

Задача А. Горилла и ферзи

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Горилл пришел в клуб настольных игр ЛКШ, но никто не захотел с ним играть. К тому же, все нормальные настолки уже были разобраны.

Горилл нашел квадратную шахматную доску $n \times n$ и придумал себе занятие. Он хочет расставить на ней n ферзей так, чтобы никакие два не били друг друга.

Помогите гориллу решить эту непростую задачку.

Формат входных данных

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

Формат выходных данных

Выведите n чисел a_i : a_i — это номер горизонтали, на которую горилл поставит ферзя, занимающего i -ю вертикаль.

Нумерация горизонталей идёт снизу вверх, от 1 до n (как на обычной шахматной доске).

Пример

стандартный ввод	стандартный вывод
5	5 3 1 4 2

Задача В. Горилла и призы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Горилл решил раздать n призов ЛКШатам. Стоимости всех призов различны и выражаются натуральными числами от 1 до n .

Перед гориллом возникла задача распределить эти призы между k ЛКШатами так, чтобы все дети получили одинаковое количество призов, и, кроме того, суммарные стоимости призов, полученных разными участниками, совпадали.

Гарантируется, что n делится на k .

Формат входных данных

На вход программы поступают два числа: n и k ($1 \leq n \leq 200$, $1 \leq k \leq 200$, k является делителем n).

Формат выходных данных

Выведите k строк по $\frac{n}{k}$ чисел в каждой. В каждой строке должны быть выведены стоимости призов, которые вручаются соответствующему ЛКШатику.

Если распределить призы требуемым образом невозможно, выведите одно число 0.

Примеры

стандартный ввод	стандартный вывод
8 2	8 1 6 3 2 4 5 7
6 3	3 4 5 2 6 1

Задача С. Три круга

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Даны $N \leq 3$ круга на плоскости. Посчитайте площадь объединения.
Данную версию задачи предлагается решать методом Монте-Карло.

Формат входных данных

В первой строке число N ($1 \leq N \leq 3$).

В следующих N строках содержится по 3 целых числа — координаты центра и положительный радиус.

Все круги целиком лежат внутри квадрата $(0; 0) \times (100; 100)$.

Формат выходных данных

Выведите одно вещественное число — площадь объединения.

Допустимая относительная погрешность — 10^{-3} .

Примеры

стандартный ввод	стандартный вывод
1 99 99 1	3.14159325171203818172
2 2 2 2 1 1 1	13.15189494562218897045

Задача D. Гориллы против ЛКШат

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

В Горилляндии есть n городов, каждые два из которых соединены дорогой. Эти дороги были построены в давние времена *восточными* и *западными* гориллами. Дороги, которые были построены *восточными* гориллами, вымощены белыми камнями, а те, что построены *западными* — черными. Поскольку гориллы уже изобрели политическую дискриминацию, ни одна *восточная* горилла не может пройти по дороге, вымощенной черными камнями, и ни одна *западная* — по белой дороге.

Когда-то давно ЛКШата решили избрать своих правителей и изгнали горилл из Горилляндии. Однако недавно *восточные* и *западные* гориллы договорились вернуть Горилляндию под свой контроль. Для этого они хотят направить в некоторые города Горилляндии горилл, которые возьмут эти и смежные с ними города под свой контроль.

Точнее, если *восточный* горилл будет направлен в некоторый город, то он возьмет под свой контроль этот город и все города, которые напрямую соединены с ним белыми дорогами. Аналогично, *западный* горилл помимо города, в который он направлен, будет контролировать все города, напрямую соединенные с ним черными дорогами. Для захвата Горилляндии требуется установить контроль над всеми городами.

Однако, при разработке плана захвата обнаружилось две трудности. Во-первых, выяснилось, что горилл согласен принять участие в операции только если все гориллы, которые будут направлены в Горилляндию, будут представлять тот же вид, что и он. То есть либо все участвующие в захвате гориллы должны быть *восточными*, либо все они должны быть *западными*. Во-вторых, общее число горилл, которые могут быть направлены в Горилляндию, не должно превышать k . Единственная надежда горилл заключается в том, что k достаточно велико, $2^k \geq n$.

