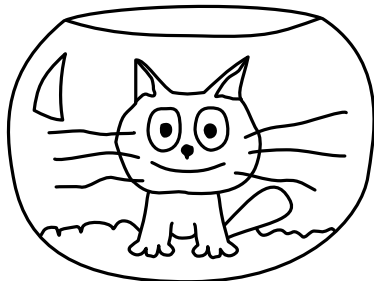


Задача А. Продавец аквариумов

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 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 выходного файла

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 В. 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 которо-

му 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, причём j -й символ i -й строки равен 'Y', если i -й и j -й компьютеры могут обмениваться информацией, иначе он равен 'N'. Верно, что i -й символ i -й строки всегда равен 'N' и, кроме того, матрица символов симметрична.

Формат выходного файла

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

Пример

<code>network.in</code>	<code>network.out</code>
5	4
NYYYY	
YNNNN	
YNNNY	
YNNNY	
YNYYN	

Задача С. 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 её детей. Аналогично, значение вершины с операцией „ИЛИ“ — это логическое „ИЛИ“ значений её 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 нечётно. Следующие n строк описывают вершины дерева. Вершины нумеруются от 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.

Примеры

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 D. Ящички

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

В недавно открытой раздевалке школы «И.» решено поставить такие же шкафчики, как и в уже давно используемых раздевалках, но более новой модификации — состоящие из $H \times W$ ячеек. Напомним, что в каждую ячейку можно поставить ящичек, чтобы хранить в нём свои вещи. Однако новый директор школы запретил ученикам хранить свои вещи вне ящичков, поэтому тем, кому ящички не достались, приходится просить кого-то из владельцев соседних четырёх (или менее, если ячейка находится на границе) ячеек похранить свои вещи у себя. Если же ни у кого из соседей по ячейкам нет ящичков, этот ученик жалуется в администрацию.

Классному руководителю вдруг стало интересно, сколько же существует способов для каждого ученика определить, давать ли ему ящичек, чтобы никто не пожаловался в администрацию.

Количество учеников равно количеству ячеек.

Формат входного файла

В единственной строке входного файла содержатся два натуральных числа H и W ($1 \leq H, W \leq 8$).

Формат выходного файла

Выведите единственное натуральное число — искомое количество способов.

Примеры

boxes.in	boxes.out
1 1	1
2 2	11

Примечание

Пояснение ко второму примеру: из шестнадцати возможных расстановок следующие пять являются неприемлемыми для владельца выделенной ячейки (за 0 обозначим отсутствие ящичка в ячейке, за 1 — его наличие):

00	00	00	01	10
00	01	10	00	00