

Задача 1. Садоводы

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 MiB

На заводе купили землю под садоводческое товарищество и решили поделить её на треугольные участки. Два сотрудника решили проверить, одинаковые ли участки они получили. По спутниковым снимкам они определили координаты вершин своих участков и теперь хотят проверить, что участки одинаковые. Они считают, что два участка одинаковы, если один из другого можно получить перемещением и/или поворотом, но не переворотом (отражением относительно одной из сторон).

Формат входных данных

В единственной строке 12 целых чисел через пробел, все числа по модулю не превышают 10^9 . Каждая из 6 пар чисел – координаты вершины участка. Первые 3 пары – координаты первого участка, перечисленные в порядке обхода участка против часовой стрелки, следующие 3 пары – координаты второго участка, перечисленные в порядке обхода участка против часовой стрелки.

Формат выходных данных

В единственной строке «YES», если один участок из другого можно получить перемещением и/или поворотом, но не переворотом, и «NO» в противном случае.

Примеры

<i>standard input</i>	<i>standard output</i>
0 0 3 0 0 4 13 10 10 14 10 10	YES
0 0 3 0 0 4 13 10 10 14 10 11	NO

Задача 2. Какой участок лучше?

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 MiB

Раздавая садоводам треугольные участки, директор завода из предыдущей задачи обнаружил, что на некоторые участки спрос заметно больше. Он попробовал узнать, в чем причина, и обнаружил, что на заводе возникла мода на так называемые “хорошие” участки. “Хорошим” люди считали треугольный участок ненулевой площади с целочисленными длинами сторон a , b , c , такими, что $a < b < c$ и c нацело делится на $b - a$. Помогите директору определить, сколько всего может быть “хороших” участков, периметр которых не превышает заданной величины X .

Формат входных данных

В единственной строке одно целое число X , $1 \leq X < 10^{18}$.

Формат выходных данных

В единственной строке одно целое число – количество “хороших” участков с периметром, не превышающим X .

Примеры

<i>standard input</i>	<i>standard output</i>
12	1
100	9

Задача 3. Числоварь

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 MiB

Математики-языковеды решили провести эксперимент и вместо слов использовать числа. Для этого они взяли N слов из словаря и поставили им в соответствие числа от 1 до N , назвав их числами. При использовании чисел сразу возникла путаница, потому что запомнить, какое число что означает, было очень непросто. Возникла идея составить числоварь чисел, упорядочив их лексикографически. Помогите математикам-языковедам быстро определить, какое число будет K -ым в этом новом числоваре при нумерации с единицы.

Формат входных данных

В первой строке 2 целых положительных числа через пробел: N – количество чисел и K – номер искомого числа, $1 \leq N \leq 10^{100}$, $1 \leq K \leq N$.

Формат выходных данных

В первой и единственной строке число, которое окажется K -ым при лексикографической сортировке чисел от 1 до N включительно.

Примеры

<i>standard input</i>	<i>standard output</i>
30 10	18

Задача 4. Задачка из учебника

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 MiB

Читая старый учебник по программированию, начинающий программист встретил следующую задачу.

Дан массив натуральных чисел `pushed`, причем элементы в нем не обязательно являются уникальными. Есть изначально пустой стек. Элементы массива `pushed` можно добавлять в стек по-одному, не меняя порядок. Из стека можно извлекать элементы, помещая извлеченные элементы в массив `popped` в порядке извлечения. Операции добавления в стек и извлечения из него можно применять в произвольном порядке. Для заданного массива `popped` определите, мог ли он быть получен из массива `pushed` описанным способом.

Например, если `pushed = [1, 2, 3]`, `popped = [2, 3, 1]`, то используя команды `push` (поместить в стек) и `pop` (извлечь из стека) следующим образом: `push push pop push pop pop` (содержимое стека $[] \rightarrow [1] \rightarrow [1, 2] \rightarrow [1] \rightarrow [1, 3] \rightarrow [1] \rightarrow []$), из массива `pushed` будет получен массив `popped`.

