

Задача А. Число по модулю

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Рассмотрим два числа n и m . Вычислите $r = n \bmod m$.

Формат входных данных

В первой строке заданы через пробел два целых числа n и m ($0 \leq n \leq 10^9$, $1 \leq m \leq 10^9$).

Формат выходных данных

Выведите целое число $r = n \bmod m$.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 3	2

Задача В. Найдите модуль

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Рассмотрим два числа n и m . Вычислим $r = n \bmod m$.

Теперь известны только числа n и r . Восстановите возможное значение m .

Формат входных данных

В первой строке заданы через пробел два целых числа n и r ($0 \leq n \leq 10^9$, $0 \leq r < 10^9$). Гарантируется, что существует целое число m от 1 до 10^9 , для которого $n \bmod m = r$.

Формат выходных данных

Выведите любое целое число m от 1 до 10^9 , для которого $n \bmod m = r$.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 2	3

Задача С. Пути в графе

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Рассмотрим ориентированный граф из n вершин и m дуг. Вершины пронумерованы целыми числами от 1 до n . Каждая дуга (u, v) такова, что $u < v$. В графе не может быть кратных дуг.

Найдите количество различных путей из вершины 1 в вершину n .

Формат входных данных

В первой строке заданы два целых числа n и m ($1 \leq n \leq 100$, $0 \leq m \leq 1000$). Далее следуют m строк с описанием дуг. В каждой из этих строк записаны два целых числа u и v , которые означают, что в графе есть дуга из вершины u в вершину v ($1 \leq u < v \leq n$). Гарантируется, что в графе нет кратных дуг.

Формат выходных данных

Выведите одно целое число: количество различных путей из вершины 1 в вершину n . Гарантируется, что во всех тестах ответ не превосходит 10^9 .

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
9 13 1 2 2 7 1 3 3 7 1 4 4 7 1 5 5 7 1 6 6 7 7 8 7 9 8 9	10

Задача D. Количество путей

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Рассмотрим ориентированный граф из n вершин и m дуг. Вершины пронумерованы целыми числами от 1 до n . Каждая дуга (u, v) такова, что $u < v$. В графе не может быть кратных дуг.

Постройте граф, в котором количество различных путей по дугам из вершины 1 в вершину n равно заданному числу x .

Формат входных данных

В первой строке задано целое число x ($1 \leq x \leq 10^9$).

Формат выходных данных

В первой строке выведите через пробел два целых числа n и m ($1 \leq n \leq 100$, $0 \leq m \leq 1000$). Далее выведите m строк с описанием дуг в любом порядке. Дуга (u, v) задаётся двумя целыми числами u и v через пробел ($1 \leq u < v \leq n$). В графе не может быть кратных дуг. Количество путей из вершины 1 в вершину n должно быть равно заданному числу x .

Если возможных решений несколько, выведите любое из них.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
10	9 13 1 2 2 7 1 3 3 7 1 4 4 7 1 5 5 7 1 6 6 7 7 8 7 9 8 9

Задача Е. Длина пути

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Рассмотрим клетчатое поле $m \times n$. Некоторые клетки свободны, а другие содержат препятствия.

Робот перемещается по полю, каждым шагом переходя на соседнюю по стороне свободную клетку. Робот начинает движение из левой верхней клетки и должен оказаться в правой нижней клетке. Какое минимальное число шагов нужно сделать роботу?

Формат входных данных

В первой строке заданы через пробел два целых числа m и n ($1 \leq m, n \leq 50$). Далее даны m строк, по n символов в каждой. Символ «.» обозначает свободную клетку, а символ «#» — препятствие.

Гарантируется, что левая верхняя и правая нижняя клетки свободны, а также что путь между ними существует.

Формат выходных данных

Выведите одно число: минимальное количество шагов, за которое робот может добраться из левой верхней клетки в правую нижнюю.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 6 .#.... .#.... ...#..	9

Задача F. Длинный путь

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Рассмотрим клетчатое поле $m \times n$. Некоторые клетки свободны, а другие содержат препятствия.

Робот перемещается по полю, каждым шагом переходя на соседнюю по стороне свободную клетку. Постройте поле, на котором кратчайший путь из левой верхней клетки в правую нижнюю существует, но требует не менее $\frac{2}{3}(m-2)(n-2)$ шагов.

Формат входных данных

В первой строке заданы через пробел два целых числа m и n ($2 \leq m, n \leq 50$).

Формат выходных данных

Выведите m строк, по n символов в каждой. Символ «.» обозначает свободную клетку, а символ «#» — препятствие.

Левая верхняя и правая нижняя клетки должны быть свободны. Кроме того, кратчайший путь из левой верхней клетки в правую нижнюю должен существовать, но требовать не менее $\frac{2}{3}(m-2)(n-2)$ шагов.

