

Задача А. Сделай треугольник!

Имя входного файла: стандартный ввод или `input.txt`
Имя выходного файла: стандартный вывод или `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

У Маши есть три палочки длиной a , b и c сантиметров. За одну минуту Маша может увеличить длину любой из палочек на один сантиметр. Ломать палочки запрещается. За какое минимальное время Маша сможет собрать треугольник положительной площади, сторонами которого будут палочки, если концы палочек должны быть вершинами треугольника?

Формат входных данных

В первой строке заданы три целых числа a , b и c ($1 \leq a, b, c \leq 100$) — длины палочек, которые есть у Маши.

Формат выходных данных

Выведите одно целое число — минимальное количество минут, за которое Маша сможет сделать треугольник положительной площади из своих палочек.

Примеры

стандартный ввод	стандартный вывод
3 4 5	0
2 5 3	1
100 10 10	81

Замечание

В первом примере Маша может сделать треугольник из палочек, не меняя длины ни одной из палочек.

Во втором примере Маша не может построить треугольник положительной площади из палочек, которые у нее есть, но может удлинить палочку длиной 2 сантиметра на один сантиметр за одну минуту, после чего построить треугольник из палочек со сторонами 3, 3 и 5 сантиметров.

В третьем примере Маша может за 33 минуты удлинить одну из палочек длиной 10 сантиметров на 33 сантиметра, а затем за 48 минут удлинить другую палочку длиной 10 сантиметров на 48 сантиметров, после чего построить треугольник из палочек с длинами 43, 58 и 100 сантиметров. На это Маша потратит 81 минуту. Можно доказать, что Маша не сможет получить треугольник за меньшее время.

Задача В. Ох уж эти палиндромы

Имя входного файла: стандартный ввод или `input.txt`
Имя выходного файла: стандартный вывод или `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Назовем *палиндромом* непустую строку, которая читается одинаково справа налево и слева направо. Например, “`abcba`”, “`a`” и “`abba`” являются палиндромами, а “`abab`” и “`xy`” не являются.

Назовем *подстрокой* строки строку, полученную отбрасыванием некоторого (возможно, нулевого) количества символов с начала и с конца строки. Например, “`abc`”, “`ab`” и “`c`” являются подстроками строки “`abc`”, а “`ac`” и “`d`” не являются.

Назовем *палиндромностью* строки количество ее подстрок, которые являются палиндромами. Например, палиндромность строки “`aaa`” равна 6, так как все ее подстроки являются палиндромами, а палиндромность строки “`abc`” равна трем, так как только подстроки длины 1 являются палиндромами.

Дана строка s . Вы можете произвольным образом переставлять символы в ней. Требуется получить строку, имеющую максимальную палиндромность.

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 100\,000$) — длина строки s .

Во второй строке задана строка s , состоящая из n строчных букв латинского алфавита.

Формат выходных данных

Выведите строку t , которая состоит из тех же символов (с учётом кратностей), что и s , и имеет максимальную палиндромность среди всех строк, которые могут быть получены из s перестановкой символов.

Если подходящих строк несколько, выведите любую.

Примеры

стандартный ввод	стандартный вывод
5 oolol	ololo
16 gagadbcgghhchbdf	abccbaghghghgdfd

Замечание

В первом примере у строки “`ololo`” есть 9 подстрок-палиндромов: “`o`”, “`l`”, “`o`”, “`l`”, “`o`”, “`olo`”, “`lol`”, “`olo`”, “`ololo`”. Обратите внимание, что некоторые подстроки совпадают, но учитываются несколько раз.

Во втором примере палиндромность строки “`abccbaghghghgdfd`” равна 29.

Задача С. Индра и *tex*

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Индра — большой любитель математики. Читая книгу по теории игр, он наткнулся на интересную функцию *tex*. Она определяется следующим образом: $tex(A)$ — это минимальное **положительное** целое число, которое отсутствует в множестве A . Например, *tex* множества $\{1, 2, 3, 5, 100\}$ равен 4, а *tex* множества $\{2, 3, 4, 5\}$ равен 1.

