

Задача A. Bear

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

В лесу есть n полян, на каждой из которых растёт клубника. Некоторые поляны соединены двусторонними тропинками. Возле поляны с номером 1 живёт медведь. Каждый день медведь просыпается, идёт по тропинкам к некоторой поляне и ест клубнику. При этом медведь вытаптывает всю клубнику на полянах, через которые он проходит. Съев все ягоды на выбранной поляне, медведь к вечеру возвращается назад. Однако каждую ночь птицы съедают по одной ягоде с каждой поляны, на которой ещё осталась как минимум одна ягода. Гарантируется, что медведь может добраться до любой поляны.

Требуется найти максимальное количество ягод, которое может съесть медведь.

Формат входного файла

В первой строке входного файла заданы два целых числа m и n — количество полян и количество соединяющих их тропинок соответственно ($1 \leq n \leq 10^5$, $0 \leq m \leq 5 \cdot 10^5$). Поляны занумерованы числами от 1 до n , медведь живёт около поляны с номером 1 (и каждый день начинает поход за ягодами с этой поляны).

Далее следуют m целых чисел k_i ($0 \leq k_i \leq 10^5$). i -е из этих чисел задаёт количество ягод на i -й поляне к началу первого похода медведя за ягодами.

В последующих n строках заданы тропинки. Каждая тропинка задаётся парой целых чисел (a_i, b_i) — номерами полян, которые она соединяет ($1 \leq a_i \neq b_i \leq n$).

Формат выходного файла

Выведите одно число — максимальное количество ягод, которые может съесть медведь

Пример

<code>bear.in</code>	<code>bear.out</code>
5 5 1 2 3 7 4 1 2 3 1 3 2 5 3 2 4	11

Задача В. Cards game

Имя входного файла:	<code>cards.in</code>
Имя выходного файла:	<code>cards.out</code>
Ограничение по времени:	2 seconds
Ограничение по памяти:	64 Mebibytes

Трус, Балбес и Бывалый играют в преферанс. Слева от Труса сидит Балбес, справа — Бывалый. Каждый подозревает своих товарищей в нечестной игре, поэтому они решили автоматизировать процесс проверки партий, чтобы выявлять нарушение правил.

В преферанс играют колодой из 32-х карт. Масти обычные — пики (Spades), трефы (Clubs), бубны (Diamonds), червы (Hearts). Достоинства карт — туз (Ace), король (King), дама (Queen), валет (Jack), 10, 9, 8, 7. Каждую игру в преферансе разыгрывают три игрока. Игроки сидят по кругу и ходят по часовой стрелке.

После раздачи карт и размышления игроков начинается торговля. Мы опустим правила торговли, поскольку они не важны для данной задачи. При некоторых исходах торгов устанавливается козырная масть. После торговли на руках у игроков остаётся по 10 карт, поэтому игра состоит из десяти кругов. Каждый круг разыгрывается так: игрок, имеющий право первого хода в данном круге, кладёт на стол любую свою карту рубашкой вниз. Следующим кладет карту его левый сосед, далее кладёт свою карту оставшийся игрок.

Каждый игрок должен соблюдать следующее правило: делая ход, он обязан положить карту той же масти, что и первый игрок. Если у него нет такой масти, он должен положить карту козырной масти (если игра идёт с козырем). Если же у него нет и такой масти, он может положить любую из своих карт. Выигрывает круг (говорят: «берёт взятку») игрок, положивший козырь наибольшего достоинства, или, если никто не сходил козырем, игрок, положивший карту наибольшего достоинства той же масти, что и первая карта в круге. Взавший взятку игрок ходит первым в следующем круге. Каждый круг из игры выбывают три карты, которыми сходи ли игроки, и к началу 10го круга у них остаётся по одной карте.

Таким образом, нарушить правила можно двумя способами:

- Сходить картой не той же масти, что у карты, которую положили первой в круге, хотя такая масть у игрока есть.
- Сходить не козырем и не той же мастью, что у карты, которую положили первой в круге, хотя у игрока есть козырь.

Формат входного файла

В первой строке указан игрок, который получил право первого хода в первом круге (“Trus”, “Balbes” или “Byvaly”).

