

Задача А. График патрулирования

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

Разведгруппа получила задание проникнуть на территорию крупного железнодорожного узла, занятого противником, и добыть информацию о транспортных потоках, проходящих через эту станцию и о возможностях организации диверсий.

Командир разведгруппы поставил задачу — в первую очередь выяснить, какие именно участки железной дороги патрулируются, а какие нет. Результаты наблюдений были получены от местного населения. Каждый наблюдатель сообщал начальное и конечное расстояние от станции известной ему зоны патрулирования противника. Зоны начинались и заканчивались на целом расстоянии от станции. Командир решил упорядочить поступившую информацию.

От вас требуется вывести в порядке возрастания список непересекающихся отрезков пути, которые патрулируются противником.

Формат входного файла

В первой строке входного файла содержится количество тестовых примеров T , $1 \leq T \leq 20$.

Далее следуют T тестовых примеров. В первой строке каждого примера задано целое число N ($1 < N \leq 20000$) — общее количество сообщений от наблюдателей. В следующих N строках идут пары целых чисел a_i и b_i — расстояние вдоль железной дороги от станции соответственно до начала и до конца i -й зоны патрулирования ($0 < a_i \leq b_i < 10^9$).

Формат выходного файла

Для каждого тестового примера выведите в отдельных строках список непересекающихся отрезков пути, которые патрулируются противником. Отрезок выводится в формате $A_i B_i$, где A_i — расстояние от станции до начала отрезка, B_i — расстояние от станции до конца отрезка. Каждый отрезок выводить в отдельной строке, отрезки упорядочить по возрастанию A_i .

Пример

patrol.in	patrol.out
2	34 3123
3	1 8
34 1023	156 4567
2045 3123	
1024 2187	
3	
234 4567	
1 8	
156 33	

Задача В. Уровень глубины

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

Анализ поступившей от наблюдателей информации показал, что наилучшим местом для диверсии будет неохраняемый участок железной дороги, проходящий вдоль довольно глубокого болота.

Болото было обширным и довольно глубоким. Его можно было бы назвать непроходимым, если бы не существовавший с давних пор путь, проходящий по неглубоким местам. У разведчиков имелись с собой результаты аэрофотосъёмки, представленные в виде карты $W \times K$ пикселей. Каждый единичный пиксел A_{ij} этой карты имеет яркость p_{ij} , заданную неотрицательным целым числом. При этом было известно, что для «проходимого» участка болота верны следующие утверждения.

Во-первых, для любых двух пикселей, принадлежащих данному участку, существует путь, соединяющий эти два пикселя и проходящий по данному участку. При этом пары соседних пикселей в этом пути обязательно граничат по стороне.

Во-вторых, разность яркостей любых двух пикселей, принадлежащих этому участку, не превосходит заданного порога P .

По заданной Вам карте необходимо вычислить площадь максимального участка, обладающего такими свойствами.

Формат входного файла

В первой строке входного файла содержится количество тестовых примеров T , $1 \leq T \leq 20$.

Далее следуют T тестовых примеров. В первой строке каждого примера задаётся одно целое число — порог яркости P ($0 \leq P \leq 10$). В следующей строке заданы два целых числа K и W — размеры карты в пикселях ($1 \leq K, W \leq 300$). Далее следуют W строк по K целых чисел p_{ij} в каждом — яркости соответствующих пикселей. При этом $0 \leq p_{ij} \leq 10^9$.

Формат выходного файла

Для каждого тестового примера в отдельной строке выведите площадь в пикселях максимального участка, обладающего описанными в условии свойствами.

Пример

level.in	level.out
2	3
4	3
2 2	
5 2	
12 1	
2	
2 3	
24 28 25	
25 23 27	

Задача С. Эшелон с техникой

Имя входного файла: `train.in`
Имя выходного файла: `train.out`
Ограничение по времени: 6 секунд
Ограничение по памяти: 64 Мб

Оставалось выяснить расписание прохождения эшелонов со стратегическими грузами через соответствующий участок железной дороги. Одному из разведчиков удалось проникнуть в кабинет дежурного по станции в момент, когда тот был вызван к начальству.

