

Задача А. Звёздочка

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

Недавно математики Кукуляндии придумали новую бинарную операцию «звёздочка», аргументами которой являются последовательности. Операция просто приписывает первую последовательность в конец второй. Например, $(2, 4) * (1, 3) = (1, 3, 2, 4)$. Порядок выполнения нескольких операций «звёздочка» в одной строке — слева направо, но, используя скобки, порядок можно изменить (операции в скобках выполняются раньше). Так, $(3) * ((1, 5) * (2, 7)) = (2, 7, 1, 5, 3)$. Заметим, что если в качестве одного из элементов последовательности указано выражение, то сперва вычисляется его значение, а потом в получившейся последовательности стираются внутренние скобки. Например, $(1, (2 * 3), 4) = (1, 3, 2, 4)$.

Теперь кукуляндцы хотят использовать эту операцию для генерации перестановок. Точнее, они хотят получить заданную перестановку из перестановки $(1, 2, \dots, N)$, дописав скобки, запятые и звёздочки и вычислив значение получившегося выражения.

Формат входного файла

В первой строке записано число N ($1 \leq N \leq 10\,000$). Во второй строке записана перестановка чисел от 1 до N . Числа в строке разделены пробелом.

Формат выходного файла

Выведите единственную строку — корректное выражение, в результате вычисления значения которого получится данная перестановка. При этом в выражении числа от 1 до N должны идти в возрастающем порядке. Длина выражения не должна превосходить 100 000 символов. Если искомого выражения не существует, выведите «IMPOSSIBLE». Обратите внимание на то, что в выражении не должно быть пробелов и все последовательности обязательно должны быть заключены в скобки.

Примеры

asterisk.in	asterisk.out
4 3 4 2 1	(1)*(2)*(3,4)
6 5 1 2 6 4 3	IMPOSSIBLE

Задача В. Паутина Ананси

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

Усатый-Полосатый XIII решил отомстить Ананси за освобождение бабочек, разрушив дом Ананси — его паутину. Паутина состоит из N узлов, некоторые из которых соединены нитями. Будем говорить, что два узла принадлежат одному кусочку, если от одного узла до другого можно добраться по нитям паутины. Усатый-Полосатый уже решил, какие нити и в каком порядке он будет рвать, и теперь хочет узнать, на сколько кусочков будет распадаться паутина после каждого из его действий.

Формат входного файла

В первой строке через пробел записаны числа N и M — количество узлов и нитей в паутине ($1 \leq N, M \leq 100\,000$). В каждой из следующих M строк через пробел записаны два различных числа — номера узлов, которые соединяет очередная нить. Узлы занумерованы числами от 1 до N , нити занумерованы числами от 1 до M в том порядке, в котором они перечислены. Далее записано число Q — количество нитей, которое собирается порвать Усатый-Полосатый ($1 \leq Q \leq M$). В последней строке записаны номера этих нитей — различные числа, отделяемые друг от друга пробелом.

Формат выходного файла

Выведите через пробел Q чисел — число кусочков, из которых будет состоять паутина Ананси после каждого обрыва нити.

Примеры

<code>cobweb.in</code>	<code>cobweb.out</code>
4 4 1 2 2 3 1 3 3 4 3 2 4 3	1 2 3
3 1 1 2 1 1	3

Задача С. Мухи-дроиды

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

Вскоре после того, как на планете Набу воцарился мир, обычный гунган Дао-Дао решил сходить на охоту. Совсем недалеко от озера он нашёл странный металлический контейнер с эмблемой Торговой Федерации. Думая, что это просто кусок мусора, Дао-Дао закинул его подальше в лес. Но секунду спустя из леса с металлическим визгом на бедного гунгана накинудся целый рой мух-дроидов — новейшей военной разработки Федерации. Надо было срочно предупредить других гунганов! Однако Дао-Дао решил справиться с мухами в одиночку, заключив весь рой в непробиваемое силовое поле. Для максимальной устойчивости силовое поле Дао-Дао обязательно должно иметь кубическую форму, более того, на каждой грани поля обязательно должна располагаться хотя бы одна муха-дронд. Помогите гунгану вычислить, как расположить силовое поле в пространстве.

Формат входного файла

В первой строке записано целое число N ($2 \leq N \leq 100$) — количество мух-дроидов. В следующих N строках записаны координаты мух X_i, Y_i, Z_i — целые числа, по модулю не превосходящие 100. Никакие две мухи не расположены в одной точке.

