

Задача А. Island. Островные государства

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Суровые феодальные времена переживала некогда великая островная страна Байтландия. За главенство над всем островом борются два самых сильных барона. Таким образом, каждый город страны контролируется одним из правителей. Как водится издревле, некоторые из городов соединены двусторонними дорогами. Бароны очень не любят друг друга и стараются делать как можно больше пакостей. В частности, теперь для того чтобы пройти по дороге, соединяющей города различных правителей, надо заплатить пошлину — один байтландский рубль.

Программист Вася живет в городе номер 1. С наступлением лета он собирается съездить в город N на Всебайтландское собрание программистов. Разумеется, он хочет затратить при этом как можно меньше денег и помочь ему здесь, как обычно, предлагается Вам.

Формат входного файла

В первой строке входного файла записано два числа N и M ($1 \leq N, M \leq 100\,000$) — количество городов и количество дорог соответственно.

В следующей строке содержится информация о городах — N чисел 1 или 2 — какому из баронов принадлежит соответствующий город.

В последних M строках записаны пары $1 \leq a, b \leq N$, $a \neq b$. Каждая пара означает наличие дороги из города a в город b . По дорогам Байтландии можно двигаться в любом направлении.

Формат выходного файла

Если искомого пути не существует, выведите единственное слово **impossible**. В противном случае в первой строке напишите минимальную стоимость и количество посещенных городов, а во вторую выведите эти города в порядке посещения. Если минимальных путей несколько, выведите любой.

Пример

стандартный поток ввода	стандартный поток вывода
7 8 1 1 1 1 2 2 1 1 2 2 5 2 3 5 4 4 3 4 7 1 6 6 7	0 5 1 2 3 4 7
5 5 1 2 1 1 2 1 2 2 3 3 5 1 4 4 5	1 3 1 4 5

Задача В. Игры на графе

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Противник начал e2–e4. Я
проанализировал его архитектуру и сдался

Из мемуаров 20-го чемпиона мира Фрица
Рыбкина

Прибыв на место, Ааз тут же потребовал организовать совещание букмекеров, на котором он изложит свой план.

— Главная задача, — начал Ааз своё выступление перед букмекерами, — научиться использовать достижения прогресса. Мы планируем запуск множества новых видов соревнований, что — вполне возможно — приведёт к тому, что появятся какие-то игры по правилам, придуманным не нами. А значит, необходимо уметь быстро выяснять, насколько эти правила могут быть нам полезны.

— А можно ли хотя бы в общем пояснить, как это будет делаться? — последовал вопрос из зала.

— Вот пример задачи, решив которую, мы сможем разобраться с целым классом игр. Дан ориентированный граф некоторой игры для двух игроков и начальная позиция в ней. Напомним, что в игре на графе игрок имеет право переходить из позиции в любую позицию, в которую есть ребро из текущей. Игроки ходят по очереди; проигрывает тот, кто не может сделать ход. Требуется проверить, верно ли, что при любой игре сторон всегда выигрывает первый игрок.

Формат входного файла

Во входном файле содержится описание одного или нескольких тестов. В первой строке каждого теста заданы число вершин V и число рёбер E ($1 \leq V \leq 100\,000$, $1 \leq E \leq 100\,000$), а также номер начальной позиции a ($1 \leq a \leq V$). Далее следуют E строк — описания рёбер в формате $u_i v_i$ ($1 \leq u_i, v_i \leq V$), что означает наличие ребра, направленного из вершины u_i в вершину v_i . Файл завершается тремя нулями. Сумма всех E по всем тестам не превосходит 100 000, количество тестов в файле не превосходит 1000.

Формат выходного файла

Следуйте формату примера максимально точно — проверка производится автоматически.

Пример

стандартный поток ввода	
3 2 1	
1 2	
1 3	
1 1 1	
1 1	
4 3 1	
1 2	
1 3	
3 4	
0 0 0	
стандартный поток вывода	
First player always wins in game 1.	
Players can avoid first player winning in game 2.	
Players can avoid first player winning in game 3.	

Задача С. Экскурсия

Ограничение по времени: 0.5 секунды
Ограничение по памяти: 256 мегабайт

ЛКШата собираются на экскурсию в Кострому. Кострома — красивый старый город, в котором площади соединяются друг с другом короткими улицами, каждую из которых можно пройти не более чем за десять минут. ЛКШата хотят составить интересный маршрут экскурсии. Так как они поедут на автобусах, то маршрут экскурсии должен начинаться и заканчиваться на одной и той же площади. К сожалению, у ЛКШат будет очень мало времени. Поэтому они решили выбрать наиболее короткий кольцевой маршрут, не проходящий ни по какой улице дважды.

Помогите ЛКШатам найти такой маршрут.