Выясните, *восточных* или *западных* горилл следует использовать для захвата Горилляндии, а также в какие города их следует направить.

Формат входных данных

В первой строке вводятся целые числа n и k ($2 \leq n \leq 256$, $2^k \geq n$, $k \leq n$).

Следующие n строк содержат по n целых чисел каждая. На i -й позиции i -й из этих строк расположено число 0, которое означает, что город не соединен дорогой сам с собой. Для всех $j \neq i$ число на j -й позиции i -й из этих строк равно 1, если i -й город соединен с j -м белой дорогой, и равно 2, если они соединены черной дорогой. Числа в строках разделены пробелами.

Гарантируется, что входные данные корректны, то есть если i -й город соединен с j -м белой дорогой, то и j -й соединен с i -м белой дорогой, аналогично в случае черных дорог.

Формат выходных данных

Если захватить Горилляндию при заданных условиях невозможно, выведите единственное число 0. В противном случае в первой строке выведите 1, если удастся захватить Горилляндию с использованием восточных горилл, и 2, если требуется использовать западных горилл.

В следующей строке выведите число $l \leq k$ — количество использованных горилл.

Третья строка должна содержать l целых чисел — номера городов, в которые следует направить горилл.

Заметьте, что вам не требуется минимизировать l . Если решений несколько, выведите любое.

Пример

стандартный ввод	стандартный вывод
8 3	1
0 1 1 1 1 2 2 2	3
1 0 1 1 2 1 2 2	4 6 7
1 1 0 1 2 2 1 2	
1 1 1 0 2 2 2 1	
1 2 2 2 0 2 1 1	
2 1 2 2 2 0 2 1	
2 2 1 2 1 2 0 2	
2 2 2 1 1 1 2 0	

Задача E. SO-SAT

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Найдите решение 3-SAT. Гарантируется, что оно существует.

Формулировка 3-SAT: нужно подобрать значения n булевых переменных так, чтобы все m утверждений вида $x_{i_1} = e_1 \vee x_{i_2} = e_2 \vee x_{i_3} = e_3$ обратились в истину.

Формат входных данных

На первой строке число переменных n и число утверждений m ($1 \leq n \leq 90, 1 \leq m \leq \min(n^2, 1000)$).

Каждая из следующих m строк содержит числа $i_1, e_1, i_2, e_2, i_3, e_3$ и задает утверждение $x_{i_1} = e_1 \vee x_{i_2} = e_2 \vee x_{i_3} = e_3$.

Все тесты случайны, тем не менее гарантируется, что решение существует.

Формат выходных данных

Выведите строку из n нулей и единиц — значения переменных.

Если у данной задачи 3-SAT есть несколько решений, выведите любое.

Пример

стандартный ввод	стандартный вывод
2 3 1 0 1 0 1 0 2 0 2 1 1 1 1 1 2 1 1 1	01

Задача F. Хорошие раскраски

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Назовем раскраску клеток таблицы $n \times m$ хорошей, если никакие четыре клетки, центры которых образуют вершины прямоугольника со сторонами, параллельными осям координат, не покрашены в один цвет. Иначе говоря, для раскраски не должно быть четверки целых чисел x_1, x_2, y_1, y_2 , что $1 \leq x_1 < x_2 \leq n, 1 \leq y_1 < y_2 \leq m$, и клетки $(x_1, y_1), (x_2, y_1), (x_1, y_2)$ и (x_2, y_2) покрашены в одинаковый цвет.

Требуется написать программу, которая по заданным целым числам n, m и c находит любую хорошую раскраску таблицы $n \times m$ в c цветов.

Формат входных данных

В первой строке записаны три целых числа n, m, c ($2 \leq n, m \leq 10, 2 \leq c \leq 3$).

Гарантируется, что для заданных во входных данных значений существует хотя бы одна хорошая раскраска.