Формат выходного файла

Выведите координаты трёх нормированных векторов, коллинеарных сторонам искомого куба. Числа следует выводить с максимально возможной точностью. Известно, что любой рой мух-дроидов можно заключить в силовое поле, удовлетворяющее условию задачи.

Примеры

droids.in	droids.out
6 0 1 1 1 0 1 1 1 0 1 1 2 1 2 1 2 1 1	1 0 0 0 1 0 0 0 1
2 1 0 0 -1 0 0	0.577350269 0.211324865 -0.788675135 0.577350269 0.577350269 0.577350269 0.577350269 -0.788675135 0.211324865

Задача D. Допуск к экзамену

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

В прошлом семестре студенты матмеха Екатеринозаводского университета должны были сдавать экзамен по сетевым технологиям. N преподавателей, ведущих этот предмет, договорились между собой следующим образом: за семестр по этому предмету состоится N^2 лабораторных работ, причём первый преподаватель проведёт лабораторные с номерами $1, N + 1, 2N + 1, \dots, N^2 - N + 1$, второй — лабораторные с номерами $2, N + 2, 2N + 2, \dots, N^2 - N + 2$, и так далее. N -й преподаватель проведёт лабораторные с номерами $N, 2N, 3N, \dots, N^2$. Также преподаватели вспомнили, что за последние годы ленивые студенты стали пропускать много лабораторных, из-за чего потом плохо сдают экзамен. Поэтому они решили, что студент будет допущен к экзамену только если посетит хотя бы одну лабораторную каждого преподавателя.

N студентов, живущих в одной комнате общежития, не знали, сколько лабораторных состоится в течение семестра и сколько преподавателей ведёт их. У этих студентов было разное отношение к учёбе: первый студент в течение семестра ходил на все лабораторные, второй — только на лабораторные с номером, кратным двум, третий — только на лабораторные с номером, кратным трём, и так далее. . . После завершения всех лабораторных оказалось, что к экзамену допущено лишь K из этих студентов.

Формат входного файла

Целое число K ($1 \leq K \leq 2 \cdot 10^9$).

Формат выходного файла

Выведите минимально возможное N , удовлетворяющее условию задачи. Если ни при каком N к экзамену не может быть допущено ровно K студентов, выведите 0.

Примеры

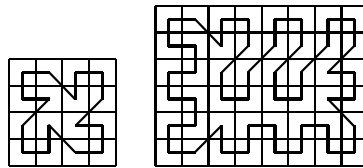
<code>exam.in</code>	<code>exam.out</code>
8	15
3	0

Задача Е. Пьяный король

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

Пьяный король — это фигура, которая ходит, как обычный шахматный король (то есть, на соседнюю клетку), но не может сделать подряд два хода в одном направлении. Пьяный король стоит в левом верхнем углу шахматной доски размера $N \times M$ и хочет обойти доску, побывав в каждой клетке один раз, и вернуться назад. При этом запрещается пересекать свой путь.

Примеры корректного обхода:



Помогите королю найти требуемый обход.

Формат входного файла

Единственная строка содержит 2 целых числа: N и M ($2 \leq N, M \leq 500$).

Формат выходного файла

В первой строке выведите «Yes» или «No» в зависимости от того, можно ли обойти доску необходимым способом. Если обход существует, то в следующих строках выведите его, используя символы «o» (код 111), «|» (код 124), «-» (код 45), «/» (код 47), «\» (код 92) и пробелы. Следуйте формату, приведённому в примере ниже. Если есть несколько возможных обходов, то разрешается вывести любой.

Примеры

king.in	king.out
6 8	Yes o-o o-o o-o o-o \ / / o-o o o o o o-o / \ \ o-o o-o o-o o-o \ \ \ o-o o-o o-o o-o \ \ \ o-o o o o-o o o / / o-o o-o o-o o-o
5 5	No

Задача F. Лунокод 2

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

Немногие помнят, что способ кодирования информации, известный ныне как лунокод, был изобретён ещё в ходе лунно-марсианской войны. С незначительными модификациями он и сейчас используется лунатиками для передачи данных. Передаваемая информация в виде набора нулей и единиц записывается в матрицу размером $M \times N$. На матрицу наложено следующее ограничение (контрольное условие): у неё должно быть ровно K нулевых строк и ровно L нулевых столбцов. Если после приёма оказывается, что полученная матрица не удовлетворяют контрольному условию, значит, некоторое количество её ячеек было искажено при передаче.

