

Задача А. Возможно из ЕГЭ

Имя входного файла: *стандартный ввод* или `input.txt`
Имя выходного файла: *стандартный вывод* или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Маленький Вася нашел на полке у брата книгу, в которой описаны много различных задач, и одна из них его сильно заинтересовала. В ней требуется определить, равны ли 0 все разряды с l по r включительно в двоичной записи числа N .

Так как Вася ещё маленький и не знает других систем счисления, кроме десятичной, то он просит Вас о помощи в решении данной задачи.

Формат входных данных

В единственной строке заданы три целых числа N , l и r ($0 \leq l, r, N \leq 10^9$; $l \leq r$).

Формат выходных данных

Выведите «YES», если все разряды с l по r числа N в двоичной системе счисления равны 0 и «NO» — в противном случае.

Примеры

<i>стандартный ввод</i> или <code>input.txt</code>	<i>стандартный вывод</i> или <code>output.txt</code>
10 0 1	NO
8 1 2	YES

Задача В. Браслет

Имя входного файла:	<i>стандартный ввод</i> или <code>input.txt</code>
Имя выходного файла:	<i>стандартный вывод</i> или <code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Поликарпу подарили подарок — браслет с OLED-экраном по всей поверхности. Экран имеет n пикселей, расположенных вдоль браслета. Каждый пиксель может гореть одним из двух любимых цветов Поликарпа — оранжевым или фиолетовым.

Браслет горит определённой расцветкой в течение дня, но на утро цвета меняются: каждый пиксель загорается произвольным цветом. Поликарпу нравится, когда браслет горит так, что в нём образованы две области, одна из оранжевых пикселей, вторая из фиолетовых. Областью будем называть максимальную по включению последовательность подряд идущих пикселей.

К счастью, экран браслета ещё и сенсорный. Поликарп может нажать на пиксель и область, которой он в данный момент принадлежит, инвертируется по цвету. То есть, оранжевый становится фиолетовым, и наоборот.

Поликарп показал вам новую утреннюю раскраску экрана и просит вас о помощи. Он хочет изменить её касаниями так, чтобы было ровно две области и чтобы разница между их размерами была как можно меньше. Формально, если оранжевая область имеет размер x , а фиолетовая — y , то Поликарп хочет минимизировать $|x - y|$.

Формат входных данных

Единственная строка содержит последовательность длиной от 2 до 5×10^6 , состоящую из цифр «0» и «1», которые обозначают фиолетовый и оранжевый цвета соответственно.

Гарантируется, что оба цвета встречаются в последовательности хотя бы по одному разу.

Формат выходных данных

Выведите целое число — минимальную возможную разность.

Примеры

<i>стандартный ввод</i> или <code>input.txt</code>	<i>стандартный вывод</i> или <code>output.txt</code>
1011100110	0
101	1

Замечание

В первом примере в одном из способов нужно коснуться 2-го, а потом 9-го пикселя, после чего получатся две области по 5 пикселей.

Во втором примере экран уже показывает две области и ничего изменить нельзя.

Задача С. Головоломка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Это интерактивная задача.

Поликарп очень хочет научиться хорошо играть в шахматы. Для этого он решает различные шахматные головоломки. Сейчас, например, перед ним стоит задача по заданным позициям черного короля и двух белых ладей, поставить мат не более чем за 50 ходов. Напомним, что

- Шахом называется ситуация, в котором король находится под боем, то есть на следующем ходу хотя бы одна фигура противника сможет взять короля.
- Матом — позиция, в которой король находится под боем и не может никуда походить.
- Пат — позиция, в которой королю ничего не угрожает, но он не может сделать ход.
- Ладья ходит на произвольное не нулевое количество клеток по горизонтали и вертикали, при этом она не может перескакивать через другие фигуры.
- Король может походить в любую из соседних клеток, если та не находится под боем.

Вам предстоит решить данную головоломку, учитывая, что белые ходят первыми.

Протокол взаимодействия

В самом начале в первых трех строках находятся по 2 целых числа $K_x, K_y, A_x, A_y, B_x, B_y$ ($1 \leq K_x, K_y, A_x, A_y, B_x, B_y \leq 8$) — позиции короля и двух ладей на шахматной доске (x — номер строки, y — номер столбца). Гарантируется, что король не находится под боем.

Если ход за вами, то вы должны вывести его в формате «`name x y`», где «`name`» — идентификатор ладьи («`first`» или «`second`»), а x и y — новая позиция указанной ладьи.

После чего компьютер запишет позицию, в которую сместится король, или 0 0, если игра завершена.

Если вы не смогли поставить мат за 50 ходов, один из ваших ходов является некорректным или король срубил вашу ладью, вам будет засчитан неправильный ответ на тест.

