

Задача А. Общая подстрока

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

Заданы две строки, состоящие из 0 и 1. Рассмотрим все строки, которые являются подстроками обеих данных строк. Найдите среди них k -ую в лексикографическом порядке.

Строка S меньше строки T в лексикографическом порядке, если выполняется одно из двух условий:

- S является префиксом T ;
- существует i , не превышающее длин строк S и T , такое что для $j < i$ выполняется $S[j] = T[j]$ и $S[i] < T[i]$.

Формат входного файла

Первые две строки входного файла содержат заданные строки, длиной не более 4000 символов каждая. Третья строка содержит целое положительное число k , не превышающее количества общих подстрок двух заданных строк.

Формат выходного файла

Выведите в выходной файл k -ую в лексикографическом порядке общую подстроку заданных строк.

Пример

<code>common.in</code>	<code>common.out</code>
0100 0010 3	01

Задача В. Расстояние между многоугольниками

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

Нужно найти расстояние между двумя выпуклыми непересекающимися многоугольниками.

Формат входного файла

Во входном файле содержатся описания двух многоугольников.

Многоугольник задается числом вершин — N ($1 \leq N \leq 50\,000$). И координатами N вершин. Вершины даны в порядке обхода по часовой стрелке. Координаты целые и не превосходят 10^9 по очереди.

В обоих многоугольниках никакие три точки не лежат на одной прямой.

Формат выходного файла

Выведите одно вещественное число — расстояние между многоугольниками. Выводите ответ с максимально возможной точностью. Ваше решение будет считаться верным, если относительная или абсолютная погрешность ответа не превосходит 10^{-12} .

Пример

<code>distance.in</code>	<code>distance.out</code>
4 0 0 0 1 1 1 1 0 3 2 0 2 2 4 0	1.00000000000000000000

Задача С. Кладбищенский сторож

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

В одном чёрном-чёрном лесу, на чёрном-чёрном кладбище стояло надгробие из чистого золота. Это надгробие охраняли две страшных, чёрных-пречёрных собаки. Каждая собака сидит на цепи около чёрного-чёрного столба, а неподалеку, в том же лесу, стоит еще и чёрный-чёрный домик сторожа. Каждое утро сторож выходит из домика и несёт собакам миски с едой. Он ставит миски на землю так, чтобы собаки могли есть, оставаясь на привязи у своих столбов. Вычислите кратчайшее расстояние, которое необходимо пройти сторожу, чтобы накормить обеих собак. (Сторож легко может нести еду сразу для двух собак и может кормить их в любом порядке).

Формат входного файла

В первой строке записаны три числа: расстояние в метрах от домика сторожа до первого столба R_1 , от домика сторожа до второго столба R_2 , и расстояние между столбами R_3 . Во второй строке одно целое число — длина цепи каждой из собак в метрах R_4 (цепи у собак одинаковые). Все числа R_i ($i = 1, 2, 3$) удовлетворяют ограничению $0 \leq R_i \leq 10000$; $1 \leq R_4 \leq 10000$.

Формат выходного файла

Выведите одно число — длину кратчайшего пути сторожа в метрах с точностью 3 знака после запятой.

Пример

guard.in	guard.out
1000 2000 1000 250	3500.000

Задача D. Котята с пирожками

Имя входного файла: kittens.in
Имя выходного файла: kittens.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

Однажды n котят решили покушать пирожков. Однако котят много, поэтому им непросто выбрать начинку, которая всех порадует. Известно, что группа из не более чем k котят всегда может прийти к консенсусу, а вот большая группа обязательно разобьётся на две, принципиально несогласные друг с другом. Котята — существа справедливые, поэтому размеры этих групп будут отличаться не более чем на один.

Группы, размер которых всё ещё окажется больше k после такого разделения, продолжат спорить и разделяться на меньшие по тому же принципу. Определите, сколько групп котят в итоге отправятся за пирожками.

Формат входного файла

В единственной строке входного файла записаны целые числа n и k — количество котят и критический размер группы, соответственно ($1 \leq n, k \leq 10^{18}$). Числа записаны без ведущих нулей.

Формат выходного файла

Выведите единственное целое число — итоговое количество групп. Число должно быть также записано без ведущих нулей.

Пример

kittens.in	kittens.out
17 4	5

Задача E. Робот

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

Дано поле $[-N..N] \times [-N..N]$. В начальный момент времени робот стоит в точке $(0, 0)$ и направлен на восток в сторону точки $(1, 0)$.

Робот перемещается в соответствии со своей программой. Программа — это корректное логическое выражение, содержащее операции NOT, AND, OR (NOT имеет наибольший приоритет, OR наименьший), скобки, константы 'TRUE' и 'FALSE' и регистры (переменные) 'A', ..., 'Z'.

Изначально, все регистры робота имеют значение FALSE. Робот движется по прямой, пока не достигнет развилки. После этого робот считает значение выражения и поворачивает направо, если оно истинно, и налево, в противном случае. Однако при попадании в некоторые точки поля один из регистров робота меняет свое значение на противоположное.