Формат выходных данных

Выведите n строк по m чисел в каждой.

В качестве j -го числа i -й строки выведите $a_{i,j}$ — цвет клетки (i, j) ($1 \leq a_{i,j} \leq c$).

Если есть несколько хороших раскрасок, можно вывести любую из них.

Пример

стандартный ввод	стандартный вывод
2 2 2	1 2 2 2

Замечание

Примените какой-нибудь неточный алгоритм нахождения минимума функции, например алгоритм отжига.

Задача G. ЮграНефтеТранс

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

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 входных данных

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 выходных данных

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

стандартный ввод	стандартный вывод
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

Задача Н. Голосование

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Это задача на два запуска!

В Соединительных Штатах Америки в прошлом году проходили выборы, на которых выиграл Д. Трамп, набрав $100\% \cdot p_t = 49.91\%$ голосов избирателей. Камала Харрис набрала $p_k \cdot 100\% = 48.43\%$ голосов. Остальные кандидаты в сумме набрали $p_z \cdot 100\% = 1.66\%$ голосов избирателей.

В память о своей победе Д. Трамп решил сохранить первые n избирательных бюллетеней с самыми красивыми номерами (можете считать, что они выбраны случайно, и что n сильно меньше общего числа бюллетеней).

Но бумага может сгнить со временем, а поэтому Д. Трамп нанял вас для того, чтобы вы закодировали последовательность кандидатов, за которых голосовали в его n бюллетенях.

Гарантируется, что все наборы входных данных сгенерированы случайно, в соответствии с указанными вероятностями.

Формат входных данных

В первой строке будет введена задача для вашей программы $\text{type} \in \{\text{encode}, \text{decode}\}$, которая обозначает, нужно вам закодировать информацию о бюллетенях или раскодировать.

Если $\text{type} = \text{encode}$, то во второй строке будет указано число n ($1 \leq n \leq 100\,000$). В третьей строке будет указана строка s длины n , состоящая из символов «Т», «К», «.», очередной символ обозначает, за кого был отдан голос: за Трампа, за Камалу или за иного кандидата.

Если $\text{type} = \text{decode}$, то во второй строке будет указано число n ($1 \leq n \leq 100\,000$) — количество бюллетеней. В третьей строке будет указана бинарная строка t , которую вывела ваша программа на первом запуске. Обратите внимание, что t не обязательно имеет длину n .

Формат выходных данных

Если $\text{type} = \text{encode}$, то ваша программа должна вывести бинарную строку t , которая состоит из символов «0» и «1» и имеет длину не более чем 112 000 символов.

Если $\text{type} = \text{decode}$, то просто выведите строку s , которая была закодирована на первом запуске.

Примеры

стандартный ввод	стандартный вывод
encode 15 ТКТКТТТТКТТ.ТТТ	10100000111101011100100101
decode 15 10100000111101011100100101	ТКТКТТТТКТТ.ТТТ

Замечание

Вспомните лекцию, энтропия одного бюллетеня примерно равна 1.105 бит. Вас просят написать эффективный способ кодирования информации, который почти не уступает самому эффективному из возможных.

Задача I. Научное доказательство любви

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Синъя Юкимуру и Аямэ Химуро работают в одной Computer Science лаборатории. Юкимуру раньше занимался спортивным программированием, а поэтому иногда рассказывает Химуро различные алгоритмы. К удивлению Юкимуру, Химуро всегда выслушивает его, и долгое время Химуро думал, что ей просто интересно слушать про алгоритмы.

Но недавно Юкимуру и Химуро сходили в кино, потом в кофейню, а потом домой к Химуро, где два сотрудника случайно поцеловались. Эти события смутили Юкимуру, и он начал сомневаться в том, что Химуро относится к нему просто как к другу, а не как к возлюбленному. Поэтому он решил воспользоваться своими знаниями и достоверно определить, правда ли это.

