

Задача А. Вполоборота

Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 мегабайта
 Штраф за неудачную посылку 5 баллов

Словом считается непустая последовательность больших и маленьких латинских букв. Вам необходимо развернуть все слова во входном файле задом наперед.

Формат входных данных

Во входном файле записано все что угодно. Его размер не превышает 1 Мб.

Формат выходных данных

В выходном файле должен содержаться входной файл, в котором все слова развернуты. Все прочие символы, в том числе переводы строк, должны остаться нетронутыми

Пример

Входные данные	Выходные данные
Summer Informatics School. Now in Perm!	remmuS scitamrofni loohcS. woN ni mreP!

Задача В. Дерево?

Максимальное время работы на одном тесте: 2 секунды
 Максимальный объем используемой памяти: 64 мегабайта
 Штраф за неудачную посылку 2 балла

Имеется неориентированный граф, состоящий из N вершин и M ребер. Необходимо проверить, является ли граф деревом. Напомним, что дерево — это связный граф, в котором нет циклов (следовательно, между любой парой вершин существует ровно один простой путь). Граф называется связным, если от одной вершины существует путь до любой другой.

Формат входных данных

Во входном файле в первой строке содержатся два целых числа N и M ($1 \leq N \leq 100$, $0 \leq M \leq 1000$), записанные через пробел. Далее следуют M различных строк с описаниями ребер, каждая из которых содержит два натуральных числа A_i и B_i ($1 \leq A_i < B_i \leq N$), где A_i и B_i — номера вершин, соединенных i -м ребром.

Формат выходных данных

В выходной файл выведите слово YES, если граф является деревом или NO в противном случае.

Пример

Входные данные	Выходные данные
3 2 1 2 1 3	YES

Задача С. Раскраска кубиков

Максимальное время работы на одном тесте: 2 секунды
 Максимальный объем используемой памяти: 64 мегабайта
 Штраф за неудачную посылку 2 балла

На день рождения Пете подарили коробку кубиков. На каждом кубике написано некоторое целое число. Петя выложил все n своих кубиков в ряд, так что числа на кубиках оказались расположены в некотором порядке $a[1], a[2], \dots, a[n]$. Теперь он хочет раскрасить кубики в разные цвета таким образом, чтобы для каждого цвета последовательность чисел на кубиках этого цвета была строго возрастающей. То есть, если кубики с номерами $i[1], i[2], \dots, i[k]$ покрашены в один цвет, то $a[i[1]] < a[i[2]] < \dots < a[i[k]]$. Петя хочет использовать как можно меньше цветов. Помогите ему!

Формат входных данных

Первая строка входного файла содержит число n - количество кубиков у Пети ($1 \leq n \leq 250000$). Затем следует n чисел, разделенных пробелами и/или переводами строки - $a[1], a[2], \dots, a[n]$

Формат выходных данных

На первой строке выходного файла выведите число L - наименьшее количество цветов, которое потребуется Пете. На следующей строке выведите n чисел из диапазона от 1 до L - цвета, в которые Петя должен покрасить кубики.

Пример

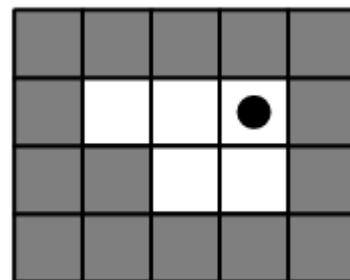
Входные данные	Выходные данные
5 1 2 4 3 5	2 1 1 1 2 2

Задача D. Вслепую по лабиринту

Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 мегабайт
 Штраф за неудачную посылку: 2 балла

Это интерактивная задача.

Ваша цель — написать программу, управляющую роботом, идущим вслепую по лабиринту. Лабиринт состоит из N на M ($1 \leq N, M \leq 30$) клеток. Каждая из клеток может быть свободной или заблокированной (непроходимой). Все клетки на границе лабиринта непроходимые. Робот начинает работу в свободной клетке, он может переместиться на юг, запад, север или восток в свободную клетку. Робот не имеет оптических сенсоров, только сенсор удара, так что при попытке перемещения в заблокированную клетку сработает сенсор и робот останется в той же клетке. Робот должен побывать во всех проходимых клетках лабиринта. Из начальной клетки гарантированно можно попасть во все достижимые клетки лабиринта.



Протокол интерактивного взаимодействия

Программа должна выводить на стандартный вывод одну строку с действием робота и ждать строки в стандартном вводе с ответом, затем выводить очередную строку с действием и считывать ответ и так далее до тех пор, пока все проходимые клетки лабиринта не будут посещены. Программа должна завершать работу только когда все проходимые клетки будут посещены. Проходимые клетки могут быть посещены несколько раз. Допустимо передвигаться даже если все проходимые клетки уже посещены.

Формат выходных данных

Каждая строка выходных данных должна представлять собой команду для робота. Это должна быть одна из пяти возможных строк: SOUTH, WEST, NORTH, EAST, или DONE. Строка DONE должна быть напечатана после посещения роботом всех проходимых клеток. После вывода DONE программа должна завершать свою работу. Необходимо очищать выходной буфер после вывода каждой команды (flush).

Формат входных данных

Каждая строка входных данных представляет собой ответ на действие робота. Это может быть строка EMPTY если робот успешно переместился в заданном направлении или строка BLOCKED если робот не смог переместиться из-за того, что клетка, в которую он хотел попасть, непроходима.

Пример

Входные данные	Выходные данные
NORTH	BLOCKED
EAST	BLOCKED
SOUTH	EMPTY
EAST	BLOCKED
SOUTH	BLOCKED
WEST	EMPTY
SOUTH	BLOCKED
WEST	BLOCKED
NORTH	EMPTY
WEST	EMPTY
WEST	BLOCKED
NORTH	BLOCKED
EAST	EMPTY
NORTH	BLOCKED
DONE	