

Задача А. Airport

Имя входного файла: `airport.in`
Имя выходного файла: `airport.out`
Ограничение по времени: 5 seconds
Ограничение по памяти: 256 Mebibytes

Здание аэропорта города Пловдив можно представить в виде трёхмерного куба $n \times n \times n$ со сторонами, параллельными осям координат, две противоположные вершины которого размещаются в точках $(0, 0, 0)$ — вход в аэропорт со стороны лётного поля и (n, n, n) — выход из аэропорта. Холлы аэропорта находятся в точках с целыми координатами, причём движение организовано так, что из каждого холла прибывшие пассажиры могут перейти (или переехать на лифте) в любой из холлов, у которых ровно одна координата больше соответствующей координаты текущего холла на 1, а остальные две координаты совпадают с соответствующей координатой текущего холла.

Для эффективной организации встречи организаторы олимпиады хотят узнать количество различных путей, по которым прибывшие пассажиры могут пройти до выхода из аэропорта.

Формат входного файла

Во входном файле задано единственное целое число n ($1 \leq n \leq 10^6$).

Формат выходного файла

Выведите количество различных путей по модулю 2946859.

Примеры

<code>airport.in</code>	<code>airport.out</code>
1	6

Задача В. Mr. X

Имя входного файла:	<code>mrx.in</code>
Имя выходного файла:	<code>mrx.out</code>
Ограничение по времени:	5 seconds
Ограничение по памяти:	64 Mebibytes

Между первым и вторым турами двум корреспондентам специализированного издания StAN (Statistics And News) было поручено идентифицировать промежуточного лидера IOI, таинственного Мистера Икс, набравшего больше 361 балла. От благожелательно настроенного члена жюри на известный форум поступила информация, что Мистер Икс прогуливается по местному парку.

Парк устроен следующим образом: равносторонний треугольник со стороной в 6 миль разбит аллеями, идущим по сторонам и прямым, параллельными сторонам, на 36 одинаковых равносторонних треугольников со стороной в 1 милю.

Всего получается 28 точек пересечения (включая и три исходные вершины треугольника). Точки пересечения занумерованы с единицы, начиная с одной из вершин треугольника, следующим образом: расположим треугольник так, чтобы противолежащая вершине с номером 1 сторона была горизонтальна и находилась ниже неё. Если известен номер точки пересечения N , то номер $N + 1$ имеет точка справа от неё; если же точка с номером N крайняя справа, то номер $N + 1$ имеет самая левая точка на ближайшей снизу "горизонтальной" аллее.

За квант времени Мистер Икс (желающий сохранить инкогнито и остатки интриги) может пробежать 0, 1 или 2 мили, не сворачивая, по некоторой аллее от одной точки пересечения до другой. После чего он отдыхает в этой точке пересечения до конца кванта. В этот момент каждый из корреспондентов (отдыхавших всё то время, пока Мистер Икс бежал) узнаёт новые координаты Мистера Икс и до конца текущего кванта может пробежать только 1 милю аналогичным образом. Считается, что ни Мистер Икс, ни корреспонденты территорию парка покинуть не могут и что каждому из троих известны координаты двух остальных.

Если в какой-то момент времени участник и любой из корреспондентов оказываются в состоянии покоя в одной точке пересечения, Мистер Икс идентифицируется, и его результат попадает в таблицу промежуточных результатов на сайте StAN.

Требуется по заданным точкам, в которых находился участник и каждый из корреспондентов в момент начала «охоты», выяснить, через какое количество квантов времени корреспонденты смогут идентифицировать участника.

Формат входного файла

Входной файл состоит из нескольких тестовых примеров. В каждом тестовом примере на отдельной строке записаны три числа от 1 до 28 — точки пересечения, в которых находятся два корреспондента и участник соответственно.

Формат выходного файла

Для каждого тестового примера в выходной файл выведите строчку «StAN wins in N .», где N — время, за которое корреспонденты StAN гарантированно получают необходимую информацию. Если Мистер X сможет действовать так, чтобы не быть пойманным в течение любого конечного промежутка времени, выведите строчку «stalemate».

