

Задача А. Вовочка и шифр

Имя входного файла: `cypher.in`
Имя выходного файла: `cypher.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В городе П есть исследовательский институт. В этом институте работает Владимир (в прошлом — Вовочка). В последнее время его коллеги заметили, что он остается после работы допоздна, и это плохо сказывается на результатах его работы. Его коллеги решили узнать, чем же занимается Владимир. После тщательного просмотра журналов системы, они обнаружили, что Владимир отправляет зашифрованные сообщения. Оказалось, что шифр был довольно простой — каждый символ алфавита заменяется каким-либо другим символом. При этом одинаковые символы имеют один и тот же шифр. Помогите им разгадать шифр.

Формат входного файла

Входной файл содержит две строки одинаковой длины K , ($1 \leq K \leq 255$). В первой строке содержится сообщение Владимира, а второй — шифр этой строки. Строки состоят из заглавных букв латинского алфавита. Поскольку система шифрования Вовочки может быть несовершенной, нескольким буквам может соответствовать одна и та же буква.

Формат выходного файла

Если входные данные противоречивы, выведите "NO SOLUTION" (без кавычек). В противном случае выведите одну строчку из 26 символов, первый элемент которой — символ, который соответствует букве *A*, второй — букве *B* и так далее. Если невозможно определить шифр какого-либо символа, выведите в соответствующей позиции символ "?".

Примеры

<code>cypher.in</code>	<code>cypher.out</code>
AA AB	NO SOLUTION
VOVOCHKA ABABABAB	B?A????B??A???B?????A????

Задача В. Васин сон

Имя входного файла: `dream.in`
Имя выходного файла: `dream.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Однажды Васе приснился очень странный сон. Он был шахматным королем, причем он был не на обычной шахматной доске, а на доске размером 3×3 . Вася не помнит почему, но он должен был обойти всю доску, побывав в каждой клетке ровно один раз и в конце оказаться в начальной клетке. Вася мог сам выбрать клетку, с которой начинал обход, но при этом не мог ходить по диагонали, а только на четыре соседние клетки. Он долго блуждал по доске, но так и не смог ее обойти. Проснувшись, Вася заинтересовался, для каких m и n таким образом можно обойти доску $m \times n$? Он так и не нашел решение этой задачи. Сможете ли Вы решить ее?

Формат входного файла

Во входном файле заданы 2 натуральных числа, разделенных пробелом: m и n - размеры доски. ($1 \leq n \leq 50, 1 \leq m \leq 50, mn > 1$).

Формат выходного файла

Если решения Васиной задачи для доски $m \times n$ нет, единственная строка выходного файла должна содержать "NO", иначе в первой строке выходного файла выведите "YES" (без кавычек). Далее для каждой клетки доски выведите ее номер в обходе (нумерация начинается с единицы). Таким образом, после "YES" должны следовать m строк по n чисел в каждой, разделенных пробелом. Каждое число от 1 до $n \times m$ должно встретиться среди них ровно один раз, и для любых двух соседних чисел клетки с этими числами должны быть соседними. Кроме того, соседними должны быть клетки с числами 1 и $n \times m$.

Примеры

dream.in	dream.out
3 3	NO
2 2	YES 2 3 1 4

Задача С. Нумерация

Имя входного файла: `enum.in`
Имя выходного файла: `enum.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В городе П проводится олимпиада по программированию. После тестирования программ жюри обнаружило, что задачу решили несколько участников. Жюри постановило, что победителем будет тот участник, у которого самая короткая программа. Но для этого нужно перенумеровать строки каждой программы, так как жюри сбивается со счета. Нумеруются все строки, кроме пустых. Нумерация производится так, как показано в примере (после двоеточия необходимо вывести 2 дополнительных пробела, все двоеточия должны находиться в одной колонке).

Формат входного файла

Все строки во входном файле не длиннее 200 символов и их количество не превосходит 200.

Формат выходного файла

В выходной файл необходимо вывести программу с пронумерованными строчками. При этом все символы (табуляции, пробелы, переводы строк и т.д.) должны сохраниться. Запрещается добавлять любые другие символы, кроме оговоренных в условии.

Примеры

enum.in	enum.out
<pre>int main() { return 0; return 1; return 2; return 3; return 4; return 5; return 6; }</pre>	<pre>1 : int main() 2 : { 3 : return 0; 4 : return 1; 5 : return 2; 6 : return 3; 7 : return 4; 8 : return 5; 9 : return 6; 10: }</pre>

Примечание: Пробельными символами являются: пробел (ASCII код 32), табуляция (ASCII код 9), перевод строки (ASCII коды 13 и 10).