Чтобы поупражняться с применением функции *tex*, Индра взял множество чисел A , состоящее из n целых положительных чисел, и положительное число k . Затем Индра k раз произвёл следующую операцию: он добавил в множество A ещё одно число, равное $tex(A)$, тем самым, каждый раз увеличивая размер множества A на один.

По заданному множеству A и числу k определите последнее число, которое Индра добавит в множество.

Формат входных данных

В первой строке заданы два целых числа n и k ($1 \leq n \leq 100\,000$, $1 \leq k \leq 10^9$) — количество чисел в множестве и количество операций добавления числа, произведённых Индрой.

Вторая строка содержит n **различных** целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 100\,000$) — элементы множества A .

Формат выходных данных

Выведите одно целое число — последнее число, которое Индра добавит в множество.

Примеры

стандартный ввод	стандартный вывод
4 1 4 2 1 5	3
7 10 1 3 20 2 7 45 5	15

Замечание

В первом примере *tex* множества $\{1, 2, 4, 5\}$ равен 3, после добавления *tex* в множество, оно станет равным $\{1, 2, 3, 4, 5\}$.

Задача D. Уравнения математической магии

Имя входного файла: стандартный ввод или `input.txt`
Имя выходного файла: стандартный вывод или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Колоссально! — воскликнул горбоносый. —
Программист! Нам нужен именно
программист.

Аркадий и Борис Стругацкие,
Понедельник начинается в субботу

Изучая книгу «Уравнения математической магии» Роман Ойра-Ойра и Кристобаль Хунта обнаружили интересное уравнение: $a - (a \oplus x) - x = 0$ для заданного a , где знаком $\oplus$ обозначено побитовое исключающее ИЛИ (XOR) двух чисел (эта операция обозначается как `^` или `xor` во многих современных языках программирования). Ойра-Ойра быстро нашел x , являющееся решением, однако Кристобаль Хунте результат Ойры-Ойры показался недостаточно интересным, поэтому он спросил коллегу, сколько существует **неотрицательных** решений данного уравнения. Такая задача оказалась для Ойры-Ойры слишком сложной, поэтому он обратился за помощью к Вам.

Формат входных данных

Вам предстоит решить задачу для нескольких возможных значений параметра a . В первой строке находится целое число t от 1 до 1000 включительно — количество этих значений.

В последующих t строках находятся значения параметра a , каждое значение — целое число от 0 до $2^{30} - 1$ включительно.

Формат выходных данных

Для каждого значения параметра a из входных данных выведите в отдельной строке одно целое число — количество неотрицательных решений уравнения с данным значением параметра. Ответы выводите в том же порядке, в каком параметры следуют во входных данных.

Можно доказать, что количество решений всегда конечно.

Пример

стандартный ввод	стандартный вывод
3	1
0	2
2	1073741824
1073741823	

Замечание

Определим операцию побитового ИЛИ (XOR). Пусть даны два целых неотрицательных числа x и y , рассмотрим их двоичные записи (возможно с ведущими нулями): $x_k \dots x_2 x_1 x_0$ и $y_k \dots y_2 y_1 y_0$. Здесь x_i это i -й бит числа x , а y_i это i -й бит числа y . Пусть $r = x \oplus y$ — результат операции XOR над числами x и y . Тогда двоичной записью r будет $r_k \dots r_2 r_1 r_0$, где:

$$r_i = \begin{cases} 1, & \text{если } x_i \neq y_i \\ 0, & \text{если } x_i = y_i \end{cases}$$

Для первого значения параметра только 0 является решением уравнения.

Для второго значения параметра решениями уравнения являются 0 и 2.

Задача Е. Угадай-ка!

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Это интерактивная задача.

В выходной Алексей вместе со своими одноклассниками решил посетить парк аттракционов. Поскольку Алексею очень нравится решать головоломки и получать за это призы, его заинтересовал новый павильон «Угадай-ка!», где посетителям предлагается сыграть в следующую игру.

