

Задача A. DJ

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайт

Все стало на свои места в Уэльсе, война закончилась, и для поднятия настроения, в село Лланвайрпуллгвнгллогерихурндробулллантисилогогох приехал DJ с самым длинным именем на планете: Adolph Blaine Charles David Earl Frederick Gerald Hubert Irvim John Kenneth Martin Nero Oliver Paul Randolph Sherman Thomas Uncas Victor Willian Xerxes Yancy Zeus. Он гордится своим именем, потому и взял его как свое диджейское прозвище. Этот DJ известен не только своим именем, но еще и необычным порядком проигрывания треков. Пусть ему доступно N треков. Сначала он помечает каждый из треков некоторой буквой от A до Z, причем i -й трек будет обозначен i -м символом последовательности AB...Z. После этого последовательность символов, соответствующая трекам, которые будут проигрываться, строится так: сначала идут все возможные слова из одной буквы в лексикографическом порядке, затем все слова из двух букв в лексикографическом порядке, и так далее. Поэтому, если $N = 3$, то последовательность треков будет следующей:

ABCAAABACBAVBVVCCACVCCSAAAAABAACAABAABVAVC...

После завершения вечеринки некоторые из посетителей спросили у DJ название трека, который был проигран j -м. DJ нужно помнить свое имя, поэтому он не хочет запоминать лишнюю информацию. Пожалуйста, напишите для него программу, которая по номеру в проигранной последовательности треков будет определять название трека.

Формат входного файла

В первой строке входного файла находятся два числа N и Q - количество треков у DJ, и количество запросов от посетителей соответственно. В следующей строке записаны N названий треков (каждое из которых состоит минимум из 1 и максимум из 100 символов, каждый из которых является либо цифрой, либо буквой латинского алфавита), разделенных одним пробелом. Последняя строка содержит последовательность запросов, каждый из которых – одно целое число k ($1 \leq k \leq 10^9$)- номер запрошенного трека.

Формат выходного файла

Для каждого запроса выведите отдельную строку – имя трека с соответствующим порядковым номером в проигранной последовательности треков.

Пример

стандартный ввод	стандартный вывод
3 5 Hot Contagious Girlfriend 1 2 3 4 16 0 0	Hot Contagious Girlfriend Hot Girlfriend

Задача В. МАКСИМАЛЬНО СВЯЗНЫЕ ВЕРШИНЫ

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайт

Дан неориентированный граф. Назовем две вершины s и t «максимально связными», если из одной вершины достижима другая и максимальное количество непересекающихся по ребрам путей между этими вершинами равно минимуму из степеней этих двух вершин.

Необходимо определить минимальное количество ребер (петли запрещаются), которое нужно добавить в граф, чтобы в нем образовалась хотя бы одна пара таких вершин.

Формат входного файла

В первой строке входного файла заданы n и m , $1 \leq n \leq 10^5$, $0 \leq m \leq 10^5$ – количество вершин и ребер соответственно. В следующих m строках заданы ребра – пары чисел u_i, v_i , $1 \leq u_i, v_i \leq n$, $u_i \neq v_i$.

Формат выходного файла

В выходной файл необходимо вывести единственное число – минимальное количество ребер, которое нужно добавить в граф, чтобы получить хотя бы одну пару максимально связных вершин. Если ответ не существует, необходимо вывести -1.

Пример

стандартный ввод	стандартный вывод
1 0	-1
3 2 1 2 2 3	0
2 0	1

Задача С. ВЕРЕВКА ПРИНЦА

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайт

В романе Стивена Кинга «Глаза дракона» есть эпизод, где принца заточили в башню. Чтобы освободиться, он стал вынимать нити из тканых салфеток, которые приносили ему вместе с едой. Из этих нитей принц делал веревку. Чтобы не вызвать подозрения, он вынимал разное количество нитей в разные недели. Будем считать, что каждую неделю принц случайным образом брал столько нитей, что их хватало на участок веревки, длина которого равномерно распределена между 0 и 1 метром. Для побега нужна веревка длиной хотя бы n метров.

Определите среднее количество недель, когда длина веревки впервые превысит n метров, и принц сможет убежать.

Формат входного файла

Вводится единственное действительное число n ($0 < n \leq 20$).

Формат выходного файла

Выведите среднее количество недель, после которых получится веревка длиной хотя бы n метров. Ответ округлите до второго знака после запятой.

