

Задача А. Разбиения на слагаемые

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

Перечислите все разбиения целого положительного числа N ($1 \leq N \leq 40$) на целые положительные слагаемые. Разбиения должны обладать следующими свойствами:

1. Слагаемые в разбиениях идут в невозрастающем порядке.
2. Разбиения перечисляются в лексикографическом порядке.

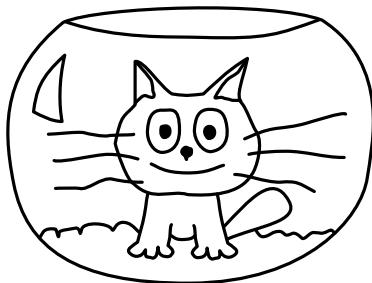
Пример

<code>partition.in</code>	<code>partition.out</code>
4	1 1 1 1
	2 1 1
	2 2
	3 1
	4

Задача В. Продавец аквариумов

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

Продавец аквариумов для кошек хочет объехать n городов, посетив каждый из них ровно один раз. Помогите ему найти кратчайший путь.



Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 13$) — количество городов. Следующие n строк содержат по n чисел — длины путей между городами.

В i -й строке j -е число — $a_{i,j}$ — это расстояние между городами i и j ($0 \leq a_{i,j} \leq 10^6$; $a_{i,j} = a_{j,i}$; $a_{i,i} = 0$).

Формат выходных данных

В первой строке выходного файла выведите длину кратчайшего пути. Во второй строке выведите n чисел — порядок, в котором нужно посетить города.

Пример

<code>aquarium.in</code>	<code>aquarium.out</code>
5	666
0 183 163 173 181	1 3 2 5 4
183 0 165 172 171	
163 165 0 189 302	
173 172 189 0 167	
181 171 302 167 0	

Задача С. Количество циклов

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

Формально, *путь* в графе — это чередующаяся последовательность вершин и рёбер $u_1, e_1, u_2, e_2, u_3, \dots, u_k$, начинающаяся и заканчивающаяся вершиной и такая, что любые соседние вершина и ребро в ней инцидентны.

Цикл — это путь, начальная и конечная вершины которого совпадают. В цикле должно быть хотя бы одно ребро.

Простой путь отличается от обычного пути тем, что в нём не может быть повторяющихся вершин.

Простой цикл — это цикл, в котором нет повторяющихся вершин и рёбер.

Дан неориентированный граф. Посчитайте, сколько в нём различных простых циклов. Заметим, что циклы считаются одинаковыми, если они обходят одно и то же множество вершин в одном и том же порядке, возможно, начиная при этом из другой вершины, или если порядок обхода противоположный. Например, циклы с порядком обхода вершин 1, 2, 3, 1, 2, 3, 1, 2 и 1, 3, 2, 1 считаются одинаковыми, а циклы 1, 2, 3, 4, 1 и 1, 3, 4, 2, 1 — нет, поскольку порядок обхода вершин различен.

Формат входных данных

В первой строке входного файла заданы числа N и M через пробел — количество вершин и рёбер в графе, соответственно ($1 \leq N \leq 10$). Следующие M строк содержат по два числа u_i и v_i через пробел ($1 \leq u_i, v_i \leq N$, $u_i \neq v_i$); каждая такая строка означает, что в графе существует ребро между вершинами u_i и v_i . В графе нет кратных рёбер.

Формат выходных данных

Выведите одно число — количество простых циклов в заданном графе.

Примеры

<code>numcycle.in</code>	<code>numcycle.out</code>
3 2 1 2 2 3	0
4 5 1 2 2 3 3 4 4 1 1 3	3

Задача Д. Мнемоника и палиндромы 3

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

Как вы помните, когда Васечкин готовил задачу для последнего студенческого соревнования, он долго пытался придумать для неё необычное и сложное название. Но название, придуманное им, оказалось настолько сложным, что никто из участников того соревнования даже не

стал читать текст задачи Васечкина. После этого председатель программного комитета заявил, что отказывается участвовать в подготовке соревнований, пока в программном комитете работают такие неадекватные люди, как Васечкин. Так Васечкин стал новым председателем программного комитета и начал готовить следующее соревнование по программированию. Васечкин решил, что на этот раз названия всех задач будут состоять только из букв a , b и c и иметь длину n . Кроме того, все названия должны быть *чрезвычайно сложными*, а именно, никакая подстрока никакого названия, состоящая из двух или более символов, не должна являться палиндромом. Помогите Васечкину и найдите все чрезвычайно сложные названия для задач предстоящего соревнования.