Формат входных данных

В первой строке одно целое число N , $1 \leq N \leq 100$, – размер массивов `pushed` и `popped`. Во второй строке N неотрицательных целых чисел через пробел – массив `pushed`. В третьей строке N неотрицательных целых чисел через пробел – массив `popped`. Все элементы массивов не превышают 100. Гарантируется, что оба массива содержат одни и те же элементы.

Формат выходных данных

В единственной строке «YES», если `popped` можно получить из `pushed` описанным в задаче способом, и «NO» в противном случае.

Примеры

<i>standard input</i>	<i>standard output</i>
4 1 2 3 4 2 4 3 1	YES
4 1 2 3 4 2 4 1 3	NO

Задача 5. Парад

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 MiB

В городе Энск усиленно готовятся к проведению праздничных мероприятий по случаю очередного Дня Победы и перекрывают улицы (естественно, строго по графику!) для репетиции парада. Однако даже в праздничное время люди могут внезапно заболеть, поэтому необходимо знать, за какое минимальное время от момента вызова скорая помощь сможет добраться от места A до места B .

План города представлен в виде взвешенного ориентированного графа с дугами – улицами. Вес дуги – длина проезжей части улицы в километрах. Если улица имеет не одностороннее движение, то ей соответствуют 2 дуги, веса которых могут быть различны.

Скорая помощь едет с постоянной скоростью 1 км/мин. Скорая не должна быть на улице в то время, когда та перекрыта, но может останавливаться как на открытой улице, так и в вершинах графа, если это необходимо. Скорая может начать движение в любой момент после ее вызова. Места A и B являются вершинами графа.

Формат входных данных

В первой строке пять целых чисел: N , $2 \leq N < 10^5$ – число вершин графа, M , $1 \leq M \leq \min(N \cdot (N - 1), 3 \cdot 10^5)$ – число дуг, R , $0 \leq R < 10^5$ – число перекрытий улиц, A – номер начальной вершины, B – номер конечной вершины (вершины нумеруются с нуля).

Далее M строк по 3 целых числа, которые описывают дугу графа: начальная вершина, конечная вершина и вес w , $1 \leq w \leq 10\,000$.

Далее в R строках по 4 целых неотрицательных числа, которые описывают перекрытия улиц: номер начальной вершины и номер конечной вершины дуги графа, соответствующей перекрываемой улице, время начала перекрытия и время окончания перекрытия в минутах (не более 10^9 , время окончания всегда больше времени начала) относительно момента вызова скорой. Обращаем внимание, что улицы могут перекрываться не один раз! Гарантируется, что интервалы перекрытий не совпадают и что для каждого заданного перекрытия дуга графа существует.

Формат выходных данных

В единственной строке целое число – кратчайшее время в минутах от момента вызова скорой, за которое она доедет из A в B .

Примеры

<i>standard input</i>	<i>standard output</i>
3 3 1 0 2 0 1 3 0 2 5 1 2 4 0 2 2 3	7

Задача 6. Супер-игра

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 MiB

Вы выиграли денежный приз. Точную сумму выигрыша N выложили перед Вами из карточек, на каждой из которых написана ровно одна цифра. Теперь Вам предлагают сыграть в супер-игру, суть которой заключается в следующем: ровно K раз Вы меняете местами любые две соседние карточки, и какое число получится в конце – такую сумму и получите. Определите максимальный выигрыш, который Вы можете получить в этой супер-игре. Вдруг и начинать не стоит.

Формат входных данных

В первой строке одно целое число K , $1 \leq K < 10^9$, – количество перестановок двух соседних карточек, которое необходимо осуществить. Во второй строке одно целое число N , $10 \leq N \leq 10^{50000}$, без ведущих нулей – исходный денежный приз.

Формат выходных данных

В первой и единственной строке максимальный выигрыш, который можно получить, переставив соседние карточки с цифрами числа N ровно K раз. Если в полученном числе есть ведущие нули, их необходимо вывести.

Примеры

<i>standard input</i>	<i>standard output</i>
5 123456789	612345789

Задача 7. Все перекаты да перекаты

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 MiB

Рассмотрим игральный кубик, стоящий на бесконечном поле, на гранях которого написаны некоторые целые положительные числа, возможно, повторяющиеся. Будем перемещать этот кубик по полю, перекатывая через ребра в его основании, при этом не будем возвращаться на тот участок поля, где кубик стоял ранее. В ходе перемещений будем суммировать значения, попадающие на верхнюю грань кубика. Определите, какую минимальную сумму мы можем получить в результате. Числа, находящиеся на верхней грани в начальном и конечном положениях кубика, также участвуют в сумме.

