

Задача А. Отожги мирных ферзей!

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

Расставить N ферзей на квадратной доске $N \times N$ так, чтобы никакие два не били друг друга — очень непростая задача. Именно поэтому её поручили вам.

Формат входного файла

В единственной строке входного файла находится число $4 \leq N \leq 200$ — размеры доски.

Формат выходного файла

Выведите N чисел a_i : a_i — это номер горизонтали, на которую вы поставите ферзя, занимающего i -тую вертикаль. Нумерация горизонталей идёт снизу вверх, от 1 до N (как на обычной шахматной доске).

Примеры

<code>queens.in</code>	<code>queens.out</code>
5	4 2 5 3 1

Задача В. ЮграНефтеТранс

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

Ханты-Мансийский автономный округ — Югра является важнейшим нефтяным регионом России. Добыча нефти составляет 267 млн т в год, её транспортировка осуществляется по трубопроводам, общая длина которых превышает длину экватора Земли.

Система транспортировки нефти представляет собой совокупность n распределительных станций и m трубопроводов. Каждый трубопровод соединяет две различные станции. Между любыми двумя станциями проложено не более одного трубопровода.

Эффективность работы станций существенно зависит от вязкости нефти. Поэтому компания «ЮграНефтеТранс», в ведении которой находится сеть трубопроводов, заказала инновационному исследовательскому предприятию разработку и изготовление новых сверхточных датчиков вязкости на основе самых современных технологий.

Изготовление датчиков — процесс трудоёмкий и дорогостоящий, поэтому было решено изготовить k датчиков ($k \leq 40$) и выбрать k различных станций, на которых датчики будут установлены. Необходимо осуществить выбор станций так, чтобы датчики контролировали все трубопроводы: для каждого трубопровода хотя бы один датчик должен быть установлен на станции, где начинается или заканчивается этот трубопровод.

Напишите программу, которая проверяет, существует ли требуемое расположение датчиков, и в случае положительного ответа находит это расположение.

Формат входного файла

В первой строке входного файла записаны три натуральных числа — n , m и k ($k \leq n \leq 2000$, $1 \leq m \leq 10^5$, $1 \leq k \leq 40$). Далее следуют m строк, каждая из которых описывает один трубопровод. Трубопровод задаётся двумя целыми числами — порядковыми номерами станций, которые он соединяет. Станции пронумерованы от 1 до n . Гарантируется, что к любой станции подведён хотя бы один трубопровод и между любыми двумя станциями проложено не более одного трубопровода. Числа в каждой строке разделены пробелами.

Формат выходного файла

В первую строку выходного файла выведите слово `Yes`, если требуемое расположение датчиков существует, в противном случае — слово `No`. В случае положительного ответа выведите во вторую строку выходного файла k различных целых чисел — номера станций, на которых необходимо установить датчики. Номера можно выводить в любом порядке. Если существует несколько подходящих расположений датчиков, выведите любое из них. Разделяйте числа во второй строке пробелами.

Примеры

<code>pipeline.in</code>	<code>pipeline.out</code>
2 1 2 1 2	Yes 2 1
3 3 1 1 2 2 3 3 1	No
7 6 2 1 2 1 3 1 4 2 5 2 6 2 7	Yes 1 2
5 5 2 1 2 1 3 1 4 1 5 4 5	Yes 4 1