Пример

стандартный ввод	стандартный вывод
1	2.72
15.5	31.67
18	36.67

Задача D. ПРОГУЛКА ПО ГОРОДУ

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайт

В Киеве, недалеко от Днепра стоит Казак: каменный, с большими усами, с огромной саблей, к небу гордо поднятой головой. Киев – любимый город туристов, которые съезжаются сюда отовсюду. Вася – один из них. Карта Киева представляет собой таблицу размером $N \times M$, ячейками которой являются кварталы, некоторые из которых доступны для движения, а некоторые – нет. Также на карте отмечены интересные места, причем это также кварталы, по которым возможно движение. Вася выбрал маршрут для ознакомления с городом. Он начнет с ячейки $(1; 1)$, придет в ячейку $(N; M)$, а потом снова вернется туда откуда пришел (в ячейку $(1; 1)$). Конечно, Вася хочет посмотреть как можно больше разных интересных мест. Поскольку Вася знает Киев очень плохо, да и, вообще, заблудиться здесь очень нежелательно, он решил, что на пути к ячейке (N, M) будет всегда идти вправо, или вниз (относительно карты), а на обратном пути – влево, или вверх. Помогите Васе найти максимальное количество разных интересных мест, в которых он может побывать.

Формат входного файла

В первой строке входного файла содержатся два числа N и M ($1 \leq N, M \leq 100$). В следующих N строках содержится описание карты – строка из M букв, причем “#” означает квартал недоступный для движения, “*” – место для просмотра, а “.” (точка) – квартал доступный для движения. Гарантируется, что ячейки $(1; 1)$ и $(N; M)$ не будут помечены символом “#”.

Формат выходного файла

В единственной строке выходного файла должно содержаться одно число – максимальное количество разных мест для просмотра, в которых может побывать Вася в течение своего маршрута, или -1, если маршрут осуществить не удастся.

Пример

стандартный ввод	стандартный вывод
<pre>4 5 **.*. ..##. *#*.. *.*.*</pre>	7

Задача Е. ПРОЕКЦИИ – 3

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти 64 мегабайт

Есть куб $n \times n \times n$, который частично заполнен кубиками размером $1 \times 1 \times 1$. Даны его три проекции – спереди, справа и сверху. Все три проекции представляют собой матрицы размером $n \times n$, содержащие 1 – если есть хотя бы один кубик в этом ряду, и 0 – в противном случае.

По заданным проекциям необходимо определить максимальное количество кубиков, которое есть в кубе.

Формат входного файла

В первой строке входного файла задано n , $1 \leq n \leq 10^3$. Далее следуют $3 \times n$ строк по n чисел $\{0,1\}$, заданных через пробел – три проекции (спереди, справа, сверху).

Формат выходного файла

В единственной строке нужно вывести максимальное количество кубиков. Если проекции противоречат друг другу, необходимо вывести -1.

Пример

стандартный ввод	стандартный вывод
2 1 0 1 1 0 1 1 1 1 1 1 0	4
1 1 0 0	-1

Задача F. РОМАНТИЧЕСКАЯ ПОЭЗИЯ

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	8 секунд
Ограничение по памяти:	64 мегабайт

Стас хочет посвятить своей подруге Алине стихотворение. Он решил составить стихотворение с кольцевым парным рифмованием. Стихотворение с кольцевым парным рифмованием имеет такую схему: AB BC CD DE ... YZ ZA. То есть каждая строфа имеет две строки, причем вторая строка всегда рифмуется с первой следующей строфы, кроме того, вторая строка последней строфы рифмуется с первой строкой первой строфы.

У Стаса действительно богатая фантазия, поэтому он уже успел определить содержание всего стихотворения и отдельных его строф. Для некоторых строф парень придумал несколько различных вариантов, которые могут быть образованы различными парами рифм, но все они выражают одну и ту же мысль.

Для того, чтобы не гадать, какие строки достаточно рифмуются, а какие – нет, Стас пронумеровал все типы рифм натуральными числами. Каждый вариант некоторой строфы он обозначил парой (x, y) , где x – тип рифмы первой строки строфы, y – тип рифмы второй строки строфы. После строфы, которая задается парой (x, y) , может следовать строфа (y, z) для некоторого z .

Стас решил, что стихотворение будет состоять из n строф. Для каждой строфы известно количество вариантов, которые придумал Стас, – s_i , и каждый из этих вариантов. Необходимо определить, возможно ли построить стихотворение из n строф с кольцевым парным рифмованием, используя придуманные варианты строф?

Формат входного файла

Первая строка содержит натуральное число n – количество строф стихотворения ($1 \leq n \leq 1000$). Далее идет описание каждой из n строф. Описание i -й строфы начинается строкой, содержащей число s_i – количество вариантов, которые придумал Стас для i -й строфы. Далее записаны s_i строк, каждая из которых содержит два числа $a_{i,j}$ и $b_{i,j}$ – тип рифмы первой и второй строки j -го варианта i -й строфы ($1 \leq s_i \leq 1000, 1 \leq a_{i,j}, b_{i,j} \leq 10^9$).

Формат выходного файла

Программа должна вывести единственный символ «Y», если из существующих вариантов строф можно построить стихотворение из n строф по описанным условиям, или символ «N» в противном случае.

