

Задача А. Binary palindromes

Имя входного файла: `binpal.in`
Имя выходного файла: `binpal.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Назовём целое положительное число *двоичным палиндромом*, если при чтении двоичной записи числа (не содержащей ведущих нулей) справа налево получается то же самое число. Например, первыми двоичными палиндромами являются числа 1, 3, 5, 7, Ваша задача — по заданному M найти M -й по возрастанию двоичный палиндром и вывести его десятичную запись

Формат входного файла

Во входном файле задано одно целое число M ($1 \leq M \leq 5 \cdot 10^4$).

Формат выходного файла

Выведите одно число — десятичную запись M -го двоичного палиндрома

Пример

<code>binpal.in</code>	<code>binpal.out</code>
1	1
2	3
3	5
4	7

Задача В. Fold

Имя входного файла: `fold.in`
Имя выходного файла: `fold.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Прямоугольный лист бумаги складывается несколько раз по линиям сгиба, параллельным его сторонам, после чего прокалывается в некоторой точке. Ваша задача — вычислить, сколько проколов будет в листе после того, как его развернут (при этом прокол на границе листа учитывается).

Формат входного файла

В первой строке входного файла содержится целое число T — количество тестовых примеров ($1 \leq T \leq 1600$).

В первой строке каждого тестового примера заданы пять целых чисел — координаты x_h и y_h правого верхнего угла первоначального листа бумаги (левый нижний угол находится в начале координат, при этом две стороны проходят по координатным осям). Далее заданы координаты x_d и y_d точки, в которой был произведён прокол, а также количество складываний S ($0 \leq x_h, y_h, x_d, y_d \leq 10^4$, $0 \leq S \leq 8$). В следующих S строках заданы складывания (в том порядке, в котором они производятся).

Каждое складывание описывается следующим образом: если лист складывается вертикально вдоль прямой $x = N$, сначала идёт символ 'X', затем — целое число N ; если лист складывается горизонтально вдоль прямой $y = N$, то сначала идёт символ 'Y', затем — целое число N ($0 \leq N \leq 10^4$). При этом всегда в случае горизонтального складывания верхняя часть листа перегибается вниз, а в случае вертикального складывания — правая часть перегибается налево. Если соответствующая прямая на момент складывания не имеет общих точек с листом бумаги, то соответствующего складывания не производится.

Формат выходного файла

Для каждого тестового примера в отдельной строке выведите одно число — количество проколов в листе бумаги, которое получится при разворачивании листа из соответствующего тестового примера.

Пример

fold.in	fold.out
2	0
100 100 70 80 1	4
X 40	
98 26 6 10 3	
X 20	
Y 18	
X 40	

Задача C. Knights

Имя входного файла: `knights.in`
Имя выходного файла: `knights.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Компания MAD, являющаяся спонсором проекта LoopCoder, объявила о начале разработки принципиально новой игры — параллельных шахмат. В параллельных шахматах игрок может делать ход несколькими фигурами одновременно. При этом фигуры перемещаются по обычным правилам.

Вам поручено реализовать следующую задачу, относящуюся к параллельным шахматам. На доске $N \times N$, в которой «вырезано» несколько клеток, необходимо расставить коней так, чтобы одновременно все кони могли бы сделать ход. При этом фигуры могут быть поставлены (а впоследствии — сделать ход) только на невыколотые клетки, не занятые как в момент начала, так и в момент конца хода другой фигурой. Среди всех таких расстановок вам необходимо вывести произвольную расстановку, для которой количество коней на доске максимально.

Формат входного файла

В первой строке входного файла задано число N ($1 \leq N \leq 100$). Далее следует описание конфигурации поля — N строк по N символов в каждой. Строки состоят из символов '.', обозначающих клетки, оставшиеся на доске, и '*', обозначающих вырезанные клетки.

Формат выходного файла

Выведите любую отвечающую требованию задачи конфигурацию доски с уже расставленными фигурами в том же формате, что и на входе, за исключением того, что на полях, занятых конями, точка заменяется символом 'K'.

Примеры

<code>knights.in</code>	<code>knights.out</code>
3	K.K ... K.K
3 *.*	*.* ... K.K

Задача D. Numbers on circle

Имя входного файла: `numcir.in`
Имя выходного файла: `numcir.out`
Ограничение по времени: 25 seconds
Ограничение по памяти: 64 Mebibytes

По кругу расставлены N более мелких кругов, пронумерованных от 1 до N . Также заданы N целых чисел. Требуется расставить числа в мелкие круги так, чтобы сумма любых трёх идущих подряд по кругу чисел была бы простым числом, при этом из всех расстановок выбрать такую, что сумма N произведений каждого из чисел на номер круга, в котором оно размещено, была бы минимальной. Если расстановка возможна, требуется вычислить соответствующую сумму, если нет — вывести “impossible”

Формат входного файла

В первой строке входного файла задано количество тестовых примеров T ($1 \leq T \leq 666$).

