Задача A. Stack permutations

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

Настоящий хит среди игрушек этого года — стек для перестановки детских кубиков. Используя такой стек и исходную последовательность из кубиков от 1 до n, любой ребенок может с легкостью получить множество новых чудесных последовательностей, следуя следующим правилам:

1. Взять очередной кубик из начала исходной последовательности и добавить его в стек.
2. Вытащить кубик из стека и добавить его в конец результирующей последовательности.

Стеком называется простая структура, в которой кубики вытаскиваются в порядке, обратном к тому, в котором они были добавлены. Результирующая последовательность изначально пуста.

Однако со временем дети заметили, что даже эта чудесная игрушка не идеальна — с ее помощью нельзя получить все возможные последовательности! Теперь главный вопрос, который волнует всех обеспокоенных детей и их родителей — сколько возможных последовательностей длины n невозможно получить с помощью этой игрушки? Вы можете им помочь?

Формат входных данных

Единственная строка входных данных содержит целое число n ($1 \leqslant n \leqslant 100000$) — длина исходной последовательности кубиков.

Формат выходных данных

Выведите количество последовательностей длины n, которые нельзя получить с помощью стека для перестановки кубиков, по модулю $10^9 + 7$.

Примеры

stack.in	stack.out
2	0
3	1
5	78

Задача B. Network

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

Интернет-протокол версии 4 (IPv4) является четвертой версией протокола (IP), используемого на данный момент в Интернете .

IPv4 использует 32-битные адреса, что ограничивает адресное пространство до 429496720096 (2^{32}) адресов. Многие адреса были розданы для пользования частным лицам, и поэтому количество свободных адресов уменьшилось значительно. Истощение адресов IPv4 произошло 3 февраля 2011 года, хотя данный срок был значительно отложен применением таких технологий как классовое проектирование сети, CIDR (Classless Inter-Domain Routing) и NAT (Network Address Translation). Ограниченность IPv4 стимулировало развитие новой версии протокола IPv6 в 1990 году, которая начала внедряться в Интернет в 2006 году.

В IPv4 некоторые адреса зарезервированы для частных сетей, и часть зарезервирована для так называемых адресов многоадресной рассылки (multicast addresses). Адреса IPv4 могут быть записаны в любом виде, соответствующем 32-битному целому числу, но для удобства чтения человеком обычно используется формат в виде четырех целых десятичных чисел, разделенных точками.

Процесс разделения сети на подсети включает в себя разделение IP-адреса, идентифицирующего сеть, на две части, одна из которых соответствует новой подсети, а другая соответствует адресу внутри данной подсети. Это делается путем битовой операции AND маски подсети и IP адреса изначальной сети. Полученное число является идентификатором новой подсети, а оставшаяся часть – адресом хоста внутри подсети.

Маска подсети состоит из 32 битов, последовательности единиц, за которой следует последовательность нулей. Нули соответствуют части идентификатора хоста. Маска подсети обычно выражается в виде “/”, где n – число единиц в маске в пределах от 0 до 32.

	Бинарная форма	Десятичная запись
IP адрес	11000000.10101000.00000101.10000010	192.168.5.130
Маска подсети	11111111.11111111.11111111.00000000	255.255.255.0
Префикс сети	11000000.10101000.00000101.00000000	192.168.5.0
Хост	00000000.00000000.00000000.10000010	0.0.0.130

Также в сети есть специальные адреса, зарезервированные для обеспечения работы сети:
0.0.0.0 – 0.255.255.255 (все адреса с 0.0.0.0 до 0.255.255.255, оба включительно),
10.0.0.0 – 10.255.255.255
100.64.0.0 – 100.127.255.255
127.0.0.0 – 127.255.255.255
169.254.0.0 – 169.254.255.255
172.16.0.0 – 172.31.255.255
192.0.0.0 – 192.0.0.7
192.0.2.0 – 192.0.2.255
192.168.0.0 – 192.168.255.255
198.18.0.0 – 198.19.255.255
198.51.100.0 – 198.51.100.255
203.0.113.0 – 203.0.113.255
240.0.0.0 – 255.255.255.254
и
255.255.255.255

Артем получил IP-адрес на свой день рождения. Но при получении он потерял символы (“.”) и ("/"). Поэтому он захотел узнать все возможные IPv4-адреса, которые он мог получить. Но он сейчас очень занят, поэтому он попросил помощи от вас. Можете ли вы помочь определить все возможные адреса IPv4?

Артем уверен, что он получил действительный адрес IPv4, но он не помнит, была ли там подсеть или нет. Каждая часть правильного IP-адреса не содержит ведущих нулей в десятичной записи. Например, 1.20.0.0, 1.20.0.0/1, 1.20.0.0/0 – правильные адреса, а 1.020.0.0, 01.20.0.0/1, 01.20.00.0/0, 1.20.0.0/01 неверны. Кроме того, правильный IP-адрес не содержится в наборе зарезервированных адресов (см. выше). Например, 198.18.0.0, 198.18.0.0/2, 198.18.0.0/32, 198.19.2.10 неправильны, а 1.92.180.0/32, 19.21.80.0/32 правильны.