На столе в павильоне есть клетчатое прямоугольное поле размера $n \times m$, каждая клетка раскрашена в один из k разных цветов, и есть колода из $n \cdot m$ карт с написанными на них числами от 1 до $n \cdot m$. Ведущий игры в павильоне загадывает одну из карт колоды, а посетитель должен её отгадать. Чтобы отгадать неизвестную ему карту, посетитель задаёт ведущему один или несколько запросов следующим образом:

- Посетитель по своему усмотрению раскладывает на поле все карты номерами вверх, ровно по одной карте в каждой клетке поля
- Ведущий смотрит на поле и говорит посетителю цвет клетки, на которой лежит загаданная карта

Всего посетителю разрешено разложить карты и получить ответ ведущего не более 99 раз, после чего он должен правильно назвать номер загаданной карты. Гарантируется, что ответы ведущего не противоречивы, то есть всегда существует по крайней мере одна карта, для которой все данные ведущим ответы верны.

Напишите программу, которая поможет Алексею получить приз!

Протокол взаимодействия

Ваша программа будет общаться с тестирующей системой по протоколу, описанному ниже.

При запуске решения на вход подаются три целых положительных числа n , m и k ($1 \leq n, m \leq 10$, $2 \leq k \leq n \cdot m$) — размеры игрового поля и количество цветов, в которые раскрашено поле. В каждой из следующих n строк передаётся по m целых чисел через пробел, каждое от 1 до k — раскраска клеток игрового поля. Гарантируется, что для каждого из k цветов на поле есть по крайней мере одна клетка, раскрашенная в этот цвет.

Вам разрешается задать ведущему **не более 99 запросов**, чтобы угадать загаданную ведущим карту.

Чтобы задать запрос, выведите строку «?», после чего выведите n строк по m целых чисел в каждой — номера карт, раскладываемых вами в соответствующие клетки поля. Каждое число должно быть от 1 до $n \cdot m$, номера карт не должны повторяться.

В ответ на каждый запрос вводится целое число c — цвет клетки, в которую вы положили загаданную карту, или число -1 в случае, если программа превысила ограничение по числу запросов. Обратите внимание, в случае считывания числа -1 вы **обязательно** должны сразу завершить вашу программу. В противном случае, вердикт тестирующей системы может быть некорректным!

Как только вы найдете ответ, выведите строку «Ready!». В следующей строке выведите целое число x ($1 \leq x \leq n \cdot m$) — номер загаданной ведущим карты. После этого ваша программа должна завершиться.

Для установки правильности вашей программы жюри может запускать её произвольное количество раз для одного и того же игрового поля с разными загаданными картами.

Пример

стандартный ввод	стандартный вывод
2 3 3 1 2 1 2 3 3	
	?
	1 2 3 4 5 6
1	?
	2 4 6 3 1 5
2	Ready! 3

Замечание

В условии в примере взаимодействия вводимые и выводимые данные расположены для удобства восприятия в хронологическом порядке, при реальном взаимодействии никакие «лишние» переводы строк возникать не должны.

В примере поле 2 на 3 раскрашено в 3 цвета. Ответ на первый запрос — цвет 1, следовательно, ведущий загадал карту с номером 1 или с номером 3. Ответ на второй запрос — цвет 2, это говорит о том, что ведущий загадал карту с номером 3 или 4. Следовательно, была загадана карта с номером 3. Обратите внимание, что при проверке вашей программы на этом тесте в тестирующей системе может быть загадана другая карта.

В точности соблюдайте формат выходных данных. После каждого вывода обязательно выводите один перевод строки и сбрасывайте буфер вывода — для этого используйте `flush(output)` на языке Паскаль или Delphi, `fflush(stdout)` или `cout.flush()` в C/C++, `sys.stdout.flush()` на языке Python, `System.out.flush()` на языке Java.

Задача F. LinkedList's Bizarre Adventure

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

В двусвязном списке, он же **LinkedList**, каждый элемент может быть связан максимум с двумя другими элементами — с предыдущим элементом (если он есть) и со следующим элементом (если он есть).

