

## Задача А. Логическое дерево

Имя входного файла:	boolean.in
Имя выходного файла:	boolean.out
Ограничение по времени:	0.5 секунда
Ограничение по памяти:	256 мегабайт

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 входных данных

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 выходных данных

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 В. Деловые встречи

Имя входного файла: `meetings.in`  
Имя выходного файла: `meetings.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Алексей — успешный предприниматель, и в течение одного дня у него бывает много встреч с разными деловыми партнёрами. К сожалению, встречи бывают разные и не все приносят ему радость, после других же настроение улучшается. Также, на многие встречи не стоит приходить в слишком плохом или хорошем настроении — результат таких встреч может быть не таким, какой хочется Алексею.

К счастью, недавно Алексей научился оценивать своё настроение с помощью целых чисел. После этого для каждой встречи он оценил, при каком максимальном и минимальном настроении стоит на неё приходить, а также как изменится его настроение после этой встречи. Теперь он хочет распланировать порядок встреч так, чтобы в течение дня совершить максимальное число встреч.

Ваша задача — написать программу, которая по информации о всех встречах и настроении Алексея в начале дня находит порядок проведения встреч такой, что их количество при этом максимально.

### Формат входных данных

Первая строка входного файла содержит два целых числа  $n$  и  $k$  ( $1 \leq n \leq 20$ ,  $-100 \leq k \leq 100$ ) — количество встреч и настроение Алексея в начале дня.

Следующие  $n$  строк содержат по три целых числа  $a_i$ ,  $b_i$  и  $c_i$  ( $-100 \leq a_i, b_i, c_i \leq 100$ ) — минимальное и максимальное настроение, при котором встреча возможна, и изменение настроения по окончании встречи, соответственно.

### Формат выходных данных

В первой строке выходного файла выведите число  $m$  — максимально возможно число встреч. В следующей строке выведите  $m$  целых чисел — номера встреч в порядке их проведения. Встречи пронумерованы в порядке описания во входном файле.

Если ответов с максимальным числом встреч несколько, выведите любой.

### Примеры

<code>meetings.in</code>	<code>meetings.out</code>
3 0 1 3 3 0 1 2 1 3 1	3 2 3 1
3 1 -10 -5 3 -5 5 -2 -3 2 1	2 2 3

## Задача С. Доминошки

Имя входного файла: dominoes.in  
Имя выходного файла: dominoes.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вам дано клетчатое поле размера  $n \times m$  клеток. Некоторые клетки на этом поле отмечены как занятые. На нем необходимо разместить несколько (возможно, 0) доминошек размера  $1 \times 2$ , соблюдая следующие ограничения:

- каждая клетка может быть покрыта не более чем одной доминошкой
- никакие две соседних по стороне клетки не могут быть заняты разными доминошками

Сколькими способами можно это сделать?

### Формат входных данных

В первой строке входного файла содержится два целых числа:  $n$  и  $m$  ( $1 \leq n \leq 6, 1 \leq m \leq 100$ ). В следующих  $n$  строках содержится описание поля. Символ  $X$  означает занятую клетку, символ  $.$  означает свободную клетку.

### Формат выходных данных

Выведите одно целое число — количество способов разместить доминошки, взятое по модулю  $10^9 + 7$ .

### Примеры

dominoes.in	dominoes.out
2 2 .. ..	5
3 3 ... .X. ...	21
5 5 ..... ..... ..X.. ..... .....	5182

## Задача D. Число возрастающих подпоследовательностей

Имя входного файла: `subseq.in`  
Имя выходного файла: `subseq.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Задана последовательность из  $n$  чисел  $a_1, a_2, \dots, a_n$ . Подпоследовательностью длины  $k$  этой последовательности называется набор индексов  $i_1, i_2, \dots, i_k$ , удовлетворяющий неравенствам  $1 \leq i_1 < i_2 < \dots < i_k \leq n$ . Подпоследовательность называется возрастающей, если выполняются неравенства  $a_{i_1} < a_{i_2} < \dots < a_{i_k}$ .