Формат входного файла

Первая строка входного файла содержит два натуральных числа n и m — количество площадей и улиц в Костроме ($1 \leq n \leq 1\,000$, $1 \leq m \leq 10\,000$). Площади занумерованы от 1 до n .

Последующие m строк содержат описания улиц. Каждая улица описывается тремя целыми числами — номерами площадей, которые она соединяет, и количеством минут, которые требуются ЛКШатам на то, чтобы пройти по ней (от одной до десяти минут). Между двумя площадями может быть более одной улицы. Улица соединяет две различные площади.

Гарантируется, что в Костроме существует как минимум один кольцевой маршрут.

Формат выходного файла

Первая строка выходного файла должна содержать единственное число — продолжительность минимального маршрута в минутах.

Пример

стандартный поток ввода	стандартный поток вывода
5 6 1 2 1 2 3 10 1 3 1 2 4 1 3 4 1 1 5 1	4

Задача D. Японский компьютер

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Как известно, для обороны границ японские инженеры разрабатывают огромных боевых человекоподобных роботов. Каждый такой робот управляется японским компьютером. Понятно, что для повышения эффективности робота программа в компьютере должна быть как можно более оптимальной, чтобы компьютер мог выполнять как можно больше программ за как можно меньшее время.

На данный момент японским программистам задали следующую задачу (её смысл секретен, поэтому здесь его описывать нельзя): изначально в памяти компьютера находится единственное число x . Требуется получить в его памяти следующие числа: a_1x, a_2x, \dots, a_nx . При этом компьютер может выполнять следующие операции:

1. Сложение двух чисел
2. Вычитание двух чисел
3. Побитовый сдвиг влево (сдвиг на k бит эквивалентен умножению на 2^k)

Все полученные промежуточные значения сохраняются в памяти, так что ими можно пользоваться при вычислении других значений.

При вычислениях никогда не должно получаться значение большее, чем $42x$. Гарантируется, что при выполнении этого ограничения, в компьютере не происходит переполнений. Также, компьютер не может работать с отрицательными числами, так что вычитать большее число из меньшего также запрещено.

Порядок, в котором в памяти будут появляться числа a_1x, a_2x, \dots, a_nx , не имеет значения.

Формат входного файла

В первой строке находится число n — количество требуемых значений ($1 \leq n \leq 41$). Во второй строке находится n чисел a_i ($2 \leq a_i \leq 42$). Все a_i различны. Само число x вам не дано, так что ваша последовательность операций должна быть верной для любого x .

Формат выходного файла

В первой строке выведите единственное число — минимальное количество требуемых операций. Далее выведите требуемые операции в следующем формате:

1. Сдвиг влево ax на k бит: “ $a<<k$ ”
2. Сложение ax и bx : “ $a+b$ ”
3. Вычитание ax из bx : “ $b-a$ ”

Запись операций не должна содержать пробелов.

Примеры

стандартный поток ввода	стандартный поток вывода
3 3 5 18	5 1<<1 1<<4 1+2 2+3 2+16
1 29	4 1<<1 1<<5 1+2 32-3
4 12 19 41 42	8 1<<1 1<<4 1+2 3<<2 3+16 19<<1 3+38 1+41

Задача Е. Две инструкции

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Возможно, самой важной инструкцией процессора 8086 является команда “**pop sp**”, которая берёт значение указателя стека с вершины стека.

В дополнение к этой команде в процессоре 80486 была добавлена другая очень важная инструкция для работы с регистром указателя стека — “**bswap sp**”. Она меняет местами старший и младший байт указателя.

- “**bswap sp**” имеет код инструкции 0F CC;
- “**pop sp**” имеет код инструкции 5C;
- сегмент памяти имеет длину 65536 байт;
- Память считается зацикленной — после ячейки с адресом FFFF идет ячейка с адресом 0000 (все адреса и коды даны в шестнадцатеричной системе);
- “**pop sp**” берет значение двухбайтового слова по адресу, на который указывает **sp**, и загружает его в **sp** (младший байт идет первым).

Вам дано начальное значение регистра **sp** и содержимое памяти. Постройте самую короткую программу, которая после своего завершения получит в **sp** заданное конечное значение. Программа должна состоять только из операций “**pop sp**” и “**bswap sp**”.

Формат входного файла

Первая строка содержит два шестнадцатитрибитных числа в пределах от 0000 до FFFF — начальное и конечное значение регистра **sp**. Следующие 4096 строк содержат дампы сегмента памяти.

Дамп задается в формате [8 байт][2 пробела][8 байт], где [8 байт] - это 8 значений последовательных байтов, разделенных одним пробелом. Байт задается как шестнадцатеричное число в пределах от 00 до FF, имеющее ровно две цифры.

