если $S = A*B$, то $S = a_1+B+a_2+B+\ldots+a_n+B$, где a_i — i-ый символ строки A (нумерация символов строки начинается с 1). Например, `aca*x=axcxax`.

После этого Вова подумал: почему бы не придумать уравнения, где неизвестной величиной будет строка. После чего Вова стал решать придуманные им уравнения.

Перейдем к формальным определениям. Назовем уравнением последовательность $a_1 \circ a_2 \circ \ldots \circ a_n = b$, где a_i — либо неизвестная строка (обозначается "?"), либо какая-то определенная непустая строка (записывается символами строки без каких-либо дополнительных символов или ограничителей), b — какая-то определенная непустая строка, а на местах значков $\circ$ могут стоять знаки операций сложения или умножения (причем в разных местах могут стоять разные знаки). Все действия выполняются строго слева направо. Скобки не влияют на приоритет выполнения операций. Гарантируется, что существует i, что a_i = "?". Символы "?", "+", "*", "=" не используются как символы известных строк. Также всем a_i и b соответствуют непустые строки.

Требуется решить уравнение подобного вида, если известно, что существует непустое решение.

Формат входных данных

В первой строке находится строка S, задающая уравнение. Количество операций не превосходит 200000, длина строки S не более чем 300000 символов. В уравнении всегда присутствует хотя бы один знак "?".

Формат выходных данных

Напечатайте значение неизвестной строки.

Примеры

Входные данные	Выходные данные
ab*?+ab=aabbababbaab	abba

Система оценки

Тесты в этой задаче состоят из четырех групп:

0. Тест 1. Тест из условия. Оценивается 0 баллов.
1. Тесты 2 – 26. Тесты, где во входной строке символ "?" встречается ровно один раз. Эти тесты оцениваются в 20 баллов, при этом баллы ставятся только за прохождение всех тестов группы.
2. Тесты 27 – 50. Тесты, где в входной строке до символа "=" встречаются только символы "?", "+", "*". Эти тесты оцениваются в 20 баллов, при этом баллы ставятся только за прохождение всех тестов группы.
3. Тесты 51 – 78. Эти тесты оцениваются в 60 баллов, при этом баллы ставятся только за прохождение всех тестов группы.

Задача J. Юпитерианские ювелиры

Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Планета Юпитер славится своими искусными ювелирами. Самыми ценными во всей Солнечной системе считаются плоские украшения, представляющие собой набор бриллиантов, соединенных перемычками из золота. Перемычка — это прямой кусок проволоки, соединяющий два драгоценных камня. Хорошее украшение цельно, то есть бриллианты в нем нельзя разделить на две части так, что ни одна проволока не соединяла бы камни из разных частей.

По существующей традиции в каждое украшение ювелир вкрапляет один собственный фирменный изумруд с тремя ушками, к каждому из которых может прикрепляться одна золотая перемычка. Таким образом, изумруд, например, позволяет скрепить между собой два или три любых бриллианта. Юпитерианский Фаберже всегда вставляет свой изумруд в центр украшения, юпитерианский Леонардо — ближе к левому нижнему углу. Однако эти изумруды — не только фирменный знак, но и возможность сэкономить на золоте, поэтому самые хитрые и дотошные ювелиры вставляют свой камень так, чтобы сократить расходы ценного металла. Своих изумрудов у мастера пруд пруди,так что можно считать, что они достаются ему бесплатно. Иногда, если добавление изумруда потребует дополнительных расходов на золото, или даже просто не принесет выгоды, ювелиры жертвуют славой и обходятся без фирменной драгоценности.

Премьер-министр Юпитера каждый день требует от придворных ювелиров изготовить новое украшение для своей возлюбленной. Чтобы украшения не повторялись, обычно он сам придумывает им форму, то есть к моменту начала работы у ювелира есть план расположения всех камней. В последнее время премьеру показалось, что золота тратится слишком много, и он собирается нанять нового главного придворого ювелира. Старый мастер не хочет терять место, поэтому обратился к вам с просьбой помочь ему и составить оптимальную схему изготовления украшения (то есть такую, при которой будет потрачено меньше всего золотой проволоки).

Формат входных данных

В первой строке вводится целое число N — количество бриллиантов в украшении, которое требует изготовить премьер-министр. В следующих N строках заданы координаты бриллиантов в плане — по два числа на строке, разделенные пробелом.

Координаты во всех тестах — целые и не превосходят по модулю 10^4. Ограничения на N указаны в разделе "система оценки".

