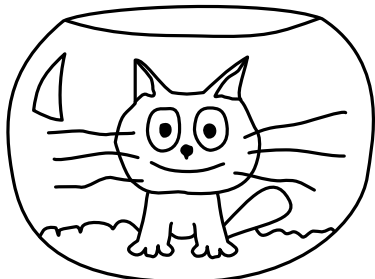


Задача Aquarium. Продавец аквариумов

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 входного файла

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

В i -ой строке j -ое 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 выходного файла

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

Пример

aquarium.in	aquarium.out
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 Barns. Коровники

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

В селе Максоярославке коровы обычно пасутся на лужайках, соединённых дорожками, на каждой лужайке пасётся хотя бы одна корова. При этом для каждой пары лужаек есть ровно один способ пройти от одной лужайки до другой. По каждой дорожке можно двигаться в обоих направлениях. Считается, что все дорожки имеют одинаковую длину.

Главный фермер села хочет построить на лужайках k коровников для своих коров. Ясно, что каждая корова вечером будет возвращаться именно в тот коровник, который ближе к её лужайке (если расстояние до коровников одинаково, то в любой из них). Поэтому возникает задача определения такого расположения коровников, при котором наибольшее из расстояний, проходимых коровами, было бы минимально.

Формат входного файла

В первой строке входного файла содержатся два числа n и k ($2 \leq n \leq 50\,000$, $1 \leq k \leq n$) — количество лужаек и планируемое число коровников, соответственно. Следующие $n - 1$ строк содержат описания дорожек. Каждая дорожка задаётся парой целых положительных чисел (a, b) , где a и b —

номера лужаек, которые соединяет данная дорожка. Лужайки нумеруются с единицы.

Формат выходного файла

В первой строке входного файла выведите l — максимальное количество дорожек, по которым придётся пройти корове, чтобы попасть в коровник. Во второй строке выведите k различных целых чисел — номера лужаек, на которых следует построить коровники. Если оптимальных решений несколько, разрешается вывести любое из них.

Пример

barns.in	barns.out
7 2	2
5 4	1 4
4 3	
1 3	
2 3	
4 6	
6 7	

Задача Boolean. Логическое дерево

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

Рассмотрим разновидность двоичного дерева которую мы назовем логическим деревом. В этом дереве каждый уровень полностью заполнен, за исключением, возможно, последнего (самого глубокого) уровня. При этом, все вершины последнего уровня находятся максимально слева. Дополнительно, каждая вершина дерева имеет ноль или двоих детей.

Каждая вершина дерева имеет связанное с ней логическое значение (1 или 0). Кроме этого, каждая *внутренняя* вершина имеет связанную с ней логическую операцию (“И” или “ИЛИ”). Значение вершины с операцией “И” — это логическое “И” значений ее детей. Аналогично, значение вершины с операцией “ИЛИ” — это логическое “ИЛИ” значений ее детей. Значения всех листьев задаются во входном файле, поэтому значения всех вершин дерева могут быть найдены.

Наибольший интерес для нас представляет корень дерева. Мы хотим, чтобы он имел заданное логическое значение v , которое может отличаться от текущего. К счастью, мы можем изменять логические операции некоторых внутренних вершин (заменить “И” на “ИЛИ” и наоборот).

Дано описание логического дерева и набор вершин, операции в которых могут быть изменены. Найдите наименьшее количество вершин, которые следует изменить, чтобы корень дерева принял заданное значение v . Если это невозможно, то выведите строку “IMPOSSIBLE” (без кавычек).

Формат входного файла

В первой строке входного файла находятся два числа n и v ($1 \leq n \leq 10\,000$, $0 \leq v \leq 1$) — количество вершин в дереве и требуемое значение в корне соответственно. Поскольку все вершины имеют ноль или двоих детей, то n — нечетное. Следующие n строк описывают вершины дерева. Вершины нумеруются с 1 до n .

Первые $(n - 1)/2$ строк описывают внутренние вершины. Каждая из них содержит два числа — g и c , которые принимают значение либо 0, либо 1. Если $g = 1$, то вершина представляет логическую операцию “И”, иначе она представляет логическую операцию “ИЛИ”. Если $c = 1$, то операция в вершине может быть изменена, иначе нет. Внутренняя вершина с номером i имеет детей $2i$ и $2i + 1$.

Следующие $(n + 1)/2$ строк описывают листья. Каждая строка содержит одно число 0 или 1 — значение листа.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Пример

boolean.in	boolean.out
9 1 1 0 1 1 1 1 0 0 1 0 1 0 1	1
5 0 1 1 0 0 1 1 0	IMPOSSIBLE

Задача Network. Сеть

Имя входного файла: **network.in**
Имя выходного файла: **network.out**
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В компьютерной сети вашей фирмы n компьютеров. В последнее время свитч, к которому они подключены, сильно барахлит, и потому не любые два компьютера могут связаться друг с другом. Кроме того, если компьютер a обменивается информацией с компьютером b , то никакие другие компьютеры не могут в это время обмениваться информацией ни с a , ни с b . Вам необходимо вычислить максимальное количество компьютеров, которые могут одновременно участвовать в процессе обмена информацией.

Формат входного файла

В первой строке файла задано число n ($1 \leq n \leq 18$). Далее идут n строк по n символов, причем j -ый символ i -ой строки равен 'Y', если i -ый и j -ый компьютеры могут обмениваться информацией, иначе он равен 'N'. Верно, что i -ый символ i -ой строки всегда равен 'N', кроме того, матрица символов симметрична.

Формат выходного файла

Выведите максимальное количество компьютеров, которые могут одновременно участвовать в процессе обмена информацией.

Пример

network.in	network.out
5 NYYYY YNNNN YNNNY YNNNY YNYYN	4