Вас попросили найти маршрут робота до момента, когда он выйдет за пределы поля.

Формат входного файла

В первой строке файла содержится логическое выражение. Длина выражения не превышает 250 символов.

Вторая строка содержит три целых числа N, M, K ($1 \leq N \leq 100, 1 \leq M \leq 100, 1 \leq K \leq 100$), где N — размер поля, M — количество развилок, K — количество точек перемены регистров.

В следующих M строках содержится два целых числа X, Y — координаты развилок.

Каждая из следующих K строк содержит два целых числа X, Y и символ C — координаты точки перемены регистра и название регистра, значение которого инвертируется.

Вы можете предполагать, что в точке $(0, 0)$ нет развилки, что в каждой точке поля находится не более одного объекта (развилки или точки перемены регистра). Также гарантируется, что робот в какой-то момент выйдет за пределы поля.

Формат выходного файла

Выведите в выходной файл путь робота — каждая пара координат в отдельной строке.

Пример

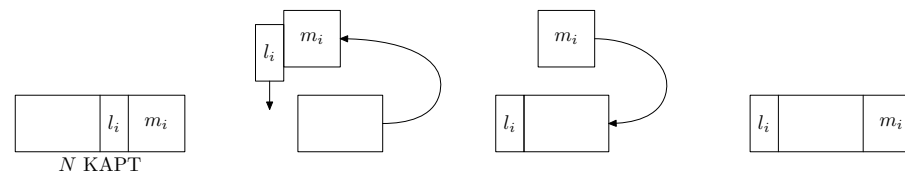
robot.in	robot.out
NOT((A OR NOT B) AND (A OR B)) OR NOT (A AND NOT B OR TRUE)	0 0
1 5 2	1 0
1 0	1 -1
1 1	0 -1
1 -1	-1 -1
-1 -1	-1 0
-1 1	-1 1
0 1 A	0 1
-1 0 D	1 1

Задача F. Карточный фокус

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

Многие карточные фокусы основаны на перемешивании колоды. Перемешивая карты особым образом, фокусник может добиться желаемого порядка карт. В этой задаче вам необходимо промоделировать этот процесс, чтобы убедиться в том, что фокусник почти всегда может добиться своей цели.

Колода содержит n карт. Перемешивание состоит ровно из t ходов. Есть p различных способов сделать каждый ход. i -й способ сделать ход состоит в следующем: тасующий берет $m_i + l_i$ карт сверху колоды и перекладывает их под низ. Снизу он пропускает l_i переложённых карт, а оставшиеся m_i возвращает наверх колоды. Взаимное расположение карт внутри блоков из l_i и m_i карт, а также всех остальных карт колоды, остается неизменным.



По заданным начальному порядку карт, желаемому порядку и параметрам перемешивания требуется определить, возможно ли перетасовать колоду указанным образом.

Формат входного файла

Первая строка входного файла содержит целые числа n , t и p ($2 \leq n \leq 36$, $1 \leq t \leq 15$, $1 \leq p \leq 5$). Следующие две строки описывают начальный и желаемый порядок карт соответственно. Карты занумерованы целыми числами от 1 до n . Все карты различны. Карты перечислены, начиная снизу колоды.

Последующие p строк содержат каждая по два целых числа — m_i и l_i ($1 \leq l_i < n$, $0 \leq m_i < n$, $l_i + m_i < n$).

Формат выходного файла

Если переупорядочить карты указанным образом, используя ровно t ходов, невозможно выведите “Impossible” на первой строке выходного файла. В противном случае выведите t целых чисел — номера ходов.

Пример

shuffle.in	shuffle.out
6 3 2 1 2 3 4 5 6 4 5 1 2 3 6 1 2 1 3	1 1 2
6 2 2 1 2 3 4 5 6 4 5 1 2 3 6 1 2 1 3	Impossible

Задача G. Кривые зеркала

Имя входного файла: `mirror.in`
Имя выходного файла: `mirror.out`
Ограничение по времени: 0.5 секунды
Ограничение по памяти: 64 Мб

За один залп BFG-9000 уничтожает три соседних балкончика (N -й балкончик соседствует с первым). После залпа оставшиеся в живых монстры наносят Леониду (главному герою романа) повреждения — по одной единице каждый. Далее следует новый залп и так до тех пор, пока все монстры не погибнут. Требуется определить минимальные повреждения, которые может понести Леонид.

Формат входного файла

Первая строка содержит целое число N , количество балкончиков, на которых монстры заняли круговую оборону ($3 \leq N \leq 30$). Во второй строке даны N целых чисел — количество монстров на каждом балкончике (на каждом не менее 1 и не более 100).

Формат выходного файла

Выведите минимальное количество единиц повреждений.

Пример

<code>mirror.in</code>	<code>mirror.out</code>
7 3 4 2 2 1 4 1	9