Во второй строке указана козырная масть:

‘S’ (пики), ‘C’ (трефы), ‘D’ (бубны), ‘H’ (червы) или символ ‘-’, если козырей нет.

Далее, в следующих десяти строках идёт описание партии: в каждой строке указаны три карты, сначала идёт карта, которую положил Трус, потом карта Балбеса, и, затем, Бывалого. Каждая карта описывается двумя символами: первый задаёт достоинство карты (A — туз, K — король, Q — дама, J — валет, T — десятка, 9, 8, 7), второй — масть (S, C, D, H). $(i + 2)$ -я строка входных данных описывает i -й круг игры.

Формат выходного файла

Следует записать имена игроков, которые нарушили правила в ходе игры, по одному в строке в алфавитном порядке. Если ни один из игроков не нарушил правила, нужно вывести “Fair play” (без кавычек).

Пример

cards.in	cards.out
Trus D AS JS 8S KS 7D 7S 7H AH QH 8H KH 8D 7C 9H AC 8C TC KC JC 9D QC QD KD TD AD JH QS 9S TH TS	Balbes
Byvaly - 7H JH AH TH KH 9H QH 8H AS 7C 8C AC TC KC 9C QC JC KS 7D 8D AD TD KD 9D QD JD QS TS 9S 8S	Fair play

Задача C. Circles and stripes

Имя входного файла: `circ.in`
Имя выходного файла: `circ.out`
Ограничение по времени: 7 seconds
Ограничение по памяти: 64 Mebibytes

На плоскости проведено некоторое количество белых полос, параллельных оси x . Требуется выяснить, можно ли расположить бесконечное количество окружностей радиуса s следующим образом:

- центры окружностей расположены на прямой, параллельной оси y ;
- расстояние между центрами двух соседних окружностей равно k ;
- ни одна из окружностей не пересекает белую полосу (при этом окружности могут касаться белых полос).

Формат входного файла

В первой строке входного файла задано одно целое число T ($1 \leq T \leq 10$) — количество тестовых примеров. Каждый тестовый пример состоит из двух строк. В первой строке тестового примера заданы три целых числа s , k и n ($1 \leq s < k \leq 10^9$, $2 \leq n \leq 500\,000$) — радиус окружности, расстояние между центрами двух соседних окружностей и количество полос (и промежутков между ними). Во второй строке задано n целых чисел p_i ($1 \leq p_i \leq 10^9$) — ширина соответствующих полос и промежутков, перечисленных снизу вверх: первое число задаёт ширину белой полосы с наименьшим y , второе — ширину первого промежутка между полосами, третье — ширину следующей белой полосы и так далее. В случае, если n чётно, последнее число можно не принимать во внимание — соответствующий «промежуток» сливается с верхней частью плоскости.

Формат выходного файла

Для каждого тестового примера выведите в отдельной строке “ТАК”, если расположить окружности указанным в задаче способом можно, и “НIE” в противном случае.

Пример

<code>circ.in</code>	<code>circ.out</code>
2	NIE
2 7 7	NIE
3 2 1 4 2 4 3	
2 7 4	
3 8 9 4	

Задача D. Another Lucky Number

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

По одной из версий, число считается «счастливым», если оно или делится на 7, или содержит семёрку в качестве одной из десятичных цифр.

По заданному числу требуется выяснить, является оно «счастливым» или нет.

Формат входного файла

В первой строке входного файла задано количество тестовых примеров T ($1 \leq T \leq 100$). Каждый тестовый пример задан в отдельной строке и состоит из одного целого числа n ($-10^{100} \leq n \leq 10^{100}$).

Формат выходного файла

Для каждого тестового примера в отдельной строке выведите 1, если число является «счастливым» и 0 в обратном случае.

Пример

<code>luck.in</code>	<code>luck.out</code>
4	1
7	1
105	1
17	0
108	

Задача E. Mouse

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

Одно время Вася занимался разведением мышей для Института Биологических Организмов (ИБО). Мышей у Васи было очень много — целых T штук. Они постоянно бегали по комнате взад вперед, создавая своему хозяину определенные неудобства. Дело в том, что компьютер в комнате Васи и дверь в комнату находились в разных углах. Когда Васе нужно было пройти от двери до компьютера, то ему приходилось двигаться очень осторожно, чтобы ненароком не наступить на какую-нибудь мышку.