Пример

стандартный ввод	стандартный вывод
5 1 1 1 2 1 5 1 6 2 5 2 6 3 1 3 8 3 10 2 8 1 4 2	Y
2 1 2 1 1 1 3	N

Пояснение к примеру

В первом примере стихотворение имеет следующую схему рифмования:

11, 16, 63, 38, 81.

Задача G. ИНТЕРЕСНАЯ ЗАДАЧА

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайт

Роману, как лучшему специалисту компании Loo..., было предложено решить весьма нелегкую задачу. Сначала в памяти компьютера есть единственное число k . Нужно за наименьшее количество операций получить в памяти компьютера следующие числа: A_1k, A_2k, \dots, A_nk . При этом компьютер может выполнять такие операции:

- сложение двух чисел,
- вычитание двух чисел,
- побитовый сдвиг влево (сдвиг на t бит эквивалентен умножению на 2^t).

Все полученные промежуточные значения хранятся в памяти, поэтому их можно использовать при вычислении других значений. Заказчики хотят чтобы при вычислении никогда не получалось число большее чем $42k$. Гарантируется, что при соблюдении этого ограничения, в компьютере не происходит переполнения. Также компьютер не может работать с отрицательными числами, поэтому отнимать от меньшего числа большее также нельзя.

Порядок, в котором в памяти появятся числа A_1k, A_2k, \dots, A_nk не имеет значения.

У Романа свидание, поэтому он просит вашей помощи.

Формат входного файла

Первая строка содержит число N ($1 \leq N \leq 41$) – количество необходимых значений. Во второй строке содержится N чисел A_i ($2 \leq A_i \leq 42$). Все числа A_i разные. Само число k не дано, поэтому ваша последовательность операций должна быть верной для любого значения k .

Формат выходного файла

В первой строке выведите одно число – минимальное количество операций. Далее выведите необходимые операции в следующем формате:

- сдвиг влево Ak на t бит: “ $A \ll t$ ”,
- сложение Ak и Bk : “ $A+B$ ”,
- вычитание Ak от Bk : “ $B-A$ ”.

Пример

стандартный ввод	стандартный вывод
1 29	4 1<<1 1<<5 1+2 32-3
3 3 5 18	5 1<<1 1<<4 1+2 2+3 2+16

Задача Н. ПУТИ В ГРАФЕ

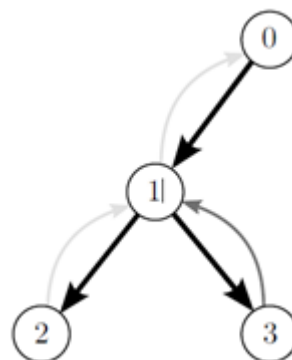
Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайт

Дан ориентированный граф, полученный из дерева (связного неориентированного графа без циклов) заменой ребер на дуги (то есть указывается направление для каждого ребра).

Вам необходимо добавить минимальное количество особых путей так, чтобы из каждой вершины по дугам графа и дугам особых путей были достижимы все остальные вершины. Особым назовем путь, который удовлетворяет условиям:

- Особый путь проходит вдоль некоторого пути заданного графа.
- В особом пути каждая дуга направлена противоположно соответствующей дуге графа.
- Каждая вершина и дуга встречаются в особом пути не более чем один раз.
- Различные особые пути могут иметь общие вершины и дуги.

На рисунке, который соответствует примеру, черные стрелочки представляют дуги исходного графа, круги соответствуют вершинам. Нам необходимо добавить два особых пути. Например, можно построить путь $2 \rightarrow 1 \rightarrow 0$ (светло-серые стрелочки) и $3 \rightarrow 1$ (темно-серые стрелочки). Вместо пути $3 \rightarrow 1$ можно добавить $3 \rightarrow 1 \rightarrow 0$. Мы не можем добавить путь типа $1 \rightarrow 3$ или $0 \rightarrow 1 \rightarrow 2$ в силу условия 2. Нельзя построить особый путь $0 \rightarrow 2$ или $2 \rightarrow 3 \rightarrow 0$ в силу условия 1.



Формат входного файла

В первой строке входного файла записано целое число N ($2 \leq N \leq 20\,000$) - количество вершин в графе. Вершины пронумерованы числами от 0 до $N-1$. Каждая из последующих $N-1$ строк содержат по два целых числа u и v ($0 \leq u, v \leq N-1$) и задает дугу в графе, которая ведет из вершины u в вершину v . Гарантируется, что если все дуги заменить ребрами (то есть убрать ориентацию), то получим дерево с N вершинами.

Формат выходного файла

Выведите единственное число – минимальное количество особых путей, которое нужно добавить, чтобы стало возможным добраться из каждой вершины графа в каждую.

Пример

стандартный ввод	стандартный вывод
4 0 1 1 2 1 3	2