Пример

<code>mrx.in</code>	<code>mrx.out</code>
3 4 1	StAN wins in 1.
11 24 22	StAN wins in 2.

Задача C. Domino

Имя входного файла: `domino.in`
Имя выходного файла: `domino.out`
Ограничение по времени: 5 seconds
Ограничение по памяти: 64 Mebibytes

Во время экскурсии тренерам и участникам предлагались различные сувениры, в том числе и местное домино. В данной модификации очки на костяшках домино могут принимать значение не от 0 до 6, а от 0 до 16.

После завершения олимпиады в номерах участников одной из команд были найдены несколько костяшек из разных наборов такого домино (участники собирались в спешке и забыли часть сувениров). Можно ли из найденных костяшек выложить в прямую цепочку по правилам домино (то есть чтобы на двух примыкающих друг к другу клетках соседних костяшек было указано одинаковое число)?

Формат входного файла

Входной файл состоит из нескольких тестовых примеров. В первой строке каждого примера содержится N ($1 \leq N \leq 500$) — количество найденных костяшек. В следующей строке идут N пар целых чисел от 0 до 16. i -я пара описывает количество очков, указанных в клетках i -й костяшки.

Формат выходного файла

В выходной файл для каждого тестового примера выведите текст «A chain is possible.» в случае, если все найденные костяшки домино можно выложить в одну цепь, и «No chain is possible.» в противном случае.

Пример

<code>domino.in</code>	<code>domino.out</code>
3	A chain is possible.
0 1 2 2 1 2	No chain is possible.
3	
0 1 2 2 1 1	

Задача D. Poker

Имя входного файла: `poker.in`
Имя выходного файла: `poker.out`
Ограничение по времени: 5 seconds
Ограничение по памяти: 64 Mebibytes

Команда Польши вовсе не скрывала свои результаты намеренно — просто всё свободное от решения задач время тренеры команды играли с участниками в двухлистный покер.

Двухлистный покер играется по следующим правилам. Участнику и тренеру раздаётся по 2 карты. Если оба участника имеют по две одинаковые карты, то выигрывает тот, у кого карта крупнее. Если один участник имеет две одинаковые карты, а другой — две разные, то выигрывает участник с двумя одинаковыми картами. Наконец, если у каждого из участников обе карты различны, то выигрывает тот, у кого старшая из двух карт крупнее, а при совпадении — тот, у кого оставшаяся карта крупнее.

Самой крупной картой является туз, затем идёт король, затем идёт дама, затем валет, затем десятка... и так далее до двойки, которая является самой младшей картой.

Ваша задача — по заданным картам участника и тренера определить, кто же выиграл.

Формат входного файла

Входной файл содержит несколько тестовых примеров. Каждый тестовый пример расположен в отдельной строке и состоит из 2 символов, обозначающих карты участника, пробела и 2 символов, обозначающих карты тренера. Карты обозначаются одним символом следующим образом: *akqjt98765432* для туза, короля, дамы, валета, десятки, девятки, восьмёрки, семёрки, шестёрки, пятёрки, четвёрки, тройки и двойки соответственно.

Формат выходного файла

Для каждого случая выведите «win», если участник выигрывает у тренера, и «loss» — если тренер выигрывает у участника или же их наборы карт полностью совпадают.

Пример

<code>poker.in</code>	<code>poker.out</code>
66 ak	win
t4 q7	loss
22 22	loss
44 kk	loss

Задача E. Closing ceremony

Имя входного файла:	closing.in
Имя выходного файла:	closing.out
Ограничение по времени:	5 seconds
Ограничение по памяти:	64 Mebibytes

Заккрытие олимпиады проходило в римском амфитеатре II века нашей эры — по информации археологов, именно так выглядела TopCoder Arena 2000 лет назад. Имена призёров должны были высвечиваться на гигантском табло. Однако по какой-то причине устройство отказалось работать...