Комната Васи представляет собой прямоугольную сетку шириной W клеток и высотой H клеток. Дверь в комнату расположена в клетке $(1, 1)$, а компьютер в клетке (W, H) . Вася и мыши в каждый момент времени могут занимать только одну клетку и не могут находиться вне сетки. За 1 секунду Вася и мыши могут перейти в любую доступную соседнюю клетку (две клетки являются соседними, если у них есть общая сторона). Мыши постоянно бегают между двумя соседними клетками взад-вперед: за 1 секунду они перебегают на соседнюю клетку, потом за 1 секунду возвращаются обратно. Вася двигается по комнате так, чтобы не пересечься путями ни с одной из мышек и не наступить ни на одну из них, в частности, нельзя двигаться навстречу мыши. Зная конфигурацию комнаты, начальные положения мышей и их манеру поведения, попробуйте определить, как быстро Вася мог добраться от двери до компьютера?

Формат входного файла

В первой строке входных данных находятся три целых числа W , H и T ($2 \leq W, H \leq 30, 0 \leq T \leq 1000$) — соответственно размеры Васиной комнаты и количество мышей.

В каждой из последующих T строк находится по три целых числа A_i , B_i и C_i , эти числа описывают поведение i -й мыши ($1 \leq A_i \leq W, 1 \leq B_i \leq H, 0 \leq C_i \leq 3, 1 \leq i \leq T$).

Числа A_i и B_i задают положение i -й мыши в нулевой момент времени — клетка с координатами (A_i, B_i) . Число C_i задает направление движения мыши в последующие моменты времени.

- Если C_i равно 0, то мышь будет бегать между клетками с координатами (A_i, B_i) и $(A_i, B_i + 1)$.
- Если C_i равно 1, то мышь будет бегать между клетками с координатами (A_i, B_i) и $(A_i, B_i - 1)$.
- Если C_i равно 2, то мышь будет бегать между клетками с координатами (A_i, B_i) и $(A_i + 1, B_i)$.
- Если C_i равно 3, то мышь будет бегать между клетками с координатами (A_i, B_i) и $(A_i - 1, B_i)$.

Гарантируется, что ни одна мышь при движении не пытается покинуть пределы сетки. Гарантируется, что клетка $(1, 1)$ не занята ни одной мышью в начальный момент времени.

Формат выходного файла

Выведите единственное число — минимальное количество времени (в секундах), за которое Вася может добраться от двери до компьютера. Если добраться до компьютера невозможно, то выведите число -1 .

Пример

mouse.in	mouse.out
4 4 6	9
2 2 3	
3 1 0	
3 2 1	
3 3 3	
2 4 1	
1 3 1	

Задача F. Tower

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

Заданы n кирпичей, пронумерованных числами от 1 до n . Все кирпичи имеют в сечении единичный квадрат и могут отличаться только длиной.

Требуется построить из существующих кирпичей башню максимальной высоты, состоящую из нескольких положенных друг на друга рядов кирпичей (уровней), удовлетворяющую следующим правилам:

- Каждый уровень представляет собой прямоугольный блок $1 \times 1 \times l_i$, составленный из некоторого количества кирпичей.
- Суммарная длина l_i кирпичей на каждом уровне не превосходит суммарной длины кирпичей на уровне, расположенном непосредственно под ним (если таковые имеются).
- Каждый блок, расположенный на некотором уровне, имеет номер больше, чем любой из блоков, расположенных ниже этого уровня (если таковые имеются).

При построении башни должны быть использованы все имеющиеся в наличии кирпичи.

Формат входного файла

В первой строке входного файла задано одно целое число n ($1 \leq n \leq 10^5$) — количество кирпичей. В следующей строке заданы длины кирпичей — n целых положительных чисел, не превышающих 10^4 .

Формат выходного файла

Выведите одно число — максимальную высоту башни, которую можно построить из заданного набора кирпичей.

Пример

<code>tower.in</code>	<code>tower.out</code>
3 2 3 4	2