Формат входных данных

В единственной строке записано целое число n ($1 \leq n \leq 20000$).

Формат выходных данных

Выведите все различные чрезвычайно сложные названия длины n , состоящие только из букв a , b и c . Названия следует выводить в алфавитном порядке, по одному в строке. Если суммарная длина названий превосходит 100000 букв, выведите единственную строку «TOO LONG».

Примеры

name.in	name.out
2	ab ac ba bc ca cb

Задача Е. Маленький холодильник

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

Дано целое число n ($1 \leq n \leq 10^{12}$). Нужно найти натуральные a, b, c : $abc = n$ и при этом $2(ab + bc + ca)$ минимально. Т.е. при фиксированном объеме минимизировать площадь поверхности.

Формат входных данных

На первой строке число n ($1 \leq n \leq 10^{12}$).

Формат выходных данных

На первой строке четыре целые числа — $2(ab + bc + ca)$ и a, b, c .

Примеры

refrator.in	refrator.out
120	148 4 6 5

Задача F. Большой холодильник

Имя входного файла: `refrigerator.in`
Имя выходного файла: `refrigerator.out`
Ограничение по времени: 5 seconds
Ограничение по памяти: 256 megabytes

Вася хочет купить новый холодильник. Он считает, что холодильник должен быть прямоугольным параллелепипедом с целочисленными длинами ребер. Вася рассчитал,

что для повседневного пользования ему понадобится холодильник объема не меньше V . Кроме того, Вася по натуре минималист, поэтому объем должен быть и не больше V — к чему занимать лишнее место в квартире? Определившись с объемом холодильника, Вася столкнулся с новой непростой задачей — чтобы холодильник было проще мыть, при фиксированном объеме V он должен иметь минимальную площадь поверхности.

Объем и площадь поверхности холодильника с ребрами a, b, c равны $V = abc$ и $S = 2(ab + bc + ca)$, соответственно.

Помогите Васе по заданному объему V найти такие целые длины ребер холодильника a, b, c , чтобы объем холодильника был равен V и при этом его площадь поверхности S была минимальна.

Формат входных данных

В первой строке записано единственное целое число t ($1 \leq t \leq 500$) — количество наборов данных.

Далее следует описание t наборов данных. Каждый набор состоит из одного целого числа V ($2 \leq V \leq 10^{18}$), заданного своим разложением на множители следующим образом.

Пусть $V = p_1^{a_1} p_2^{a_2} \dots p_k^{a_k}$, где p_i — различные простые числа, а a_i — положительные целые степени.

Тогда в первой строке описания набора данных записано единственное положительное целое число k — количество различных простых делителей V . В следующих k строках записаны простые числа p_i и их степени a_i , разделенные пробелом. Все p_i различны, все $a_i > 0$.

Формат выходных данных

Выполните t строк, в i -й строке выведите ответ на i -й набор данных — четыре целых числа, записанные через пробел: минимальная возможная площадь поверхности S и соответствующие длины ребер a, b, c . Если вариантов длин ребер, дающих минимальную площадь, несколько, разрешается вывести любой из них. Длины ребер холодильника разрешается выводить в любом порядке.

Примеры

refrigerator.in	refrigerator.out
3	24 2 2 2
1	70 1 1 17
2 3	148 4 6 5
1	
17 1	
3	
3 1	
2 3	
5 1	

Замечание

В первом наборе данных примера объем холодильника $V = 2^3 = 8$, и минимальную площадь поверхности дадут ребра одинаковой длины.

Во втором наборе данных объем $V = 17$, и его можно получить из единственного набора ребер целочисленных длин.

Задача G. Маленький трубопровод

Имя входного файла: piline.in
Имя выходного файла: piline.out
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 «Югра-НефтеТранс», 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 ($k \leq n \leq 100$, $1 \leq m \leq 100$, $1 \leq k \leq 10$). 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.

Пример

piline.in	piline.out
2 1 2 1 2	Yes 1 2
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 H. Большой трубопровод

Имя входного файла: pipeline.in
Имя выходного файла: pipeline.out
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 «Югра-НефтеТранс», 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.

Пример

pipeline.in	pipeline.out
2 1 2 1 2	Yes 1 2
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