Формат входных данных

В первой и единственной строке 7 целых положительных чисел.

Первые шесть из них – значения на гранях кубика в следующем порядке их перечисления: верхняя, нижняя, левая, правая, передняя, задняя. Каждое из значений не превышает 1 000.

Последнее, седьмое число K – количество перекатываний кубика, $1 \leq K < 18$.

Формат выходных данных

В первой и единственной строке одно целое неотрицательное число – минимальная сумма значений, попавших на верхнюю грань кубика в процессе перекатываний, включая исходное и конечное.

Примеры

<i>standard input</i>	<i>standard output</i>
1 2 3 4 5 6 2	6

Задача 8. Тяни-толкай

Имя входного файла:	<i>standard input</i>
Имя выходного файла:	<i>standard output</i>
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 MiB

План лабиринта представляет собой клеточное поле размером $N \times M$. Стены лабиринта на плане обозначены символом '1'. В свободных от стен клетках лабиринта отмечены позиции кладоискателя ('*') и сундука с сокровищами ('+'), а также позиция, в которую кладоискатель должен переместить сундук ('x'). Остальные свободные клетки отмечены символом '0'. Кладоискатель перемещается по свободным клеткам лабиринта, включая ту, в которую он должен поместить сундук, но не может находиться с сундуком в одной клетке.

При перемещении кладоискатель переходит из одной клетки лабиринта в другую, граничащую с исходной по стороне. При этом, если надо, он может передвигать сундук следующим образом:

1. Чтобы толкнуть сундук, кладоискатель должен стать в соседнюю с ним клетку и сделать шаг в клетку с сундуком. В результате кладоискатель окажется в клетке, где был сундук, а сундук передвинется на соседнюю клетку в том же направлении. Разумеется, толкать сундук можно только на свободные клетки.
2. Чтобы потянуть сундук, кладоискатель должен стать в соседнюю с ним клетку и сделать шаг в направлении от сундука. В результате сундук переместится в клетку, где был кладоискатель, а кладоискатель перейдет на соседнюю клетку в том же направлении. Разумеется, клетка, в которую переходит кладоискатель, должна быть свободна. Делая шаг в направлении от сундука, кладоискатель не обязан тащить его за собой.

На то, чтобы толкнуть сундук, кладоискатель тратит A энергии. На то, чтобы потянуть сундук, кладоискатель тратит B энергии. Без сундука кладоискатель перемещается очень легко и потому энергию не тратит.

Определите, какое минимальное количество энергии потратит кладоискатель, чтобы переместить сундук в заданную точку.

Формат входных данных

В первой строке четыре целых неотрицательных числа через пробел: N , M – размеры лабиринта, A и B – количество энергии, необходимое, чтобы сдвинуть сундук на одну клетку от себя или к себе, соответственно, $1 \leq N, M \leq 50$, $1 \leq A, B \leq 50$.

Далее N строк, в каждой из которых M символов – план лабиринта, в котором используются следующие обозначения: символ '0' – свободное пространство, символ '1' – стена, символ '*' – позиция кладоискателя, символ '+' – позиция сундука с сокровищами, символ 'x' – позиция, в которую надо поставить сундук.

Других символов на плане нет, при этом гарантируется, что символы '+', '*', 'x' встречаются ровно один раз.

Формат выходных данных

В первой и единственной строке одно целое положительное число – минимальное количество энергии, необходимое для перемещения сундука в нужное место. В случае, если такое перемещение невозможно, вывести -1 .

Примеры

<i>standard input</i>	<i>standard output</i>
4 5 1 2 *0000 01000 +0110 1000x	12

Задача 9. Калькулятор

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 MiB

У вас есть простой калькулятор, у которого есть камера, экран для отображения числа и две кнопки. Камера используется для ввода числа – наводим камеру на число, и оно отображается на экране. Первая кнопка прибавляет к числу на экране число X , а вторая кнопка умножает его на Y . Зная число A , которое ввели, и видя число B на экране, определите минимальное количество нажатий кнопок, в результате которого из числа A можно получить число B .