Каждый тестовый пример состоит из одной строки, состоящей из целых положительных чисел. Сначала идёт число N ($3 \leq N \leq 15$) — количество мелких кругов, далее заданы N чисел, не превосходящих 100, которые надо расставить в мелкие круги.

Формат выходного файла

Для каждого тестового примера выведите ответ к задаче — требуемую минимальную сумму, если указанная в условии задачи расстановка возможна, или “impossible” в противном случае.

Пример

<code>numcir.in</code>	<code>numcir.out</code>
2	126
4 13 11 17 13	impossible
4 2 4 6 8	

Задача E. Byteland Oil

Имя входного файла: `oil.in`
Имя выходного файла: `oil.out`
Ограничение по времени: 4 seconds
Ограничение по памяти: 64 Mebibytes

Система нефтепроводов Байтландии устроена в виде двоичного дерева, рёбра которого представляют собой трубопроводы, в листьях размещены l месторождений, а в корне — нефтеперерабатывающий завод. Для каждого месторождения известно количество нефти, добываемой за сутки, в тоннах. При этом максимальная пропускная способность каждого трубопровода изначально равна одной тонне в сутки. Поток нефти, идущий в сутки через трубопровод, начинающийся в точке соединения двух других трубопроводов, равен минимуму из значения пропускной способности данного трубопровода и суммы потоков нефти, идущих в сутки по «входным» трубопроводам.

Руководство завода приняло решение инвестировать в развитие инфраструктуры N миллионов байтландских тугриков. При этом если в какой-то трубопровод вкладывается k миллионов, то его пропускная способность становится равной $(1+k)^2$ тонн в сутки, если же k миллионов вкладывается в месторождение, дававшее в сутки s тонн нефти, то оно начинает вырабатывать $s+k$ тонн нефти в сутки. При этом в связи с особенностями налогообложения естественных монополий в Байтландии вкладывать можно только целое количество миллионов.

Ваша задача — вычислить, какое максимальное количество нефти может поступать на нефтеперерабатывающий завод за сутки после реализации проекта по развитию инфраструктуры.

Формат входного файла

В первой строке входного файла содержится описание дерева. Дерево задаётся рекурсивно — каждое поддереву может быть или целым числом V_i (для месторождения — объём добываемой нефти в тоннах, $1 \leq v_i \leq 10^4$), или записанной в скобках парой $(T_l T_r)$, где T_l и T_r — отделённые друг от друга пробелом описания левого и правого поддеревьев, соответственно. Количество месторождений l положительно и не превышает 50. Во второй строке задана сумма N , выделенная на развитие инфраструктуры (в миллионах байтландских тугриков; $1 \leq N \leq 2500$).

Формат выходного файла

Выведите одно число — максимальное количество нефти, которое может поступать на нефтеперерабатывающий завод за сутки после улучшения инфраструктуры.

Пример

<code>oil.in</code>	<code>oil.out</code>
<code>(6 ((8 2) (4 5)))</code> <code>4</code>	<code>9</code>

Задача F. Shooting

Имя входного файла: `shooting.in`
Имя выходного файла: `shooting.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Во время партии в пейнтбол сложилась следующая ситуация: у всех игроков осталось по одному выстрелу. По заданной взаимной видимости игроков на поле вычислите, возможна ли ситуация, когда игроки, не смещаясь с места, стреляют друг в друга таким образом, чтобы в каждого из игроков попал ровно один выстрел. Считается, что игрок может попасть в другого игрока при выстреле в том и только том случае, если он его видит.

Формат входного файла

В первой строке входного файла заданы два целых числа N и M — соответственно количество игроков и количество пар игроков, которые видят друг друга ($2 \leq N \leq 1000$, $0 \leq M \leq 5000$). Далее следуют M пар целых чисел A_i и B_i ($1 \leq A_i < B_i \leq N$). Каждая пара содержит информацию о том, что игроки с номерами A_i и B_i видят друг друга. При этом каждая пара (A_i, B_i) встречается во входном файле не более одного раза. Игроки нумеруются с единицы.

Формат выходного файла

Если описанная в условии задача ситуация невозможна, выведите “Impossible”. В противном случае выведите N чисел, по одному числу на каждую строку. В i -й строке указывается номер игрока, в которого стреляет i -й игрок. Если решений несколько, выведите любое из них.

Пример

shooting.in	shooting.out
4 5 1 4 2 4 3 4 1 2 1 3	2 4 1 3
4 3 1 4 2 4 3 4	Impossible