Пример

стандартный ввод	стандартный вывод
1 1	first 5 2
5 6	second 7 1
7 3	
2 1	
0 0	

Замечание

Для корректной работы программы после каждой операции вывода данных вам необходимо выводить перенос строки, а также очищать буфер вывода, то есть делать следующие операции:

- В языке Pascal: `flush(output);`
- В C/C++: `fflush(stdout)` или `cout.flush();`
- В Java: `System.out.flush();`
- В Python: `sys.stdout.flush()` из библиотеки `sys`;
- В C#: `Console.Out.Flush();`

Задача D. Годнота

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дана последовательность a из n различных чисел. Назовём отрезок $[l, r]$ *годным*, если крайние элементы с этого отрезка (a_l и a_r) являются максимальным и минимальным элементами с отрезка.

Ваша задача — посчитать годноту последовательности a , то есть количество таких $1 \leq l \leq r \leq n$, что отрезок $[l, r]$ является годным.

Формат входных данных

Первая строка содержит целое число n — размер последовательности ($1 \leq n \leq 10^6$).

Вторая строка содержит n различных чисел a_i , разделённых пробелом ($1 \leq a_i \leq n$).

Формат выходных данных

Выведите единственное число — ответ на задачу.

Примеры

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

Задача E. Юный робототехник

Имя входного файла: *стандартный ввод* или `input.txt`
Имя выходного файла: *стандартный вывод* или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вова – начинающий инженер, в школе он занимается в кружке робототехники. Сейчас он закончил сборку своего робота и собирается его тестировать.

Он подготовил поле, являющееся прямоугольником со сторонами A и B метров. Робот на этом поле в каждый момент времени может перемещаться только в одном из четырёх направлений, параллельных сторонам прямоугольника. Скорость робота составляет один метр в секунду.

Вова собирается проверить заложенную в робота программу достижения цели. Вова выбирает две случайные точки внутри прямоугольника: в первую он помещает робота, а во вторую роботу нужно попасть. Робот будет двигаться так, что это займёт минимальное время.

Точки из прямоугольника Вова выбирает каждый раз равновероятно и независимо.

Будем считать, что робот является материальной точкой, и что смена направления, разгон и торможение происходят мгновенно.

Вова собирается делать такой тест сегодня много раз, поэтому ему интересно, сколько понадобится энергии для робота, которая зависит от времени перемещения. Определите математическое ожидание времени, необходимого роботу для достижения цели за один тестовый запуск.

Формат входных данных

Первая строка содержит два целых числа A и B — размеры прямоугольника ($1 \leq A, B \leq 1000$).

Формат выходных данных

В единственную строку выведите ожидаемое время.

Выведенный ответ будет считаться верным, если абсолютная или относительная погрешность не превосходит 10^{-6} .

Пример

<i>стандартный ввод</i> или <code>input.txt</code>	<i>стандартный вывод</i> или <code>output.txt</code>
2 2	1.333333333

Задача F. Разложение на множители

Имя входного файла: *стандартный ввод* или `input.txt`
Имя выходного файла: *стандартный вывод* или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Назовём *представлением* целого числа N такую последовательность целых чисел a_i что их произведение равно N и при этом, $2 \leq a_i$.

По заданному числу N вам необходимо вычислить $F(N)$ — количество различных представлений числа N , а также вывести лексикографически K -е представление N .

Например, все представления $N = 12$, упорядоченные лексикографически:

$$12 = 2 \times 2 \times 3$$

$$12 = 2 \times 3 \times 2$$

$$12 = 2 \times 6$$

$$12 = 3 \times 2 \times 2$$

$$12 = 3 \times 4$$

$$12 = 4 \times 3$$

$$12 = 6 \times 2$$

$$12 = 12$$

Таким образом $F(12) = 8$.

Формат входных данных

В первой строке содержится целое число N ($2 \leq N \leq 10^{12}$).

В следующих строках содержатся целые числа K ($1 \leq K \leq 10^{18}$), по одному в строке. Ввод заканчивается числом 0. Всего чисел K в одном тесте не более 1000.

Гарантируется, что $K \leq F(N)$.

Формат выходных данных

В первую строку выведите целое число $F(N)$ — количество различных представлений числа N .

Далее для каждого K в отдельной строке выведите K -е представление числа N , разделяя в нём числа пробелами.

Примеры

<i>стандартный ввод</i> или <code>input.txt</code>	<i>стандартный вывод</i> или <code>output.txt</code>
12	8
1	2 2 3
2	2 3 2
3	2 6
4	3 2 2
5	3 4
6	4 3
7	6 2
8	12
0	
2	1
1	2
0	

Задача G. Петрович

Имя входного файла: *стандартный ввод* или `input.txt`
Имя выходного файла: *стандартный вывод* или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В дачный сезон бизнес Петровича бьёт ключом. Точнее водой — он зарабатывает поливкой дачных участков.