Билли и Рикардо проходят стажировку в компании FlexDex в группе поддержки внутреннего анонимного форума с возможностью деанонимизации. Для представления последовательных сообщений в теме необходимо использовать список, где элементами будут являться эти сообщения, и два товарища решили написать свою быструю реализацию двусвязного списка **FlexList**.

Но у них что-то пошло не так — в их списке новое сообщение может связываться не только с первым или последним сообщением, как должно быть, а с любым из уже существующих сообщений.

Если представить каждое сообщение как вершину графа, а связи между сообщениями как ребра графа, то гарантируется, что этот граф будет неориентированным, связным и ацикличным.

Говорится, что граф является *корректным графом*, если в этом графе у каждой вершины не более двух вершин, связанных с ней ребром. Изначально же **FlexList** связи между сообщениями могут выглядеть в виде **любого** связного ациклического графа, не обязательно корректного.

Билли и Рикардо не растерялись и решили, что будут вручную исправлять все неясности в ходе работы программ, которые используют их изобретение. Тимлид дал им тестовый пример для проверки кода. Они получили на этом примере граф связей и вручную делают из него корректный граф, ведь связи между сообщениями в корректном двусвязном списке могут представлять собой только корректный граф.

Это выглядит так — сначала Билли проверяет, что граф корректный. Если это не так, то он выбирает и удаляет некоторый *лист* (вершина, у которой есть ровно одна связь с другими вершинами), и отдает граф Рикардо, затем Рикардо делает то же самое и отдает граф Билли, и так продолжается до тех пор, пока кто-то не получит от товарища корректный граф. Как только один из двух друзей получает такой граф, то тут же показывает его тимлиду, с надеждой на похвалу и повышение в должности.

Каждый хочет получить первым такой граф. Кто первым попадет к тимлиду, если оба товарища удаляют листья оптимально для себя?

Формат входных данных

В первой строке содержится одно целое число n ($1 \leq n \leq 300\,000$) — число сообщений в примере тимлида.

Следующие $n - 1$ строк задают связи между сообщениями. Каждая из них содержит два целых числа a_i и b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$), которые показывают, что сообщения с номерами a_i и b_i связаны.

Формат выходных данных

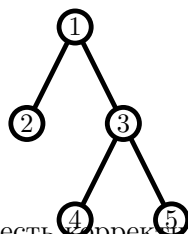
Выведите “Billy” (без кавычек), если первым попадет к тимлиду Билли, иначе выведите “Ricardo” (без кавычек).

Примеры

стандартный ввод	стандартный вывод
5 1 2 1 3 3 4 3 5	Billy
7 1 2 2 3 3 4 4 5 5 6 6 7	Billy
6 1 2 1 3 1 4 1 5 1 6	Ricardo

Замечание

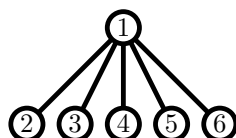
В первом примере Билли первым ходом может удалить одну из вершин с номерами 2, 4 или 5, так как они являются листьями. Он не будет удалять вершину с номером 4 или 5, так как в таком случае он передаст Рикардо корректный граф и проиграет. Значит, он удалит вершину 2. В свою очередь, Рикардо может удалить вершины 1, 4 или 5, и, в любом случае, Билли получит от него корректный граф.



Во втором примере у Билли сразу есть корректный граф.



В третьем примере можно показать, что вне зависимости от ходов Билли, Рикардо получит корректный граф первым.



Задача G. Дипломы

Имя входного файла: стандартный ввод или `input.txt`
Имя выходного файла: стандартный вывод или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Жюри Московской Олимпиады Школьников по программированию уже много лет проводит олимпиады. За это время у них накопилось огромное количество дипломов. С котятками, с видами на Воробьевы Горы, с основными достопримечательностями города-героя Москвы и многими другими рисунками на фоне.