Юкимуру решил провести несколько опытов. В каждом из опытов Юкимуру рассказывает Химуро алгоритм на одну из 5 тем. Юкимуру помнит, что когда Химуро ему говорила, что ей нравятся какие-то 2 из этих тем, но он забыл какие именно. Также Юкимуру прочитал некоторые исследования на эту тему и понял, что:

- Если Юкимуру рассказывает Химуро алгоритм на тему, которая ей **нравится** и при этом она его **любит**, то она говорит, что алгоритм ей понравился с вероятностью 90%;
- Если Юкимуру рассказывает Химуро алгоритм на тему, которая ей **не нравится** и при этом она его **любит**, то она говорит, что алгоритм ей понравился с вероятностью 50%;
- Если Юкимуру рассказывает Химуро алгоритм на тему, которая ей **нравится** и при этом она его **не любит**, то она говорит, что алгоритм ей понравился с вероятностью 70%;
- Если Юкимуру рассказывает Химуро алгоритм на тему, которая ей **не нравится** и при этом она его **не любит**, то она говорит, что алгоритм ей понравился с вероятностью 30%;

Помогите определить Юкимуру, любит ли его Химуро.

Формат входных данных

В первой строке указано число q ($q = 150$) — количество опытов, которые может совершить Юкимуру.

Протокол взаимодействия

Чтобы сделать любое из действий выведите одну из строк ниже, не забыв вывести перенос строки и очистить буфер вывода.

- `algo x` ($1 \leq x \leq 5$). Химуро рассказывает Юкимуру алгоритм на тему x , в ответ Химуро отвечает строкой `like`, если алгоритм ей понравился и `no` в ином случае;
- `love`. Химуро делает вывод, что Юкимуру любит его;
- `friend`. Химуро делает вывод, что Юкимуру не любит его;

Пример

стандартный ввод	стандартный вывод
300	algo 1
like	algo 2
no	algo 3
no	algo 4
no	friend

Задача J. Проще всего красить в три цвета

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан случайный неориентированный граф G из n вершин и m ребер. Ваша задача — покрасить его вершины в три цвета таким образом, чтобы смежные вершины были покрашены в разные цвета. Гарантируется, что покрасить граф в три цвета возможно.

Формат входных данных

На первой строке число вершин n ($1 \leq n \leq 250$) и число ребер $m \geq 1$.

Следующие m строк содержат пары чисел от 1 до n — ребра графа.

В графе нет ни петель, ни кратных ребер.

Формат выходных данных

На следующей строке n целых чисел от 1 до 3 — цвета вершин. Если требуемых раскрасок несколько, выведите любую.

Пример

стандартный ввод	стандартный вывод
5 5 1 2 2 3 3 1 4 5 1 5	1 2 3 1 2

Задача К. Мазохизм

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

На вход подается черно-белое изображение 16×16 . Каждое изображение нарисовал один из преподавателей Т-Поколения. Вам надо угадать, что рисовал преподаватель: квадрат, круг или что-то другое.

Формат входных данных

В первой строке указано число n ($1 \leq n \leq 22$) — количество картинок.

Каждая картинка описывается 16 строками, каждая из которых содержит по 16 символов «В» или «.».

Формат выходных данных

Для каждого изображения выведите одну из строк «square», «circle» или «unknown».

Примеры

стандартный ввод	стандартный вывод
<pre> 1BBBBB.... ...BB...B.... ...B...B....BBBB....BB....BB....BB....B....B....B.... </pre>	<pre> unknown </pre>
<pre> 1BBBBBBBBBBB... ..B...B... ..B...BB... ..B...B... ..B...B... ..B...B... ..B...B... ..B...B... ..B...B... ..B...B... ..B...B... ..B...B... ..BBBBBBBBBBB... </pre>	<pre> square </pre>
<pre> 1BBBBB.... ...BB...BB.... ..BB...BB.... ..B...B... ..B...BB... ..B...B... ..B...B... ..BB...B... ..B...B... ..BB...BB... ..BB.BBBBBB....BBB.B.... </pre>	<pre> circle </pre>