Разведчик осмотрел помещение. Похоже, что удалось застать противника врасплох: на столе лежит какой-то справочник и написанный «от руки» текст. В тексте сообщалось, что прибывает эшелон с боевой техникой, общий код эшелона... далее следовало некоторое число. В справочнике были обнаружены коды различных грузов военного назначения, в том числе и тяжёлой боевой техники.

Судя по вычислениям на обнаруженных в корзине черновиках, система кодирования выглядела следующим образом: код эшелона был равен сумме кодов всех грузов, которые этот эшелон доставляет. Для данного эшелона с вооружениями код эшелона, очевидно, был равен $a_1b_1 + a_2b_2 + \dots + a_nb_n$, где a_i — количество техники i -го типа и b_i — код i -го типа техники. Зная значения всех b_n по справочнику, а также зная, что все a_i не превосходят 9, найдите количество каждого из видов техники в эшелоне.

Формат входного файла

В первой строке входного файла содержится количество тестовых примеров T , $1 \leq T \leq 20$.

Далее следуют T тестовых примеров. В первой строке каждого примера задано целое число M , $1 \leq M \leq 12$ — количество типов тяжёлой боевой техники. В следующей строке идут M целых чисел b_i , $0 \leq b_i \leq 3 \times 10^7$ — коды соответствующего типа боевой техники. В последней строке тестового примера задано целое число K , $0 \leq K \leq 3 \times 10^9$ — общий код эшелона.

Формат выходного файла

Для каждого тестового примера в отдельной строке выведите возможное количество техники каждого типа — M целых чисел от 0 до 9 (в порядке задания кодов во входном файле). Наличие хотя бы одного решения гарантируется. Если решений несколько, выведите лексикографически наименьшее.

Пример

train.in	train.out
2	2 5
2	0 0 2
43 84	
506	
3	
1 1 2	
4	

Задача D. Сопротивления

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

Атаковав один из разъездов, находящихся вблизи станции, разведгруппа захватила важное устройство — радиопередатчик, позволяющий передавать и принимать зашифрованные радиogramм. Устройство было довольно тяжёлым, так что перетащить его «к своим» было затруднительно, да и особого смысла это не имело: противнику будет ясно, что аппарат захвачен, и система шифрования будет сменена. Присутствующий в разведгруппе квалифицированный радист решил перерисовать схему устройства — чтобы потом, уже вернувшись на базу, попробовать воспроизвести аналогичное.

Одна из плат, на которых было собрано устройство, представляла собой электрическую цепь, состоящую из нескольких узлов, между которыми впаяны резисторы. Каждый резистор припаян к двум различным узлам цепи, при этом к каждому узлу припаян хотя бы один резистор. Кроме того, сопротивление между двумя любыми узлами сети конечно (то есть цепь не является «разорванной»). Для проверки того, насколько корректно перерисована схема, радисту необходимо быстро вычислить сопротивление между двумя любыми узлами схемы (чтобы сравнить его с сопротивлением, полученным в результате измерений).

По заданной схеме цепи и значениям сопротивлений резисторов вычислить сопротивление между двумя заданными узлами, при этом сопротивление элементов цепи, соединяющих резисторы, пренебрежимо мало.

Формат входного файла

В первой строке входного файла содержится количество тестовых примеров T , $1 \leq T \leq 20$.

Далее следуют T тестовых примеров. В первой строке тестового примера заданы два целых числа W и O ($1 < W < 100$, $1 \leq O \leq 50$) — количество узлов сети и количество резисторов соответственно.