Формат выходного файла

В первой строке выведите число байтов кода в кратчайшей программе в десятичной системе счисления. Начиная со второй строки, выведите дампы кода получившейся программы в том же формате, что и во вводе. Последняя (возможно, незавершенная) строка дампа должна быть обязательно завершена переводом строки без лишних пробелов сразу же после последнего выведенного байта.

Если существует несколько кратчайших программ, выведите лексикографически наименьшую.

Если невозможно решить задачу для заданного дампа и значений **sp**, выведите единственную строку со словом **IMPOSSIBLE**.

Пример

стандартный поток ввода	
0003 00EF	
00 00 EF 00 02 00 00 00 00 00 00 00 00 00 00 00	
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
<i>предыдущая строка повторяется ещё 4094 раза</i>	
стандартный поток вывода	
4	
5C 0F CC 5C	

Задача F. Ретроанализ для маленьких

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дан ориентированный весёлый граф из n вершин и m ребер. Оля и Коля в игру. Изначально фишка стоит в вершине i . За ход можно передвинуть фишку по любому из исходящих ребер. Тот, кто не может сделать ход, проигрывает. Ваша задача — для каждой вершины i определить, кто выиграет при оптимальной игре обоих.

Формат входного файла

Входные данные состоят из одного или нескольких тестов. Каждый тест содержит описание весёлого ориентированного графа. Граф описывается так: на первой строке заданы два целых числа n ($1 \leq n \leq 300\,000$) и m ($1 \leq m \leq 300\,000$). Следующие m строк содержат ребра графа, каждое описывается парой целых чисел от 1 до n . Пара $a\ b$ обозначает, что ребро ведет из вершины a в вершину b . В графе могут быть петли, могут быть кратные ребра. Сумма n по всем тестам не превосходит 300 000, сумма m по всем тестам также не превосходит 300 000.

Формат выходного файла

Для каждого теста выведите для каждой вершины **FIRST**, **SECOND** или **DRAW** в зависимости от того, кто выиграет при оптимальной игре из этой вершины. Ответы к тестам разделяйте пустой строкой.

Примеры

стандартный поток ввода	стандартный поток вывода
5 5	DRAW
1 2	DRAW
2 3	DRAW
3 1	FIRST
1 4	SECOND
4 5	
2 1	FIRST
1 2	SECOND
4 4	
1 2	FIRST
2 3	FIRST
3 1	SECOND
1 4	SECOND

Задача G. Странная игра

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Двое играют в простую игру на доске $n \times n$. У первого игрока есть одна белая фишка, а у второго — одна чёрная. Игроки ходят по очереди, первым ходит первый игрок (белые).

Первый игрок имеет право двигать свою фишку на одну клетку в одном из четырёх основных направлений (влево, вправо, вверх, вниз). Второй игрок при своем ходе также выбирает одно из этих четырёх направлений, но может передвинуть свою фишку как на одну клетку в этом направлении, так и на две. Выигрывает тот, кто первым съедает фишку соперника.

Определите победителя и число ходов, требуемое для победы, при оптимальной игре сторон.

Формат входного файла

Во входном файле даны пять чисел — n ($2 \leq n \leq 20$), а также координаты белой и чёрной фишек. Фишки стоят в разных клетках доски.

Формат выходного файла

Выведите **WHITE** x , если выигрывают белые, **BLACK** x , если выигрывают чёрные, **DRAW**, если игра закончится вничью. Здесь x — число ходов обеих сторон (полуходов) до момента окончания игры.

Пример

стандартный поток ввода	стандартный поток вывода
2 1 1 2 2	BLACK 2
2 2 2 1 2	WHITE 1
3 1 1 3 3	BLACK 6

Задача Н. Королева и король

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Сегодня королева и король встретились на шахматной доске $2 \times n$, на которой между некоторыми полями расставлены стенки. Они ходят как соответствующие шахматные фигуры (ферзь и король), но не имеют права вставать под бой друг другу (т.е. фигура не может совершить такой ход, что противник может сходить на клетку, которая будет содержать эту фигуру после ее хода; но можно делать ход, при котором делающая его фигура нападает на фигуру соперника, а фигура соперника не имеет хода на клетку нападающей фигуры). Ваша задача — определить победителя и число ходов обеих фигур, требующееся для выигрыша при оптимальной игре обеих сторон. Проигравшей считается та сторона, которая не может сделать ход.

Перед началом игры король находится на клетке $(1, 1)$, а королева — на $(2, n)$. Король ходит первым.

Фигуры не имеют права ходить через стенки. Диагональные ходы запрещены в случае, если хотя бы одна из возможных стенок, касающихся соответствующего угла, присутствует на поле.

Формат входного файла

В первой строке входного файла содержатся два целых числа n и m ($2 \leq n \leq 50$, $m \leq 300$). Далее следуют m строк, каждая из которых содержит четыре целых числа x_1, y_1, x_2, y_2 — координаты соседних полей, разделенных стенкой ($1 \leq x_i \leq 2$, $1 \leq y_i \leq n$). Стенки бывают только горизонтальными и вертикальными. Одна стенка может встречаться во входном файле более одного раза.