На носу была Московская Командная Олимпиада Школьников, на которой будет раздаваться очередной набор дипломов. Последняя проблема, которую всё никак не удавалось решить — кто же будет изображён на дипломах в этом году. Кандидатов было много, но в конце концов остались только Пегас Артур и Единорог Олег. Так как всему составу жюри собраться в одно время в одном месте довольно сложно, было решено провести электронное голосование.

Процесс голосования проходит следующим образом: каждый из членов жюри в течение дня голосования отправляет письмо на почтовый ящик с единственным словом: “PEGAS” или “UNICORN”. Дальше специально обученный бот открывает каждое письмо, считывает кодовое слово и заносит в базу голосования. Более того он поддерживает результаты голосования в онлайн режиме и выкладывает на всем небезызвестный сайт.

И вот пришёл День выборов нового символа для диплома. Ровно в полдень результаты голосования складывались таким образом, что Единорог Олег выигрывал с отрывом в $a\%$ голосов от Пегаса Артура. Притом известно, что проголосовало только $p\%$ членов жюри. Пегасу Артуру стало интересно, есть ли у него еще шансы на победу. Вам известно, что победитель считается абсолютным только в том случае, если отрыв от соперника составляет **хотя бы один** процент голосов. Иначе победителем становится Единорог Олег, так как у него красивые глаза. В случае, если у Пегаса еще есть шансы на абсолютную победу, его интересует, какой **минимальный целый** процент голосов среди **ещё не проголосовавших** членов жюри он для этого должен набрать.

Формат входных данных

В первой строке заданы два целых числа a и p . ($0 \leq a \leq 100, 1 \leq p \leq 100$) — текущий отрыв (в процентах голосов) Единорога Олега от Пегаса Артура и процент членов жюри, которые уже проголосовали, соответственно.

Формат выходных данных

Выведите одно целое число — минимальное **целое** количество процентов от оставшегося количества голосов, достаточное для того, чтобы Пегас Артур стал абсолютным победителем. Если у Пегаса нет шансов на абсолютную победу, выведите «Impossible» (без кавычек).

Примеры

стандартный ввод	стандартный вывод
63 31	65
0 100	Impossible

Замечание

В первом примере у Артура 18.5% поддержки от проголосовавшего в данный момент числа членов жюри, а у Олега 81.5%. Чтобы победить, Артуру надо набрать как минимум 65% поддержки среди непроголосовавших, чтобы стать абсолютным победителем. В этом случае за Единорога Олега проголосует 49.415%, а за Артура 50.585%. Если же Артур наберет 64%, то у него будет 49.895%, а у Олега 50.105%.

Во втором примере уже все голоса посчитаны, а мнения голосовавших разделились поровну, поэтому победит Единорог Олег.

Задача Н. Слабое звено

Имя входного файла: стандартный ввод или `input.txt`
Имя выходного файла: стандартный вывод или `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

В Берляндии по воскресеньям проходит известное шоу — игра «Слабое звено».

В игре принимают участие n игроков, которые выстраиваются в круг. Каждому игроку сопоставлен рейтинг — некоторое целое число a_i .

Игра проходит в несколько раундов, каждый из которых выглядит следующим образом:

- В очередном раунде принимают участие все ещё не выбывшие игроки.
- Все игроки, которые по рейтингу строго слабее обоих своих соседей по кругу, объявляются слабым звеном и выбывают из игры.
- Все оставшиеся участники сдвигаются чуть плотнее, чтобы снова образовывать круг.
- Игра заканчивается, если после очередного раунда осталось ровно два человека или если после очередного раунда не выбыл ни один человек.
- Иначе начинается новый раунд.

Можно показать, что если до очередного раунда в игре оставалось хотя бы три участника, то после одного раунда гарантированно останется не менее двух участников.

Вам нужно выяснить для каждого участника, останется ли он до конца, или номер раунда, в котором он покинет игру.

Формат входных данных

В первой строке дано одно целое число n ($2 \leq n \leq 200\,000$) — количество участников в игре.

Вторая строка содержит n целых чисел a_i ($1 \leq a_i \leq 200\,000$) — рейтинги всех участников игры в том порядке, в котором они стоят, при этом участник с номером n является соседом участника с номером 1.