В следующих O строках заданы описания резисторов. Каждый резистор задаётся тремя целыми числами: a_i , b_i — номера узлов, между которыми впаян резистор, r_i — сопротивление, $100 \leq r_i \leq 10000$. Узлы занумерованы, начиная с единицы. В последней строке примера заданы два различных целых числа K и M — номера узлов, сопротивление участка цепи между которыми необходимо определить.

Формат выходного файла

Для каждого тестового примера выведите целую часть значения сопротивления между заданными в этом примере узлами.

Пример

<code>resist.in</code>	<code>resist.out</code>
2	523
3 3	290
1 2 800	
2 3 300	
1 3 1000	
1 3	
4 5	
1 2 100	
3 2 300	
1 3 200	
2 4 400	
4 3 500	
1 4	

Задача Е. Схема

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

Логическая схема шифрующей части устройства была построена на следующей элементной базе: **OR** (логическое «или»), **NOR** (отрицание «или»), **AND** (логическое «и»), **NAND** (отрицание «и»), **XOR** (исключающее «или»), **XNOR** (отрицание исключающего «или») и **NOT** (отрицание). В первых 4 элементах на вход может быть подано от 2 до 10 бит, в **XOR** и **XNOR** — ровно 2, в **NOT** — ровно один бит.

При этом в схеме нет «петель», а на каждый из входов единичного элемента подаётся или результат работы другого элемента, или один из битов входных данных.

Требуется по заданной схеме и набору входных данных найти результат, который будет получен на выходе данной схемы.

Формат входного файла

В первой строке входного файла содержится количество тестовых примеров T , $1 \leq T \leq 20$.

Далее следуют T тестовых примеров. В первой строке каждого примера содержится два целых числа $1 \leq W \leq 1000$ и $1 \leq B \leq 20000$ — соответственно количеств входных бит схемы и количество элементов. В следующей строке заданы W чисел 0 или 1 — на i -м месте находится значение входного бита на i -м входе.

Далее идёт B строк — описания элементов. Каждый элемент задаётся следующим образом. Сначала идёт число w_i — количество входов элемента. Затем идут описания входов — w_i целых чисел. Положительное число K , не превосходящее B , обозначает, что на данный вход подаётся результат работы K -го элемента, отрицательное число $-L$, по модулю не превосходящее W , обозначает, что на данный вход подаётся L -й в порядке перечисления входной бит. Входные биты, как и элементы цепи, занумерованы, начиная с единицы.

Завершает строку тип элемента — одно из слов **OR**, **NOR**, **AND**, **NAND**, **XOR**, **XNOR**, **NOT**.

В последней строке тестового примера задан номер элемента, результат работы которого подаётся на выход схемы.

Формат выходного файла

Для каждого тестового примера выведите число 0 или 1 — результат работы заданной в примере схемы.

Пример

<code>scheme.in</code>	<code>scheme.out</code>
1 3 3 1 1 0 2 -1 -2 XNOR 3 -1 1 3 AND 2 1 -3 OR 2	1

Задача F. Чётность

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

После возвращения разведгруппы на базу радиоинженеры по сделанным разведчиками описаниям восстановили шифрующее устройство, получив возможность чтения информации, передаваемой противником.

Один из элементов устройства, проверяющий «контрольную сумму» сообщения, по заданным 2 числам в двоичной системе m и n выдавал последнюю цифру двоичной записи числа сочетаний по n элементов из m .

Промоделируйте работу этого элемента.

Формат входного файла

В первой строке входного файла содержится количество тестовых примеров T , $1 \leq T \leq 20$.

Далее следуют T тестовых примеров. В первой строке каждого примера задана двоичная запись целого числа m , $1 \leq m \leq 2^{30} - 1$, во второй — двоичная запись целого числа n , $0 \leq n \leq m$.

Формат выходного файла

Для каждого тестового примера в соответствующей строке выведите последнюю цифру двоичной записи числа сочетаний по n элементов из m .

Пример

<code>parity.in</code>	<code>parity.out</code>
2	1
101	0
1	
100	
11	