Необходимо найти число возрастающих подпоследовательностей наибольшей длины заданной последовательности  $a_1, \dots, a_n$ . Так как это число может быть достаточно большим, необходимо найти остаток от его деления на  $10^9 + 7$ .

### Формат входных данных

Первая строка входного файла содержит целое число  $n$  ( $1 \leq n \leq 100000$ ). Вторая строка входного файла содержит  $n$  целых чисел:  $a_1, a_2, \dots, a_n$ . Все  $a_i$  не превосходят  $10^9$  по абсолютной величине.

### Формат выходных данных

В выходной файл выведите ответ на задачу.

### Примеры

<code>subseq.in</code>	<code>subseq.out</code>
5 1 2 3 4 5	1
6 1 1 2 2 3 3	8

## Задача Е. Сбалансируй-ка!

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

Лидеру команды «Отбой» на День Рождения подарили подвешенное бинарное дерево. Однако, ему не понравилось, что дерево было несбалансированным. Теперь он хочет удалить минимальное количество вершин в дереве, чтобы оно стало сбалансированным. Перед тем как удалить вершину из дерева, он обязан удалить все вершины из её поддерева. Напомним, что дерево является сбалансированным тогда и только тогда, когда для любой вершины высота её левого и правого поддеревьев отличается не более чем на 1 (высота пустого дерева равна нулю, а высота дерева из одной вершины — единице). Корнем дерева является вершина 1.

### Формат входных данных

В первой строке входного файла задано целое число  $n$  — количество вершин в дереве ( $1 \leq n \leq 1111$ ). В следующих  $n$  строках заданы по два целых числа  $left_i$  и  $right_i$  — номера левого и правого ребёнка вершины соответственно или 0, если этого ребёнка не существует.

### Формат выходных данных

В единственной строке выходного выведите одно число — искомое минимальное количество удаляемых вершин.

### Примеры

<code>balance.in</code>	<code>balance.out</code>
6 2 3 0 0 4 5 0 6 0 0 0 0	1
3 0 2 0 3 0 0	1

## Задача F. Реки

Имя входного файла: `rivers.in`  
Имя выходного файла: `rivers.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В одном уголке земного шара раскинулась живописная страна, покрытая лесами и реками. Маленькие речушки сливаются в большие, те — в ещё большие, и в конце концов все реки сливаются в одну большую реку, впадающую в море неподалёку от городка.

А ещё в этой стране есть  $n$  поселений лесорубов. Каждое поселение расположено около какой-то реки. Лесорубы валят лес, а затем сплавляют его по воде до городка. В городке находится лесопилка.

Чтобы сократить расходы на транспортировку леса по воде, решено было выбрать  $k$  поселений лесорубов и построить дополнительные лесопилки прямо у этих поселений. Из остальных поселений лес будет сплавляться вниз по течению до первой же лесопилки, будь то старая лесопилка в городке или одна из  $k$  новых. Лес, обработанный в лесопилке, сплавлять дальше по воде не требуется.

Ни одна река не разветвляется, так что из любого поселения существует единственный путь вниз по течению до городка.

Для каждого поселения лесорубов известно, сколько деревьев валят жители этого поселения за год. Требуется поставить новые лесопилки так, чтобы ежегодные расходы на сплав леса оказались как можно меньше. Известно, что транспортировка одного дерева на один километр по любому отрезку речной сети обходится в одну монету. Напишите программу, которая по заданному описанию речной сети, а также числу  $k$  вычисляет, каковы будут минимальные ежегодные расходы на сплав леса в монетах.

### Формат входных данных

В первой строке ввода содержится два целых числа  $n$  и  $k$  — количество поселений лесорубов и количество новых лесопилок, которые можно построить ( $2 \leq n \leq 100, k \leq 50, k \leq n$ ). Следующие  $n$  строк описывают деревни;  $i$ -я из этих строк содержит три целых числа  $w_i, v_i$  и  $d_i$  — количество деревьев, ежегодно срубаемых жителями  $i$ -го поселения, номер ближайшего населённого пункта вниз по течению и расстояние до него ( $0 \leq w_i \leq 10000, 0 \leq v_i \leq n, 1 \leq d_i \leq 10000$ ). Поселения пронумерованы целыми числами, начиная с единицы, в том порядке, в котором они заданы во входных данных, городок обозначается числом 0.