В ходе отчёта перед президентом Лунной Федерации министр связи предложил провести реформу лунокода. Министр аргументировал это тем, что количество различных сообщений, которые могут быть переданы, не так уж и велико. Президент поручил министерству совместно с Лунной Академией Наук исследовать данный вопрос, чтобы решить, действительно ли необходима реформа. В ходе исследования оказалось, что министр заблуждался: ведь при достаточно больших M и N количество матриц из нулей и единиц размера $M \times N$, удовлетворяющих контрольному условию, огромно. Сможете ли вы определить, сколько их?

Формат входного файла

В единственной строке через пробел записаны 4 целых числа: M, N, K, L ($1 \leq M, N \leq 100\,000$; $0 \leq K \leq M$; $0 \leq L \leq N$).

Формат выходного файла

Как было уже сказано, количество искомых матриц может быть очень велико, поэтому нет нужды выдавать его полностью. Выведите остаток от деления этого числа на $10^9 + 7$.

Примеры

<code>lunocode.in</code>	<code>lunocode.out</code>
2 2 0 0	7
2 3 1 1	6

Задача G. Смертельная битва

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

Чтобы монстры из Внешнего мира смогли завоевать Землю, они должны одержать десять побед подряд в Смертельной Битве. Турниры проводятся раз в поколение, и бойцы Земли потерпели уже девять поражений. Настал час решающей Смертельной Битвы.

В Смертельной Битве участвуют N монстров и M лучших бойцов Земли. По правилам турнира, каждый монстр должен драться с одним из людей (все монстры с разными людьми). В случае если хотя бы один монстр одержит победу, то Земля навеки перейдёт во владение ужасного императора Внешнего мира. Впрочем, за людьми остаётся право выбора соперников и очередности боёв.

Бог Рейден, покровитель Земли, должен выбрать бойцов так, чтобы все люди одержали победу над соперниками. Для каждого из бойцов Земли известно, каких монстров он способен победить. Для начала, требуется выбрать пару соперников на первый бой. Так, Лю Кэн хочет драться с Горо, но он единственный боец, способный победить Шан Цунга, в то время как Горо может быть повержен и другими бойцами, например, Джонни Кейджем. Поэтому первый бой между Лю Кэном и Горо даже в случае победы Лю Кэна неизбежно приведёт к захвату Земли, так как затем Шан Цунг одержит победу над своим соперником.

Определите, какие пары ни в коем случае не должны быть выбраны Рейденом, чтобы у людей остался шанс сохранить свою свободу.

Формат входного файла

В первой строке даны целые числа N и M . $1 \leq N \leq 300$; $N \leq M \leq 1500$. Далее приведена матрица A из нулей и единиц размера $N \times M$. $A_{ij} = 1$, если и только если j -й боец Земли способен одержать победу над i -м монстром.

Формат выходного файла

Выведите матрицу B размера $N \times M$. B_{ij} должно равняться единице, если на первый бой не может быть выбрана пара соперников (i, j) , и нулю в противном случае.

Примеры

mk.in	mk.out
4 4	1000
1111	0111
1000	1000
1111	1000
1111	
4 5	11111
10000	11111
10000	11111
10000	11111
10000	

Задача Н. Обезьяна за клавиатурой

Имя входного файла: `monkey.in`
Имя выходного файла: `monkey.out`
Ограничение по времени: 5 секунд
Ограничение по памяти: 64 мегабайта

Ведущие телепередачи «Занимательная математика» решили убедить зрителей в справедливости известного утверждения о том, что обезьяна, посаженная за клавиатуру и случайным образом нажимающая на клавиши, согласно теории вероятности, рано или поздно наберёт нужное слово. Обезьяна, которую для съёмок предоставил городской зоопарк, уже обучена работе за клавиатурой — один раз в секунду она с равной вероятностью набирает одну из N первых букв латинского алфавита. По счастливой случайности, в слове, подготовленном ведущими, тоже используются только эти N букв. Однако перед тем, как заставить обезьяну трудиться, ведущие хотят знать, сколько времени у неё займёт работа. Точнее, через сколько секунд в строке, набранной обезьяной, встретится указанное слово?

Формат входного файла

Первая строка содержит число N — количество символов, которые умеет набирать обезьяна ($1 \leq N \leq 26$). Во второй строке записано предложенное ведущими слово. Длина слова положительна и не превосходит 30 000.

