

Задача А. Найдите точку

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

Вася — управляющий небольшой нефтяной компании. Недавно геологами из этой компании было обнаружено два новых месторождения. В кратчайшие сроки были построены две нефтедобывающие вышки. Последнее, что осталось сделать — соединить их с магистральным нефтепроводом.

Магистральный нефтепровод представляет собой абсолютно прямую линию, в некоторых местах которой расположены узлы. Нефтепроводы от вышек можно подсоединять только в этих местах. Известно, что узлов бесконечно много, и расстояния между соседними узлами равны.

В связи с некоторыми обстоятельствами, которые Вася разглашать права не имеет, ему нужно подсоединить обе вышки к одному и тому же узлу. Тратить лишние деньги Вася не намерен, поэтому ему хочется выбрать такой узел, чтобы суммарное расстояние от него до каждой из вышек было минимально возможным.

Формат входного файла

В первой строке входного файла заданы координаты первой вышки — два целых числа x и y , разделенные пробелом. Во второй строке аналогичным образом заданы координаты второй вышки. Координаты первой и второй вышки не совпадают. В третьей и четвертой строке заданы координаты двух соседних узлов на магистральном нефтепроводе. Все числа во входном файле не превосходят 10^9 по абсолютному значению.

Формат выходного файла

В выходной файл выведите два целых числа — координаты искомого узла.

Примеры

<code>point.in</code>	<code>point.out</code>
0 10 0 -10 5 0 10 0	0 0
0 10 0 -10 10 0 10 -1	10 0

Задача В. Квадропалиндромы

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

Вася — большой любитель палиндромов. Он пытается искать их везде: в словах, в названиях лимонада, в номере своего паспорта и в кличке своей кошки. Но недавно Василий столкнулся с неразрешимой для него задачей — найти палиндром в матрице.

Напомним, что палиндромом называется последовательность символов которая одинаково читается в обоих направлениях. Матрицей в данном случае будем называть прямоугольную таблицу A размером $n \times m$, заполненную строчными буквами латинского алфавита.

Так вот, Вася долго думал, как же искать в матрице палиндром, и решил, что палиндромом в матрице будет называться такая последовательность пар чисел $(x_i; y_i)$ длины k , что

- $x_i \leq x_j$ для любых i, j таких, что $i < j$;
- $y_i \leq y_j$ для любых i, j таких, что $i < j$;
- $x_i < x_j$ или $y_i < y_j$ для любых i, j таких, что $i < j$;
- последовательность A_{x_i, y_i} является палиндромом.

Вася решил назвать такой палиндром *квадропалиндромом*. Теперь Василий просит вас найти в данной таблице квадропалиндром максимальной длины.

Формат входного файла

В первой строке два целых числа n и m ($1 \leq n, m \leq 50$). Далее следуют n строк длины m — таблица A .

Формат выходного файла

В первой строке выведете число k — максимальную длину квадропалиндрома. Далее должны следовать k строк — последовательность $(x_i; y_i)$.

Примеры

quadro.in	quadro.out
3 3 aba cab ada	5 1 1 1 2 2 2 2 3 3 3
2 15 thisproblemisto ohardtobesolved	5 2 5 2 9 2 13 2 14 2 15

Задача С. Разворот префиксов

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

В Лаборатории Интеллектуальных Префиксных Алгоритмов (ЛИПА) тестируют Машину, Префиксов Разворачивающую (МПР). Машина работает следующим образом: на вход ей подается строка s длины n и последовательность $1 \leq a_1 < a_2 < \dots < a_k \leq n$. Исходно строка s помещается в специальный внутренний регистр машины. После этого для каждого i от 1 до k машина берет префикс $[1..a_i]$ текущей строки и разворачивает его. Строка, которая оказывается в регистре после окончания работы машины представляет собой результат работы машины.

Например, если на вход машине подать строку $s = \text{“abacaba”}$ и последовательность $a_1 = 2, a_2 = 4$, на выходе получится строка “caababa” .

Ученые ЛИПА хотят найти для заданной строки s такую последовательность, чтобы результат работы оказался как можно меньше в лексикографическом порядке. Строка $\alpha = \alpha_1\alpha_2\dots\alpha_n$ *лексикографически меньше* строки $\beta = \beta_1\beta_2\dots\beta_m$, если для некоторого k и для всех $1 \leq t \leq k$ верно $\alpha_t = \beta_t$ и либо $\alpha_{k+1} < \beta_{k+1}$, либо длина α равна k , а длина β больше k .

Помогите им выяснить, какой минимальный лексикографически результат можно получить.

Формат входного файла

Входной файл содержит строку s (она непуста и ее длина не превышает 500 000). Она содержит из строчных букв латинского алфавита.

Формат выходного файла

Выведите минимальную в лексикографическом порядке строку, которая может быть выведена МПР на строке s в качестве входа.

Пример

<code>prefixes.in</code>	<code>prefixes.out</code>
abacaba	aaaabcb

Задача D. Теория графов

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

Сережа изучает теорию графов. Недавно он изучил понятия радиус и диаметр графа. Рассмотрим неориентированный невзвешенный граф G , обозначим длину кратчайшего пути между вершинами s и t как $\rho(s, t)$. Радиусом $r(G)$ графа называется $\min_u \max_v \rho(u, v)$. Диаметр $d(G)$ графа называется $\max_u \max_v \rho(u, v)$.

Интуитивно, можно представить эти понятия следующим образом. Пусть вы играете в некоторую игру, и ваша цель — минимизировать длину пути, а цель вашего противника — максимизировать. Если противник выбирает две вершины, между которыми надо найти кратчайший путь, то диаметр графа, это максимальный ответ, который может получиться. Радиус — максимальный ответ, который может получиться, если вы выбираете первую вершину, а затем противник — вторую.

Профессор доказал на лекции, что $d(G)/2 \leq r(G) \leq d(G)$ для любого графа G . Теперь Сережу заинтересовал вопрос: а для любых ли d и r , таких что $d/2 \leq r \leq d$, можно построить граф G , для которого $d(G) = d$ и $r(G) = r$. Помогите ему разобраться с этим вопросом.

Формат входного файла

Входной файл содержит два целых числа d и r ($d/2 \leq r \leq d \leq 50, 1 \leq r$).

Формат выходного файла

Если граф с такими радиусом и диаметром существует, выведите “YES” на первой строке выходного файла. Вторая строка должна содержать два целых числа: n и m — количество вершин и количество ребер графа ($2 \leq n \leq 400$). Следующие m строк должны содержать по два целых числа — вершины, соединенные соответствующим ребром. В графе не должно быть петель и параллельных ребер.

Если искомого графа не существует, выведите “NO” на первой строке выходного файла.

Примеры

<code>graph.in</code>	<code>graph.out</code>
1 1	YES 2 1 1 2
3 2	YES 7 9 1 2 2 3 3 4 1 4 4 5 5 6 1 6 6 7 7 2