Формат входных данных

Первая строка входного файла содержит IPv4-адрес без точек (".") и слэшей ("/") в десятичной записи. Длина строки не превосходит 20 символов.

Формат выходных данных

Первая строка вывода должна содержать количество возможных адресов. Следующие строки должны содержать эти адреса отсортированные лексикографически (т.е. если мы будем рассматривать адреса как строки).

Примеры

network.in	network.out
1234	1 1.2.3.4
192180032	8 1.92.180.0/32 19.2.180.0/32 19.21.80.0/32 19.218.0.0/32 192.1.80.0/32 192.18.0.0/32 192.180.0.3/2 192.180.0.32

Задача C. Polygons

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

У Вас имеются N досок для постройки забора. Вы хотите построить забор из этих досок в форме многоугольника. К тому же, Вы хотите построить как можно больше заборов. Заметьте, что есть одно условие: площадь каждого забора, построенного в форме многоугольника, должна быть строго больше нуля. Удачно провести выходные с постройкой заборов!

Формат входных данных

Входной файл содержит число N и N чисел: длины досок, которые у Вас имеются. Каждое число во входном файле больше нуля, но и не больше 15.

Формат выходных данных

Выходной файл должен содержать одно число - максимальное количество заборов, которые можно построить из данных Вам досок.

Примеры

polygons.in	polygons.out
6 1 2 15 10 2 15	2
6 1 1 2 10 15 15	1

Замечание

Никакая доска не может быть использована в более чем одном заборе.

Задача D. Simple Kingdom

Имя входного файла:	`simplegraph.in`
Имя выходного файла:	`simplegraph.out`
Ограничение по времени:	4 секунды
Ограничение по памяти:	64 мегабайта

В одном “простом” Королевстве есть несколько городов (их ровно $2N$) и двусторонние дороги, соединяющие их (дорог ровно $N(N+1)/2$). Известно, что из любого города можно добраться в любой другой, следуя по этим дорогам одним или несколькими вариантами. Также известно, что для любого i ($1 \leqslant i \leqslant N$), существует ровно 2 города, из которых выходит ровно i дорог.

Вам дано число N. Требуется выяснить, существует ли такое Королевство, удовлетворяющее вышеперечисленным требованиям. Если да, то необходимо найти города, которые соединены напрямую.

Формат входных данных

Входной файл содержит единственное число $1 \leqslant N \leqslant 1000$.

Формат выходных данных

Если “простое” Королевство не существует (не существует никакого способа сделать королевство с заданными ограничениями), выведите в ответ единственную строку с числом -1. Во всех остальных случаях, первая строка выходного файла должна содержать M - количество дорог в “простом"Королевстве, после чего должно следовать описание каждой дороги в виде двух чисел (номера городов, которые она соединяет). Дорога не должна вести из города в него самого, и не более одной дороги должны соединять два города напрямую.

Примеры

simplegraph.in	simplegraph.out
1	1 1 2
2	3 1 2 1 3 2 4

Замечание

Города пронумерованы числами от 1 до $2N$.

Задача E. Tree

Имя входного файла:	`tree.in`
Имя выходного файла:	`tree.out`
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Вам дано описание сообщества ВКонтакте. В него входят N людей, и между некоторыми парами из них существует связь. Два человека являются связанными, если они связаны напрямую или через какого-либо другого человека. Для того, чтобы хранить меньше информации, ВКонтакте хранит связи в таком виде, что если вы хотите узнать о связи между двумя людьми, то вы найдете ровно один путь между ними.

Однако ВКонтакте не может позволить хранить сообщество, если максимальная дистанция между людьми в нем больше K.

Таким образом, было решено удалить минимальное количество связей между некоторыми парами так, чтобы разделить сообщество на несколько новых сообществ, таких что в каждом из них максимальная дистанция между людьми не превышает K. И вам предстоит решить эту задачу!

Формат входных данных

В первой строке входного файла заданы два целых числа: N ($1 \leqslant N \leqslant 1000$) и K ($1 \leqslant K \leqslant N$). Каждая из последующих $N-1$ строк содержит два целых числа U, V ($1 \leqslant U, V \leqslant N$) означающих, что существует связь между людьми U и V .

Формат выходных данных

Первая строка выходного файла должна содержать одно целое число M – минимальное количество связей, которые нужно удалить. Каждая из последующих M строк должна содержать номера этих связей. Связи пронумерованы от единицы в порядке появления во входном файле.

Примеры

tree.in	tree.out
16 3 8 5 16 5 9 8 7 16 15 9 13 9 12 16 3 9 1 8 4 7 6 15 2 1 11 13 14 11 10 12	4 9 1 4 6
16 3 9 13 10 9 7 9 12 7 3 9 11 10 6 7 4 11 16 3 2 9 15 2 5 10 8 5 14 2 1 13	5 15 2 6 3 5