Задача D. Минимум и максимум

Имя входного файла: `minmax.in`
Имя выходного файла: `minmax.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Имеется набор, состоящий из N целых чисел. Разрешается брать любые два числа из этого набора и заменять их либо на минимальное из этих чисел, либо на максимальное. Соответственно, после каждой такой операции количество элементов в наборе уменьшается на единицу. Можно ли с помощью этих операций сделать так, чтобы в результате осталось одно заданное число S ?

Формат входного файла

В первой строке входного файла находятся два целых числа N ($1 \leq N \leq 10000$) и S ($0 \leq S \leq 1000000$). В следующей строке через пробел написаны N целых чисел A_i ($0 \leq A_i \leq 1000000$).

Формат выходного файла

В выходной файл выведите "Yes" (без кавычек), если можно получить число S . Иначе выведите "No".

Примеры

<code>minmax.in</code>	<code>minmax.out</code>
11 5 1 10 3 0 2 5 6 8 7 9 4	Yes

Задача Е. Телефонные номера

Имя входного файла: `phone.in`
Имя выходного файла: `phone.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

У Васи очень много знакомых. Номера их телефонов он записывает в память своего супертелефона `iPhone`. Однако, свободная память телефона достаточно быстро заканчивается, поэтому Вася придумал способ сокращать телефонные номера. Как известно, номер состоит из цифр, причем некоторые цифры и даже последовательности цифр могут повторяться. Чтобы сократить повторяющуюся последовательность, Вася записывает ее один раз, а после нее в скобках пишет, сколько раз она повторилась. Например, номер `26262626` может быть записан в виде `26(4)` или `26(2)26(2)`. Причем, если количество повторений равно единице, то скобки и единицу Вася может не писать. Поскольку количество повторений не может быть нулем или начинаться с нуля, записи `26(0)` и `26(04)` некорректны. Кроме того, Вася не всегда учитывает все повторения последовательности, то есть номер `26262626` может быть записан в виде `26(3)26`. Также Вася не хочет усложнять процесс распаковки телефонных номеров, поэтому вложенных скобок никогда не делает, то есть конструкция вида `26(2)(2)` недопустима.

После вторжения в Васин супертелефон нового супервируса, в Васиных записях пропали все скобки. Теперь Вася хочет узнать, можно ли по оставшейся информации восстановить исходный номер, если известно, что телефонные номера в Васином городе содержат ровно n цифр.

Формат входного файла

В первой строке задано число n — длина номеров в Васином городе ($1 \leq n \leq 100$). Во второй строке задана последовательность цифр без пробелов длиной не более 50 символов — запись в Васином телефоне.

Формат выходного файла

Выведите телефонный номер, если по приведенной записи его можно однозначно восстановить. Если номер восстановить нельзя, выведите `"Impossible"`. Если существует несколько номеров, которые можно сжать до записи в Васином телефоне, выведите `"Ambiguous"`.

Примеры

<code>phone.in</code>	<code>phone.out</code>
5 740	77770
6 0740	077770

Примечание: в первом примере после восстановления скобок запись выглядит так: `7(4)0`.

Задача F. Дороги

Имя входного файла: `roads.in`
Имя выходного файла: `roads.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В одном далеком-далеком государстве царь решил создать развитое государство, построив каменные дороги, которых раньше у него не было. Царство считается развитым, если в нем не менее m каменных дорог. Царь решил не строить лишние дороги, потому что казенные деньги нужно беречь. При этом он хочет, чтобы из каждого города можно было проехать в любой другой, используя только каменные дороги. В царстве n городов. Между двумя городами нельзя строить несколько каменных дорог. Нельзя также строить дороги, ведущие из города в этот же город. Вас, как главного советника царя, попросили составить план постройки дорог, чтобы выполнялись все условия, поставленные царем.

Формат входного файла

Входной файл содержит 2 целых неотрицательных числа n и m - количество городов в царстве и количество дорог, которое необходимо построить ($n \geq 1$). Оба числа не превосходят 10^5 .

Формат выходного файла

В первой строке выходного файла выведите "NO", если построить дороги, согласно требованиям царя невозможно. В противном случае в первой строке выведите "YES" (без кавычек). Далее выведите m строк с описанием возможного расположения дорог. В каждой строке выведите 2 числа - номера городов, между которыми нужно построить дорогу. Если существует несколько решений, выведите любое.