Формат выходных данных

В первой строке выведите одно вещественное число — суммарную длину отрезков из золотой проволоки в оптимальном плане.

В следующей строке выведите два вещественных числа — координаты изумруда. В следующей строке выведите K — число бриллиантов, соединенных с изумрудом, затем K чисел — их номера. Номера не должны повторяться. Если в оптимальной схеме изумруд отсутствует, K должно быть равно нулю, а координаты могут быть произвольными. Таким образом, K может быть равно 0, 2 или 3.

В следующей строке выведите число M — количество перемычек, соединяющих бриллианты. В следующих M строках выведите по два числа — номера бриллиантов, которые соединяет перемычка.

Бриллианты нумеруются от 1 до N.

Если верных ответов несколько, выведите любой.

Ваш ответ будет сравниваться с правильным с абсолютной или относительной точностью 10^{-6}.

Примеры

Входные данные	Выходные данные
3 0 0 4 0 0 3	6.766432 0.695788 0.751176 3 1 2 3 0
4 0 0 7 0 7 4 0 4	14.653516 6.008080 1.148322 3 1 2 3 1 1 4
3 0 0 2 0 -2 2	4.828427 0 0 0 2 1 2 1 3

Система оценки

Тесты в этой задаче состоят из четырех групп:

0. Тесты 1 – 3. Тесты из условия. Оцениваются 0 баллов.
1. Тесты 4 – 15. Тесты с ограничением $1 \le N \le 3$. Эти тесты оцениваются в 30 баллов, при этом баллы ставятся только за прохождение всех тестов группы.
2. Тесты 16 – 25. Тесты с ограничением $1 \le N \le 30$. Эти тесты оцениваются в 30 баллов, при этом баллы ставятся только за прохождение всех тестов этой и предыдущей группы.
3. Тесты 26 – 45. $1 \le N \le 250$. Тесты off-line группы (то есть решения будут тестироваться на этих тестах после окончания тура). Каждый тест оценивается в 2 балла. Решение тестируется на этих тестах только в случае успешного получения баллов за все тесты on-line групп.

Задача K. Коллекционеры монет

Ограничение по времени:	2 seconds
Ограничение по памяти:	512 megabytes

Программист Гриша коллекционирует монеты. В его коллекции K различных монет, пронумерованных от 1 до K. У Гриши есть друг Саша, который хочет одолжить у Гриши несколько монет. Саша довольно привередлив и монеты не стали исключением: есть N пар монет, которые по эстетическим соображениям Саши несовместимы друг с другом.

Подскажите Саше, сколько у него способов взять у Гриши 1, 2, 3 или 4 различных монеты так, чтобы среди взятых монет никакие две не были бы несовместимы.

Например, если у Гриши 3 монеты, и Саша считает несовместимыми 2 пары монет: $\langle 1, 2\rangle$ и $\langle 2, 3\rangle$, то Саша может взять у Гриши либо любую одну монету (3 способа), либо две монеты — 1 и 3 (любые другие будут несовместимы). А вот взять три монеты у него уже не получится (среди них окажутся несовместимые). Таким образом, в этом примере всего Саша может позаимствовать монеты 4 способами.

Формат входных данных

Сначала вводится число K ($1 \le K \le 5000$) — количество монет у Гриши. Предполагается, что все монеты Гриши занумерованы целыми числами от 1 до K. Затем следует число N ($0 \le N \le 5000$) — количество несовместимых пар. После него идёт N пар чисел $\langle r1, r2\rangle$, задающих несовместимые пары монет. Гарантируется, что $r1 \ne r2$ и все пары различные.

Формат выходных данных

Выведите одно число — общее количество вариантов у Саши взять монеты у Гриши.

Примеры

Входные данные	Выходные данные
5 3 1 2 2 3 3 4	15
3 2 1 2 2 3	4

Система оценки

Тесты в этой задаче состоят из четырех групп:

0. Тесты 1 и 2. Тесты из условия. Оцениваются 0 баллов.
1. Тесты 3 – 24. Тесты с дополнительными ограничениями $N \le 100$, $K \le 100$. Эти тесты оцениваются в 30 баллов, при этом баллы ставятся только за прохождение всех тестов группы.
2. Тесты 25 – 46. Тесты с дополнительными ограничениями $N \le 2000$, $K \le 2000$. Эти тесты оцениваются в 40 баллов, при этом баллы ставятся только за прохождение всех тестов группы.
3. Тесты 47 – 66. Эти тесты оцениваются в 30 баллов, при этом баллы ставятся только за прохождение всех тестов группы.