Формат выходного файла

Выведите ожидаемое время, которое обезьяна потратит на работу, округлённое вниз до ближайшего целого числа.

Примеры

<code>monkey.in</code>	<code>monkey.out</code>
2 aa	6
2 ba	4

Задача I. Космический покер 3

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

Космический покер. Легендарная игра, первая версия которой появилась ещё в далёком 1284 году Чужой эры. До сих пор её правила известны лишь узкому кругу профессиональных игроков. Однако Вам повезло — разработчики первой в мире программы, играющей в космический покер, обратились к Вам за помощью.

В космический покер играют N инопланетян. В начале раунда каждому из игроков раздают по M карт (назовём их *личными*). Личные карты игрока неизвестны его соперникам. Затем на стол по очереди выкладываются K *общих* карт. Общие карты кладут рубашкой вниз, поэтому они видны всем игрокам. *Рука* игрока состоит из его личных и общих карт — итого $M + K$ карт. Мастей нет, карты различаются лишь достоинством. Всего есть 13 различных достоинств: «2», «3», «4», ..., «9», «Т», «J», «Q», «K» и «А». Играют бесконечной колодой, в которой вероятность того, что очередная карта будет иметь заданное достоинство, равна $1/13$. Комбинации в космическом покере имеют вид (v_1, \dots, v_L) , где L — количество различных достоинств в комбинации. Рука игрока удовлетворяет комбинации (v_1, \dots, v_L) , если содержит v_1 карт первого достоинства, v_2 карт второго достоинства, ..., v_L карт L -го достоинства. Например, комбинации (2, 2) удовлетворяют руки «2JА2А» и «22233». Комбинации (2, 3) удовлетворяет рука «KQKQKQ», но не удовлетворяет рука «АААААА». Все комбинации имеют различную стоимость. В раунде побеждает игрок, рука которого содержит комбинацию наибольшей стоимости среди всех комбинаций на руках всех игроков. Если таких игроков несколько, объявляется ничья.

Зная личные карты первого игрока и частично открытые общие карты, посчитайте вероятность того, что этот игрок окажется единоличным победителем раунда.

Формат входного файла

В первой строке через пробел записаны числа N , M и K ($2 \leq N, M \leq 10$; $1 \leq K \leq 5$). Во второй строке записаны M символов — личные карты первого игрока. В третьей строке записано не более K символов — открытые общие карты. В четвёртой строке записано число C — количество существующих в космическом покере комбинаций ($1 \leq C \leq 100$). Далее в C строках перечислены комбинации в порядке возрастания стоимости. Каждая из них имеет вид $L \ v_1 \ v_2 \ \dots \ v_L$. Числа L и v_i положительны, сумма всех v_i не превышает $M + K$.

Формат выходного файла

Выведите вероятность победы первого игрока с точностью не менее 10^{-5} .

Примеры

poker.in	poker.out
2 5 2 23456 1 1 2	0.0883526857
2 5 2 23456 78 2 7 1 1 1 1 1 1 1 4	0.8407915043

Задача J. Башня дядюшки Скруджа

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

У Скруджа МакДака в собственности имеется круглый остров радиуса R с центром в начале координат. Он хочет построить там квадратную башню, где будет хранить свои сбережения. Денег много, поэтому квадрат основания башни должен быть вписан в круг острова. Тётушка Ключдия, начитавшись модных журналов, сказала, что, согласно фэн-шуй, вход в башню должен быть в точке с координатами (x_1, y_1) , а выход — в точке с координатами (x_2, y_2) . Безусловно, точки входа и выхода должны находиться на границе квадрата основания башни. Дядюшка Скрудж сообщил вам все эти данные и хочет узнать, удастся ли строителям выполнить требование тётушки Ключдии.

Формат входного файла

В первой строке записано число T — количество тестов ($1 \leq T \leq 1\,000$). В каждой из следующих T строк через пробел перечислены целые числа: R, x_1, y_1, x_2, y_2 . $1 \leq R \leq 10\,000$. Точки (x_1, y_1) и (x_2, y_2) лежат в круге.

Формат выходного файла

Для каждого из T тестов выведите «YES», если башню построить можно. Иначе выведите «NO».

Пример

<code>tower.in</code>	<code>tower.out</code>
3	YES
2 1 1 1 -1	NO
2 1 1 0 0	YES
1 1 0 -1 0	