Примеры

roads.in	roads.out
4 4	YES 2 1 1 3 4 3 1 4
10 2	NO

Задача G. Цифры

Имя входного файла: `digits.in`
Имя выходного файла: `digits.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Программист Вася заскучал на уроке математики в школе и решил немного развлечься. Он взял некоторое натуральное число N и начал в нем переставлять цифры, таким образом, чтобы получить минимально возможное число, в котором используются все цифры из N . Помогите Васе найти это число.

Формат входного файла

В файле на первой строке находится единственное число N . Количество цифр в числе не превышает 1000.

Формат выходного файла

Выведите минимальное число, которое можно получить перестановкой цифр в числе N .

Примеры

<code>digits.in</code>	<code>digits.out</code>
3021	123

Примечание: если в начале числа присутствуют нули, их выводить не нужно.

Задача Н. Конфеты

Имя входного файла: `sum.in`
Имя выходного файла: `sum.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В городе П есть магазин конфет. Там продаются конфеты, разложенные в n пакетов. В магазин пришли m друзей. Друзья решили купить 2 пакета конфет. Но конфет должно быть столько, чтобы всем друзьям досталось поровну, и друзья не поссорились.

Формат входного файла

В первой строчке находится число n - количество пакетов с конфетами, и m - количество друзей ($1 \leq m \leq 10^5, 2 \leq n \leq 10^5$). Во второй строке находится n чисел от 0 до 10^9 , которые указывают, сколько конфет в каждом пакете.

Формат выходного файла

Выведите "Yes", если возможно купить два пакета конфет, чтобы друзья не поссорились, иначе "No".

Примеры

sum.in	sum.out
3 4 1 3 4	Yes
2 4 2 3	No

Задача I. Два дерева

Имя входного файла: 2tree.in
Имя выходного файла: 2tree.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Жюри предлагает участникам олимпиады познакомиться с двоичными деревьями, которые очень широко применяются в программировании.

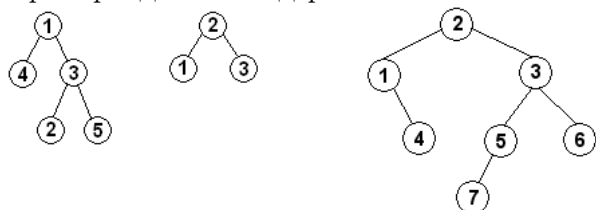
Двоичным деревом называется набор вершин, организованных в виде дерева. Каждая вершина имеет не более двух детей, один из которых называется левым, а другой - правым. Как левый, так и правый ребенок, а также оба могут отсутствовать.

Если вершина Y — ребенок вершины X , то говорят, что вершина X является родителем вершины Y . У каждой вершины дерева, кроме одной, есть ровно один родитель. Единственная вершина, не имеющая родителя, называется корнем дерева.

Соединим каждую вершину кроме корня с ее родителем. Заметим, что для каждой вершины существует ровно один путь, ведущий в нее от корня.

Будем говорить, что два дерева имеют одинаковую структуру, если можно так занумеровать вершины второго дерева, чтобы оно стало идентичным первому.

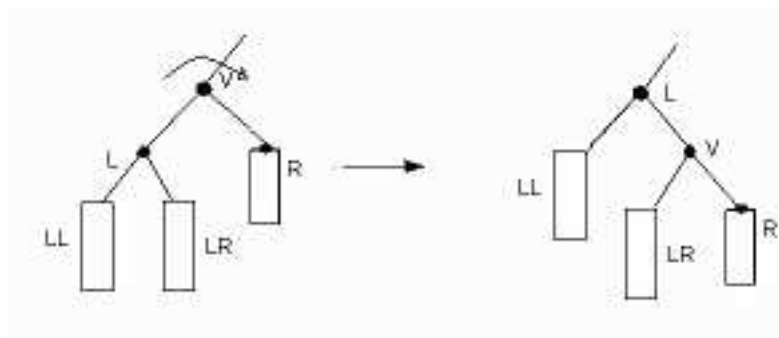
Примеры двоичных деревьев:



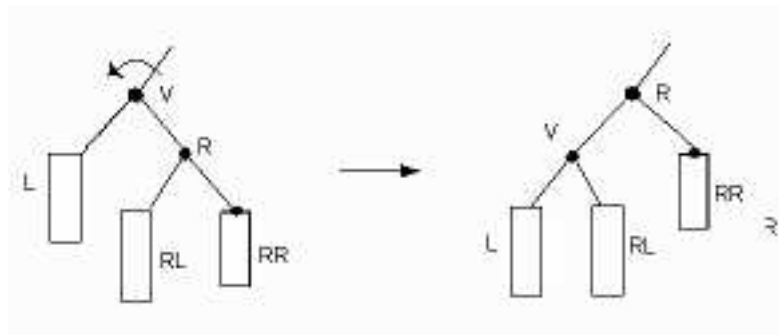
Пусть два двоичных дерева заданы следующим образом: для каждой вершины дерева нам известны номера ее левого и правого детей. Обладая такой информацией, можно однозначно определить корень каждого дерева.

Введем операцию поворота двоичного дерева относительно некоторой вершины.

1. Правый поворот дерева относительно вершины v :

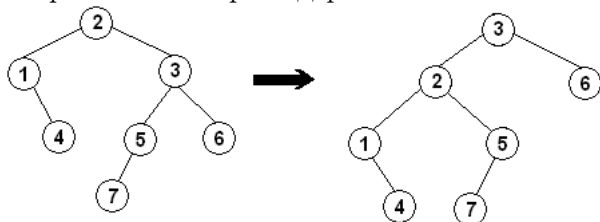


2. Левый поворот дерева относительно вершины v :

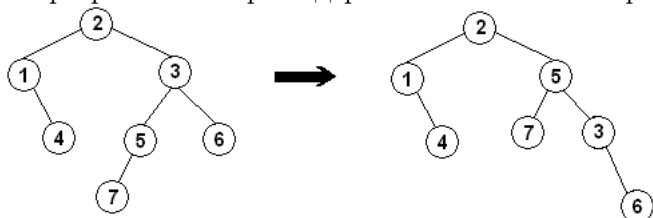


На рисунках прямоугольниками обозначены поддеревья (возможно не содержащие вершин).

Пример левого поворота дерева относительно вершины 2:



Пример правого поворота дерева относительно вершины 3:



С помощью поворотов вершин можно менять структуру дерева. Требуется определить, можно ли из первого дерева получить дерево, структура которого будет такой же, как у второго. Если решение существует, то выведите последовательность поворотов, которые нужно применить к первому дереву.

Формат входного файла

В первой строке входного файла содержится число n — количество вершин у первого и второго деревьев ($3 \leq n \leq 1000$).

Далее следуют $2n$ строк: в первых n строках идет описание первого дерева, во вторых n — второго дерева.

Описание дерева имеет следующий формат: в i строке описания содержится 2 числа — левый и правый сыновья вершины i . Если какого-то ребенка у вершины нет, то вместо его номера указан 0.

Формат выходного файла

Если решение существует, то в первую строку выходного файла выведите число поворотов m , при этом m не должно превышать 50000. Далее выведите m строк следующего вида: номер_вершины_первого_дерева поворот, где поворот — это одна из двух строк: "left" (если левый поворот) или "right" (если правый поворот). В случае, если решения не существует, выведите число -1.

Примеры

2tree.in	2tree.out
3 2 3 0 0 0 0 0 0 1 3 0 0	0
7 0 4 1 3 5 6 0 0 7 0 0 0 0 0 0 4 1 5 2 6 0 0 7 0 0 0 0 0	1 2 left
7 0 4 1 3 5 6 0 0 7 0 0 0 0 0 0 4 1 5 0 6 0 0 7 3 0 0 0 0	1 3 right

Задача J. Количество способов

Имя входного файла: `howmany.in`
Имя выходного файла: `howmany.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася хочет посчитать количество способов выбрать p чисел из последовательности $1, 2, \dots, 2p - 1, 2p$, так, чтобы сумма этих p чисел делилась на p . Совершенно случайно число p оказалось простым. Простыми называются числа, которые делятся только на себя или единицу. Вычислите искомое количество способов, но ни в коем случае не говорите его Васе.

Формат входного файла

Во входном файле содержится одно простое число p ($3 \leq p \leq 50$).

Формат выходного файла

Выведите искомое количество способов.

Примеры

<code>howmany.in</code>	<code>howmany.out</code>
3	8

Примечание: искомые способы для $p = 3$

$\{1, 2, 3\}$, $\{1, 2, 6\}$, $\{1, 3, 5\}$, $\{1, 5, 6\}$,
 $\{2, 3, 4\}$, $\{2, 4, 6\}$, $\{3, 4, 5\}$, $\{4, 5, 6\}$