Пульт управления табло представляет собой решётку $M \times N$, причём строки занумерованы от 0 до $N - 1$, а столбцы — от 0 до $M - 1$. В клетках с координатами $(0, j)$ и $(i, 0)$ для каждого положительного $j < N$ и $i < M$ в интервале от расположены светодиоды — индикаторы режима работы табло. Светодиод может быть или в выключенном, или во включённом состоянии. В некоторых из других клеток установлены кнопки управления. Кнопка в клетке с координатами (i, j) ($1 \leq i \leq M$, $1 \leq j \leq N$) действует следующим образом — при нажатии она меняет на противоположное состояние светодиодов в клетках $(i, 0)$ и $(0, j)$.

Вам задано начальное состояние светодиодов (при включении табло), а также состояние светодиодов, показывающее, что табло готово к работе. Ваша задача — найти некоторую последовательность нажатий на кнопки, переводящее табло из первого состояния во второе.

Формат входного файла

Во входном файле задано несколько тестовых примеров. В первой строке входного файла задано количество $T \leq 100$ тестовых примеров. Каждый тестовый пример состоит из 7 строк. Тестовые примеры разделяются пустой строкой.

- В первой строке заданы числа N , M и S — количество строк и столбцов, а также количество кнопок на пульте, ($1 \leq N, M \leq 1000$, $1 \leq S \leq 4000$).
- Во второй строке заданы $N - 1$ нулей или единиц, описывающих начальное состояние светодиодов в 0-м столбце.
- В третьей строке заданы $N - 1$ нулей или единиц, описывающих требуемое состояние светодиодов в 0-м столбце.
- В четвёртой строке заданы $M - 1$ нулей или единиц, описывающих начальное состояние светодиодов в 0-й строке.
- В пятой строке заданы $M - 1$ нулей или единиц, описывающих требуемое состояние светодиодов в 0-й строке.
- В шестой строке заданы S чисел в интервале от 1 до $N - 1$. i -е число указывает номер строки, в котором находится i -я кнопка.
- В седьмой строке заданы S чисел в интервале от 1 до $M - 1$. i -е число указывает номер столбца, в котором находится i -я кнопка.

Формат выходного файла

Для каждого тестового примера в выходной файл выведите одну строку. В случае, если решение есть, первым числом выведите количество нажатий на кнопки, затем — номера нажатых кнопок в порядке их задания во входном файле (кнопки занумерованы с 0 до $S - 1$). Количество нажатий на кнопки не должно превышать 1000 (гарантируется, что в случае наличия решения существует решение с не более, чем 1000 нажатий). В случае, если решения нет, выведите одно число -1 .

Пример

closing.in	closing.out
3	-1
3 3 2	2 0 2
0 0	4 0 1 3 4
1 0	
0 0	
0 1	
1 2	
1 2	
3 3 3	
0 0	
1 1	
0 0	
1 1	
1 1 2	
1 2 2	
4 4 5	
0 0 0	
0 1 1	
0 0 0	
1 0 1	
1 1 2 2 3	
1 3 1 2 2	

Задача F. Chinese contestants

Имя входного файла: `china.in`
Имя выходного файла: `china.out`
Ограничение по времени: 5 seconds
Ограничение по памяти: 64 Mebibytes

Qi Zichao и Gao YiНап набрали по 721 баллу и разделили 3-4 место. Для того, чтобы всё же выявить победителя в состязании между ними, школьники решили сыграть полученными золотыми медалями в «китайскую орлянку». Игра идёт по следующим правилам. Пусть в какой-то момент Qi имеет q юаней, а Gao - g юаней. Тогда ставка в следующем равна $\min(q, g)$. Игра идёт до того момента, пока у одного из участников не соберутся все деньги, задействованные в игре. Этот участник и становится победителем. Вероятности выпадения «орла» и «решки» при броске IOI равны $1/2$.

Ваша задача — по заданному изначальному количеству денег Q и G у Qi и Gao соответственно найти вероятность выигрыша Qi.

Формат входного файла

Во входном файле содержится несколько тестовых примеров. Каждый тестовый пример состоит из двух чисел Q и G — изначальной суммы денег у Qi и Gao соответственно.

Формат выходного файла

Для каждого тестового примера выведите вероятность выигрыша Qi, записанную в виде правильной несократимой дроби.

Пример

china.in	china.out
4 4	1/2
100 300	1/4
200 100	2/3