У него есть машина, оснащённая баком для воды и бензонасосом. На носу машины установлен распылитель, из которого вода бьёт фонтаном и поливает всё в радиусе R , включая саму машину и Петровича. Процедура запуска и остановки насоса сама по себе связана с потерей воды, поэтому лишний раз включать и выключать его не желательно. Причём когда работает бензонасос, то не работает двигатель автомобиля, то есть поливать в движении не получится.

В конце дня поступил заказ на поливку прямоугольного участка размером A на B . Воды осталось на один запуск бензонасоса. Петрович может подогнать и расположить машину как угодно возле участка, но заезжать на сам участок клиент категорически запретил. При этом он не возражает, если вода будет попадать за пределы участка.

Руководствуясь известной поговоркой «площадь — деньги», Петрович хочет рассчитать максимальную площадь участка, которую он сможет покрыть.

Формат входных данных

Ввод состоит из трёх целых чисел: A, B, R . $1 \leq A, B \leq 10000$; $1 \leq R \leq 20000$

Формат выходных данных

Вывести одно число: максимальную площадь участка, которую сможет полить Петрович.

Ответ будет считаться верным, если его относительная или абсолютная погрешность не превышает 10^{-6} .

Пример

<i>стандартный ввод</i> или <code>input.txt</code>	<i>стандартный вывод</i> или <code>output.txt</code>
2 2 1	1.570796326794897

Задача Н. Опечатки

Имя входного файла: *стандартный ввод* или `input.txt`
Имя выходного файла: *стандартный вывод* или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Известно, что человек легко читает слова, написанные с ошибками. Например, если переставить буквы в произвольном порядке, но сохранить первую и последнюю буквы, то читатель может даже не заметить подвоха.

Дано слово, измените его в соответствии с описанными правилами если возможно (новое слово должно отличаться от исходного).

Формат входных данных

Единственная строка содержит одно слово длиной от 1 до 10^5 включительно, состоящее из заглавных букв латинского алфавита.

Формат выходных данных

Вывести изменённое слово. Если возможны несколько правильных ответов, вывести любой из них. Если изменить слово невозможно, вывести фразу `NO SOLIUTON`.

Примеры

<i>стандартный ввод</i> или <code>input.txt</code>	<i>стандартный вывод</i> или <code>output.txt</code>
ACCERTED	ATCEPCED
ОКО	NO SOLIUTON

Задача I. Return of the ????

Имя входного файла: *стандартный ввод* или `input.txt`
Имя выходного файла: *стандартный вывод* или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Некоторые герои некоторых историй очень похожи. Например, им нужно появиться, пострадать, уйти, а потом вернуться и победить. Но после победы нет времени праздновать — необходимо наладить транспортную систему в своём месте событий.

Королю нужно было соединить N городов в далёком-далёком королевстве, а Джедаю — N звёздных систем в далёкой-далёкой галактике. Эти объекты пронумерованы целыми числами от 1 до N . И Король и Джедай должны были проложить минимальное количество двусторонних дорог (у Короля — это тропы, а у Джедая — гиперпространственные туннели) в своём месте действий так, чтобы объекты стали **связными** — из каждого можно было добраться до каждого по созданным дорогам. В начале ни одной дороги не существовало.

У Короля был следующий алгоритм объединения Королевства: в начале он выбрал случайный город и основывал в нём штаб. Далее $N - 1$ раз раз он выбирал случайный непосещённый город, посещал его, потом выбирал случайный ранее посещённый город и велел проложить тропу между этими двумя городами.

Джедай действовал иначе: пока Галактика не являлась связной, он выбирал случайную пару несвязных между собой звёздных систем и соединял их гиперпространственным туннелем.

Любой выбор любого героя был **равновероятным и независимым**.

Вам в руки попала карта, отображающая результат работы одного из героев — Короля или Джедая. На ней лишь изображены N точек и $N - 1$ отрезков, соединяющих их. Надпись же на карте частично стёрлась: «Return of the ????». Определите, кто это был — Король или Джедай.

Формат входных данных

Первая строка содержит целое число N ($10^4 \leq N \leq 10^5$).

В каждой из следующих $N - 1$ строк содержится по два целых числа a_i и b_i ($1 \leq a_i, b_i \leq N$), означающих, что объекты с такими номерами соединены дорогой.

Формат выходных данных

Выведите «King», если это было возвращение Короля, или «Jedi», если Джедай.

Примеры

<i>стандартный ввод</i> или <code>input.txt</code>	<i>стандартный вывод</i> или <code>output.txt</code>
5 1 2 2 4 3 5 5 2	King
5 2 1 4 2 5 3 2 5	Jedi

Замечание

Приведённые примеры не удовлетворяют ограничениям на N и даны лишь для демонстрации ввода и вывода. На данные тесты оба варианта ответов будут считаться верными.