Формат входных данных

В первой строке четыре целых числа A, B, X, Y , $0 \leq A \leq B \leq 10^7$, $1 \leq X \leq 10^7$, $1 < Y \leq 10^7$.

Формат выходных данных

В первой и единственной строке одно целое число – минимальное количество операций прибавления X и умножения на Y , за которое число A можно преобразовать в число B . Если калькулятор сломался и такое преобразование невозможно, вывести -1 .

Примеры

<i>standard input</i>	<i>standard output</i>
0 20 1 2	6
0 20 3 2	-1

Задача 10. Лазерная комната

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 MiB

В парке развлечений собирают новый аттракцион – “Лазерная комната”.

Внутри прямоугольной комнаты размером $A \times B$ установили N двусторонних плоских зеркал под углом 45° к ее стенам. Попадая в такое зеркало, луч лазера, направленный параллельно стене комнаты, очевидно, отражается под прямым углом.

Когда же понадобилось расставлять лазерные пушки по периметру, возник вопрос, как сделать аттракцион привлекательнее. Было решено, что хотя бы одну пушку надо установить так, чтобы длина запущенного из нее луча была максимальна. Отметим, что конструкция пушки позволяла запускать луч только перпендикулярно стене, к которой ее крепили.

Определите максимально возможную длину луча, выпущенного из пушки, до его попадания в стену комнаты.

Формат входных данных

В первой строке три целых числа через пробел: A, B – размеры комнаты по осям X и Y , соответственно, $1 \leq A, B \leq 2 \cdot 10^9$, и N – количество зеркал, расставленных внутри комнаты, $0 \leq N \leq \min((A - 2) \cdot (B - 2), 10\,000)$

Далее N строк по 3 целых числа в каждой через пробел: x_i, y_i – координаты i -го зеркала по осям X и Y , z_i – угол этого зеркала к оси X , $0 < x_i < A$, $0 < y_i < B$, z_i – либо 45 , либо -45 . Гарантируется, что координаты любых двух зеркал не совпадают.

Считается, что ось X направлена слева направо, а ось Y – снизу вверх.

Формат выходных данных

В первой единственной строке одно целое положительное число – максимально возможная длина луча лазера, запущенного из пушки, установленной на одной из стен комнаты, перпендикулярно этой стене. Шириной зеркал следует пренебречь, то есть считать зеркала бесконечно узкими.

Примеры

<i>standard input</i>	<i>standard output</i>
5 3 2 4 1 45 4 2 -45	9

Задача 11. Деление на три

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 MiB

На полоске бумаги записано некоторое число N . Вам предложено сделать два разреза между цифрами этого числа так, чтобы на трех полученных полосках бумаги получились числа без незначащих нулей. Определите минимально возможное значение суммы этих чисел.

Формат входных данных

В первой и единственной строке одно целое число N , $100 \leq N \leq 10^{1000000}$.

Формат выходных данных

В первой и единственной строке одно целое положительное число – минимально возможная сумма трех чисел, на которые можно разрезать исходное число N так, что ни одно из этих чисел не будет содержать незначащие нули.

Примеры

<i>standard input</i>	<i>standard output</i>
1020304	334

Задача 12. Черное и белое

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 MiB

Вы купили набор, состоящий из черных и белых кубиков со стороной 1. Известно, что, все кубики набора можно сложить в большой куб со стороной N . Зная количество белых кубиков, определите, какое наибольшее количество единичных квадратов может при этом оказаться на поверхности.

Формат входных данных

В первой строке и единственной строке два целых числа через пробел: N – длина стороны большого куба, $1 \leq N \leq 1\,000$, и K – количество белых кубиков, $0 \leq K \leq N^3$.

Формат выходных данных

В первой и единственной строке одно целое число – максимально возможное количество белых единичных квадратов на поверхности куба со стороной N , составленного из имеющихся K белых и $(N^3 - K)$ черных кубиков.

Примеры

<i>standard input</i>	<i>standard output</i>
3 10	28