Формат выходного файла

Если выигрывает король, выведите в первой строке текст **king wins**, если же королева — **queen wins**. Во второй строке должно быть выведено минимальное число ходов, требующееся для победы (считаются ходы обеих сторон).

В случае ничьей выходной файл должен состоять из единственной строки **draw**.

Пример

стандартный поток ввода	стандартный поток вывода
2 1 1 2 2 2	queen wins 2
2 2 1 2 2 2 2 2 2 1	king wins 1
2 2 1 2 2 2 1 1 2 1	draw

Задача I. Без мата не обошлось. . .

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам дана шахматная доска размера 6×6 , на которой стоят три шахматные фигуры: белый король, белая ладья и чёрный король. Ваша задача — рассчитать минимальное число ходов, требуемое белым для того, чтобы заматовать чёрного короля, или определить, что это невозможно, либо что позиция является некорректной.

Формат входного файла

Во входном файле одна строка, в которой записаны три координаты полей — координаты белого короля, белой ладьи и чёрного короля соответственно. После координат через пробел написан идентификатор стороны, которая делает первый ход ('W', если белые, или 'B', если чёрные).

Формат выходного файла

Выведите в выходной файл общее количество ходов обеих сторон, необходимое для белых, чтобы выиграть игру. Если чёрный король заматован, выведите 0. Если входная позиция некорректна, выведите -1. Если игра закончится вничью (например, на доске пат), выведите -2.

Пример

стандартный поток ввода	стандартный поток вывода
c6 f4 a5 B	2
c6 f4 b5 W	-1

Задача J. Игра с клавиатурой

Ограничение по времени: 30 секунд
Ограничение по памяти: 256 Мебибайт

Маша и Миша играют в интересную игру. Каждый из них выписывает набор слов, на чём первая стадия игры заканчивается. Во второй стадии игры участники берут клавиатуру и по очереди выламывают из неё клавиши, соответствующие латинским буквам, всего l штук. После этого игроки считают количество своих слов, которые можно напечатать с помощью оставшихся клавиш. После этого тот, чьих слов остаётся меньше, проигрывает сопернику количество конфет, равное разности количеств оставшихся слов.

Первая стадия игры уже завершена, первый ход по жребию (или по мишиной галантности) предстоит Маше. Определите, сколько конфет она может себе гарантировать при оптимальной игре обоих.

Формат входного файла

В первой строке входного файла записано число l — количество клавиш, которые будут выломаны за всю игру ($1 \leq l \leq 26$). Далее записаны наборы машинных и мишинных слов в следующем формате: на одной строке количество слов, на следующей — сами слова, разделённые пробелами.

Все слова состоят из строчных букв латинского алфавита. Каждый игрок выписал не более 15 непустых слов, состоящих из не более, чем 30 букв каждое.

Формат выходного файла

Выведите единственное целое число — выигрыш Маши при оптимальной игре. Если Маша вынуждена проиграть, выведите её минимальный проигрыш со знаком «минус».

Пример

стандартный поток ввода	стандартный поток вывода
2	1
5	
abacaba a zxxzyz trava abc	
1	
a	

Задача К. Игра с клавиатурой-2

Ограничение по времени: 5 секунд
Ограничение по памяти: 256 Мебибайт

Маша и Миша играют в интересную игру. Каждый из них выписывает набор слов, на чём первая стадия игры заканчивается. Во второй стадии игры участники берут клавиатуру и по очереди выламывают из неё клавиши, соответствующие латинским буквам, всего l штук. После этого игроки считают количество своих слов, которые можно напечатать с помощью оставшихся клавиш. После этого тот, чьих слов остаётся меньше, проигрывает сопернику количество конфет, равное разности количеств оставшихся слов.

Первая стадия игры уже завершена, первый ход по жребию (или по мишиной галантности) предстоит Маше. Определите, сколько конфет она может себе гарантировать при оптимальной игре обоих.

Формат входного файла

В первой строке входного файла записано число l — количество клавиш, которые будут выломаны за всю игру ($1 \leq l \leq 26$). Далее записаны наборы машинных и мишинных слов в следующем формате: на одной строке количество слов, на следующей — сами слова, разделённые пробелами.

Все слова состоят из строчных букв латинского алфавита. Каждый игрок выписал не более 15 непустых слов, состоящих из не более, чем 30 букв каждое.

Формат выходного файла

Выведите единственное целое число — выигрыш Маши при оптимальной игре. Если Маша вынуждена проиграть, выведите её минимальный проигрыш со знаком «минус».

Пример

стандартный поток ввода	стандартный поток вывода
2	1
5	
abacaba a zxxzyz trava abc	
1	
a	