Формат выходных данных

Выведите n целых чисел — номер раунда, в котором участник игры с номером i покинет игру, или 0, если этот игрок останется до конца игры.

Раунды нумеруются последовательными целыми числами начиная с 1.

Примеры

стандартный ввод	стандартный вывод
5 4 5 5 2 3	3 0 0 1 2
5 5 1 3 1 5	0 1 2 1 0
3 6 6 6	0 0 0
4 6 5 5 6	0 0 0 0

Замечание

В первом примере игра проходит следующим образом (при помощи `_` обозначаются выбывшие участники):

$[4; 5; 5; 2; 3] \rightarrow [4; 5; 5; _; 3] \rightarrow [4; 5; 5; _; _] \rightarrow [_; 5; 5; _; _]$

После этого игра заканчивается, так как осталось ровно два человека.

Во втором примере игра проходит следующим образом:

$[5; 1; 3; 1; 5] \rightarrow [5; _; 3; _; 5] \rightarrow [5; _; _; _; 5]$

В третьем и четвёртом примере нет ни одного игрока, который был бы одновременно слабее обоих своих соседей, поэтому игра заканчивается, не успев начаться.

Задача I. Конфеты детям

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

На детском празднике дети водили хороводы. Как только музыка закончила играть, дети всё ещё стояли в кругу. Тут Лена вспомнила, что родители дали ей коробку с k конфетками «Wilky May». Лена не жадина, поэтому она решила раздать все свои конфетки друзьям из хоровода. Лена знает, что некоторые её друзья сладкоежки, а некоторые нет. Сладкоежки берут из коробки две конфетки, если в коробке есть хотя бы две конфетки, а иначе берут одну. Остальные друзья Лены всегда берут ровно одну конфетку из коробки.

Чтобы начать раздавать конфетки, Лена вышла из хоровода, после чего в хороводе остались n ее друзей. Чтобы раздавать конфетки было проще, Лена присвоила каждому другу в хороводе номер в порядке по часовой стрелке, начиная с её лучшего друга Ромы, который получил номер 1.

Лена дала коробку другу, который получил номер l , после чего каждый друг Лены, начиная с друга с номером l , брал конфетки из коробки и передавал коробку следующему в порядке по часовой стрелке другу. После того, как друг с номером r взял конфетки, в коробке конфеток не осталось. Обратите внимание, что возможно, что некоторые друзья Лены брали конфетки из коробки несколько раз, то есть коробка могла пройти несколько полных кругов по хороводу.

Лена не знает, кто из ее друзей сладкоежки, но ее интересует, сколько максимум из её друзей могут быть сладкоежками. Если же такой ситуации не могло случиться, и Лена ошиблась в своих наблюдениях, сообщите ей об этом.

Формат входных данных

В единственной строке задаются четыре целых числа n, l, r, k ($1 \leq n, k \leq 100, 1 \leq l, r \leq n$) — количество детей в хороводе, номер друга, которому Лена отдала коробку конфет, номер друга, который взял последнюю конфетку и количество конфет в коробке, соответственно.

Формат выходных данных

Выведите одно целое число — максимально возможное количество сладкоежек среди друзей Лены или «-1» (без кавычек), если Лена ошиблась в своих наблюдениях.

Примеры

стандартный ввод	стандартный вывод
4 1 4 12	2
5 3 4 10	3
10 5 5 1	10
5 4 5 6	-1

Замечание

В первом примере любые два друга могут быть сладкоежками, тогда каждый два раза получит коробку конфет и последним, кто возьмёт конфету будет четвёртый человек.

Во втором примере сладкоежками могут быть любые три друга, кроме друга, стоящего на третьем месте.

В третьем примере только один друг возьмёт одну конфетку, но он может быть сладкоежкой, просто он не может взять две конфеты. Все остальные в кругу тоже могут быть сладкоежками, но они не могут взять ни одной конфеты.

В четвёртом примере Лена ошиблась и такой ситуации быть не могло.