Задача J. Инверсия в таблице

Имя входного файла:	<i>стандартный ввод</i> или <code>input.txt</code>
Имя выходного файла:	<i>стандартный вывод</i> или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Программист Вася устроился в стартап `ACM Soft`, где разрабатывают новый табличный редактор, который в недалеком будущем вытеснит все аналоги, как платные, так и свободно распространяемые.

В редакторе по умолчанию все ячейки заполнены нулями. Пользователи вручную (а особо продвинутые - при помощи макроса) делают N команд на инверсию значений строк и столбцов (замену нулей на единицы и единиц на нули). После этого пользователи делают M запросов на расчёт количества единиц в выбранном диапазоне ячеек таблицы.

Помогите Васе решить эту непростую задачу.

Формат входных данных

В первой строке N ($1 \leq N \leq 10^5$) — количество команд на инверсию.

В следующих N строках тип команды `R` или `C`, пробел и K ($1 \leq K \leq 10^9$) — номер строки или столбца, который нужно инвертировать. Для команды типа `R` инвертируется строка с указанным номером, для команды типа `C` — столбец.

Гарантируется, что номера строк, как и столбцов, различны.

В следующей строке число M ($1 \leq M \leq 10^5$) — количество запросов.

В следующих M строках по четыре числа r_1, c_1, r_2, c_2 ($1 \leq r_1, c_1, r_2, c_2 \leq 10^9$; $r_1 \leq r_2$; $c_1 \leq c_2$) — координаты левого верхнего и правого нижнего углов диапазона ячеек.

Формат выходных данных

Для каждого из M запросов в отдельной строке вывести количество единиц, которое находится в диапазоне ячеек, указанном в запросе.

Пример

<i>стандартный ввод</i> или <code>input.txt</code>	<i>стандартный вывод</i> или <code>output.txt</code>
5	14
C 6	3
C 2	
R 4	
C 4	
R 2	
2	
1 1 4 7	
6 3 8 5	

Задача К. Проект

Имя входного файла: *стандартный ввод* или `input.txt`
Имя выходного файла: *стандартный вывод* или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вася — руководитель проекта. Над его проектом работает команда разработчиков. У Васи есть данные, кто из сотрудников над какой задачей работает, когда эта задача начинается и когда заканчивается. Сотрудник может работать над несколькими задачами параллельно.

Помогите Васе понять загруженность сотрудников. Для этого нужно вывести список членов команды по убыванию максимального количества параллельных задач сотрудника, а также само это количество. При совпадении количества параллельных задач первым идёт сотрудник с лексикографически меньшим именем.

Формат входных данных

Первая строка содержит число N — количество задач проекта ($1 \leq N \leq 100$).

Далее следует N строк. Каждая строка описывает одну задачу проекта и состоит из четверки T, S, E, A , разделенной пробелами, где:

- T — название задачи проекта длиной от 1 до 100 букв латинского алфавита;
- S — дата начала выполнения задачи в формате «YYYY-MM-DD», например 2018-10-05;
- E — дата окончания выполнения задачи в формате «YYYY-MM-DD», например 2018-10-26.
Гарантируется, что $S \leq E$;
- A — имя сотрудника, выполняющего задачу, длиной от 1 до 100 букв латинского алфавита.

Все названия задач проекта различаются. Имена сотрудников и названия задач начинаются с заглавной буквы, остальные буквы строчные. Даты задач лежат в диапазоне от 1 января до 31 декабря 2018 года включительно. Задача выполняется со дня старта по день завершения включительно.

Формат выходных данных

Вывести список всех участников проекта по убыванию их загруженности, разделенный переводами строк. Через пробел от имени сотрудника выводится его максимальное количество параллельных задач в проекте.

Примеры

<i>стандартный ввод</i> или <code>input.txt</code>	<i>стандартный вывод</i> или <code>output.txt</code>
4 Design 2018-10-15 2018-10-27 Ivan Code 2018-10-27 2018-10-31 Masha Test 2018-10-15 2018-10-31 Ivan Check 2018-11-15 2018-11-30 Vasily	Ivan 2 Masha 1 Vasily 1
2 Something 2018-01-01 2018-01-31 Petr Nothing 2018-01-01 2018-01-31 Petr	Petr 2

Замечание

В первом примере у `Ivan` в плане две параллельные задачи `Design` и `Test` с 2018-10-15 по 2018-10-27, соответственно, он больше загружен и идёт первым. У `Masha` и `Vasily` не более одной задачи одновременно, но `Masha` выводится выше в списке из-за лексикографически меньшего имени.