Гарантируется, что до постройки дополнительных лесопилок ежегодные расходы на сплав составляли не более  $2 \cdot 10^9$  монет.

### Формат выходных данных

В первой строке выведите одно число — минимальные ежегодные расходы на сплав леса в монетах.

### Примеры

<code>rivers.in</code>	<code>rivers.out</code>
4 2 1 0 1 1 1 10 10 2 5 1 2 3	4

## Задача G. Передай другу

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

ЛКШ 2011 будет проходить в Рио де Жанейро, где все ходят с белыми бейджиками. Вы не знали этого? А ведь именно такого рода слухи распространяет среди своих друзей лкшонок Остап.

У Остапа есть несколько разных слухов. Для каждого типа слуха он выбирает друзей, которым хочет его рассказать напрямую. Сообщая слух другу, Остап также просит его передать этот слух всем его друзьям. Однако те уже не будут передавать этот слух далее. Например, если Аня и Ваня — друзья, Ваня и Саня — друзья, а Аня и Саня — не друзья, и Остап сообщает какой-то слух Ане, то новость дойдет до Вани, но не до Сани.

Каждому человеку Остап может сообщить напрямую не более одного слуха (при этом если человек получил несколько разных типов слухов через друзей, ничего страшного нет). Какое наибольшее число различных типов слухов Остап может распространить так, чтобы каждый из них дошел до каждого его друга?

### Формат входных данных

Во входном файле находится квадратная таблица  $n \times n$ , где  $n \leq 15$  — число друзей Остапа. В  $i$ -й строке в  $j$ -м столбце содержится символ Y, если  $i$ -й и  $j$ -й друзья Остапа дружат между собой, и N — иначе.

### Формат выходных данных

Выведите наибольшее количество слухов, которые может распространить Остап среди всех своих друзей.

### Примеры

friends.in	friends.out
NYYN YNY YNY NYYN	3
NYYN YNY YNN NNNN	1
NYNNY YNYNN NYNYN NNYNY NNNY YNNY	3

## Задача Н. Задобрите Инти

Имя входного файла: `makeintihappy.in`  
Имя выходного файла: `makeintihappy.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Как вы помните, племя Майя вымерло, племя Йюля вынуждено было распасться и покинуть обжитые территории, и вот теперь племени Авгуйя тоже угрожает опасность. Задоблив бога дождя Тлалока, они напрочь забыли про бога солнца Инти, который будучи обделённым вниманием грозиться ниспослать испепеляющую жару, если жители племени не порадуют его божественный взор новой коллекцией симпатичных узоров, выложенных из огромных гранитных плит, покрашенных в чёрный и белый цвета. Напомним, что симпатичным узором называется прямоугольник шириной  $W$  и высотой  $H$  ни какой квадрат  $2$  на  $2$  которого не покрашен в один цвет.

В данной задаче таскать гранитные плиты Вам не придётся, однако потребуется определить  $K$ -й лексикографически симпатичный узор. Один узор будем считать лексикографически меньше другого, если при просмотре узоров по столбцам сверху вниз найдётся такая позиция, что все плиты рассмотренные ранее совпадают попарно по цвету, а плита на данной позиции у первого узора белая, а у второго — чёрная.

### Формат входных данных

Во входном файле заданы числа  $W$ ,  $H$  и  $K$ . ( $1 \leq W \leq 1000$ ,  $1 \leq H \leq 10$ ,  $1 \leq K \leq 10^{18}$ )

### Формат выходных данных

Выведите требуемый симпатичный узор, обозначая белые плиты символом  $'w'$ , а чёрные — символом  $'b'$ . Если не существует узора с таким номером — выведите "Impossible" (без кавычек).

### Примеры

<code>makeintihappy.in</code>	<code>makeintihappy.out</code>
5 1 17	bwww
7 4 1234567	wbwbwb wbbwwb wwwbbwb wbwbwb