Если возможных решений несколько, выведите любое из них.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 6	.#.... .#.#.# ...#..

Задача Г. Полиномиальный хеш

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Рассмотрим строку s длины n , состоящую из маленьких английских букв. Зафиксируем числа p и q . Вычислим *полиномиальный хеш* строки s : $h(s) = (s_1p^{n-1} + s_2p^{n-2} + \dots + s_np^0) \bmod q$. Здесь числовые значения s_i — это ASCII-коды символов строки.

Дана строка s , а также целые числа p и q . Вычислите значение $h(s)$.

Формат входных данных

В первой строке заданы три целых числа n , p и q ($1 \leq n \leq 10^5$, $0 \leq p < q \leq 10^6$).

Во второй строке задана строка s . Строка состоит ровно из n маленьких букв английского алфавита.

Формат выходных данных

Выведите одно целое число: значение $h(s)$.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
12 3 100 abcdefghijklz	8

Задача Н. Хеш строки

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Рассмотрим строку s длины n , состоящую из маленьких английских букв. Зафиксируем числа p и q и вычислим её *полиномиальный хеш* $h(s) = (s_1p^{n-1} + s_2p^{n-2} + \dots + s_np^0) \bmod q$. Здесь числовые значения s_i — это ASCII-коды символов строки.

Задан шаблон для строки, в котором не менее десяти символов заменены на «?». Также задано число r . Восстановите буквы, заменённые на «?», так, чтобы выполнялось равенство $h(s) = r$.

Формат входных данных

В первой строке задан шаблон t для строки. Этот шаблон имеет длину не более 10^5 символов и содержит не менее десяти символов «?». Остальные символы строки — маленькие буквы английского алфавита. Во второй строке заданы целые числа p , q и r ($10 \leq q \leq 10^6$, число q простое, $0 \leq p, r < q$). Гарантируется, что все числа $p^i \bmod q$ для $i = 0, 1, \dots, |t| - 1$ различны, а позиции символов «?» выбраны случайно во всех тестах, кроме примера.

Формат выходных данных

В первой строке выведите строку s . Она должна получаться из шаблона t заменой каждого символа «?» на маленькую английскую букву. Кроме того, должно быть выполнено равенство $h(s) = r$.

Если возможных решений несколько, выведите любое из них.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
a?????????z 3 100 8	abcdefghijklz

Задача I. Флойд

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Дан ориентированный взвешенный граф. В нём вам необходимо найти пару вершин, кратчайшее расстояние от одной из которых до другой максимально среди всех пар вершин.

Формат входных данных

В первой строке записано целое число N — количество вершин графа ($1 \leq N \leq 100$). В каждой из следующих N строк задано по N чисел — матрица смежности графа, где -1 означает отсутствие ребра между вершинами, а любое неотрицательное число — присутствие ребра данного веса.

На главной диагонали матрицы всегда стоят нули. Все числа во входных данных не превосходят 100. Гарантируется, что хотя бы одно ребро в графе присутствует.

Формат выходных данных

Выведите искомое максимальное кратчайшее расстояние.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 0 5 9 -1 -1 0 2 8 -1 -1 0 7 4 -1 -1 0	16

Задача J. Порядок циклов

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Дан неориентированный граф из n вершин, заданный матрицей смежности a ($a[u][u] = \text{True}$; $a[u][v] = a[v][u]$; $a[u][v] = \text{True}$ тогда и только тогда, когда есть ребро между вершинами u и v). На нём запускают следующий алгоритм:

```
for x := 1 to n do  
  for y := 1 to n do  
    for z := 1 to n do  
      if a[i][k] and a[k][j] then  
        a[i][j] := True;
```

Перед запуском буквы x , y и z заменяют буквами i , j и k в некотором порядке. Утверждается, что после работы этого алгоритма $a[u][v] = \text{True}$ тогда и только тогда, когда в исходном графе существует путь между вершинами u и v . Выясните, верно ли это, и если нет, приведите пример исходного графа, на котором это неверно.

Формат входных данных

В первой строке ввода записаны через пробел три буквы — «i», «j» и «k» — в некотором порядке. Первая буква подставляется в программу вместо «x», вторая — вместо «y», третья — вместо «z».

Формат выходных данных

Если искомый граф существует, в первой строке выведите через пробел целые числа n и m — количество вершин и рёбер в графе, соответственно ($1 \leq n \leq 10$, $0 \leq m \leq 45$). В следующих m строках выведите пары вершин, соединённых рёбрами, по одной паре на строке. Номера вершин в паре должны быть упорядочены по возрастанию; вершины нумеруются с единицы. Кратные рёбра и петли не допускаются.

Если же программа с заданным порядком циклов корректно работает на любом графе, вместо n и m выведите в первой строке два нуля через пробел.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
k i j	0 0