

Задача А. Математическое сложение

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

Иван решил подготовиться к контрольной работе по решению уравнений в целых числах. Он заметил, что все задания в контрольной имеют следующий вид:

- Вам даны два положительных целых числа u и v , найдите любую пару целых чисел (необязательно положительных) x и y таких, что:

$$\frac{x}{u} + \frac{y}{v} = \frac{x+y}{u+v}.$$

- Решение $x = 0, y = 0$ запрещено, поэтому вы должны найти любое решение с $(x, y) \neq (0, 0)$.

Пожалуйста, помогите Ивану решить некоторые уравнения такого вида.

Формат входных данных

Первая строка содержит единственное целое число t ($1 \leq t \leq 10^3$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Единственная строка описания каждого набора входных данных содержит два целых числа u и v ($1 \leq u, v \leq 10^9$) — параметры уравнения.

Формат выходных данных

Для каждого набора входных данных выведите два целых числа x и y — возможное решение уравнения. Должно быть выполнено, что $-10^{18} \leq x, y \leq 10^{18}$ и $(x, y) \neq (0, 0)$.

Можно показать, что ответ всегда существует. Если существует несколько возможных решений, вы можете вывести любое из них.

Пример

стандартный ввод	стандартный вывод
4	-1 1
1 1	-4 9
2 3	-18 50
3 5	-4 9
6 9	

Замечание

В первом наборе входных данных: $\frac{-1}{1} + \frac{1}{1} = 0 = \frac{-1+1}{1+1}$.

Во втором наборе входных данных: $\frac{-4}{2} + \frac{9}{3} = 1 = \frac{-4+9}{2+3}$.

В третьем наборе входных данных: $\frac{-18}{3} + \frac{50}{5} = 4 = \frac{-18+50}{3+5}$.

В четвертом наборе входных данных: $\frac{-4}{6} + \frac{9}{9} = \frac{1}{3} = \frac{-4+9}{6+9}$.

Задача В. Покраска прямоугольников

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

У Васи есть **красный** клетчатый прямоугольник размера $n \times m$. Но он ему не нравится. Поэтому Вася несколько раз разрежет исходный или любой другой, получившийся в ходе разрезания, прямоугольник на две части вдоль одной из линий сетки. Он может сделать эту операцию сколько угодно раз.

В результате он получит набор прямоугольников. Прямоугольники 1×1 **запрещены**.

Еще Вася умеет красить клеточки в **синий** цвет. Он хочет, чтобы каждый прямоугольник из набора был раскрашен так, что соседние по стороне клетки имели разные цвета.

Какое минимальное количество клеточек Васе придется покрасить?

Формат входных данных

В первой строке находится единственное целое число t ($1 \leq t \leq 10^3$) — количество наборов входных данных. Далее следуют описания наборов входных данных.

Единственная строка описания каждого набора входных данных содержит два целых числа n , m ($1 \leq n, m \leq 3 \cdot 10^4$, $n \cdot m \geq 2$).

Формат выходных данных

Для каждого набора входных данных выведите единственное целое число — минимальное количество клеток, которое придется покрасить Васе.

Пример

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

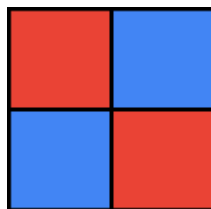
Замечание

Приведенные ниже картинки показывают, как прямоугольник может быть разрезан, и какие клетки могут быть покрашены синим после этого.

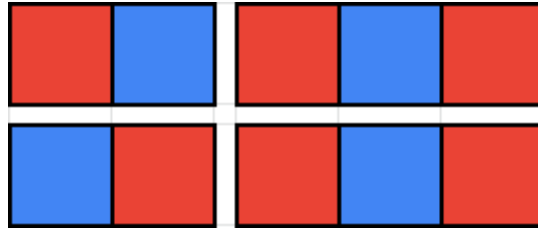
В первом наборе входных данных:



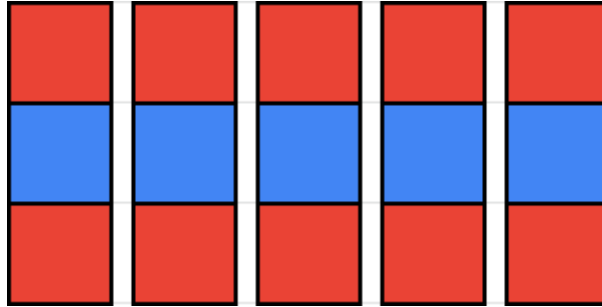
Во втором наборе входных данных:



В третьем наборе входных данных:



В четвертом наборе входных данных:



Задача С. Два массива

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

Вам даны два массива из целых чисел a_1, a_2, \dots, a_n и b_1, b_2, \dots, b_n .

Определим следующее преобразование массива a :

1. Выберите любое целое неотрицательное число k , удовлетворяющее условию $0 \leq k \leq n$.
2. Выберите k различных индексов $1 \leq i_1 < i_2 < \dots < i_k \leq n$.
3. Прибавьте 1 ко всем элементам $a_{i_1}, a_{i_2}, \dots, a_{i_k}$, остальные элементы массива a остаются неизменными.
4. Переставьте элементы массива a в любом порядке.

Можно ли применить какое-то преобразование массива a **ровно один раз**, и получить массив, равный b ?

Формат входных данных

Первая строка содержит единственное целое число t ($1 \leq t \leq 100$) — количество наборов входных данных. Далее следуют описания наборов входных данных.

Первая строка описания каждого набора входных данных содержит единственное целое число n ($1 \leq n \leq 100$) — длину массивов a и b .

Вторая строка описания каждого набора входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($-100 \leq a_i \leq 100$).

Третья строка описания каждого набора входных данных содержит n целых чисел b_1, b_2, \dots, b_n ($-100 \leq b_i \leq 100$).

Формат выходных данных

Для каждого набора входных данных выведите «YES» (без кавычек), если существует такое преобразование массива a , в результате которого получится массив, равный b . Выведите «NO» (без кавычек) иначе.

Можно выводить каждый символ в любом регистре (верхнем или нижнем).

Пример

стандартный ввод	стандартный вывод
3	YES
3	NO
-1 1 0	YES
0 0 2	
1	
0	
2	
5	
1 2 3 4 5	
1 2 3 4 5	

Замечание

В первом наборе входных данных можно сделать следующее преобразование:

- Выбираем $k = 2$.
- Выбираем $i_1 = 1, i_2 = 2$.

- Добавляем 1 к a_1 and a_2 . В результате получится массив $[0, 2, 0]$.
- Меняем элементы на второй и третьей позициях.

Во втором наборе входных данных нет подходящего преобразования.

В третьем наборе можно выбрать $k = 0$ и не менять порядок элементов.

Задача D. Угадай перестановку

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

Это интерактивная задача.

У жюри была тождественная перестановка a длины n ($a_i = i$).

Жюри выбрало три целых числа i, j, k такие, что $1 \leq i < j < k \leq n$, $j - i > 1$. После этого жюри развернуло отрезки $[i, j - 1]$ и $[j, k]$ в последовательности a .

Разворачивание подотрезка $[l, r]$ последовательности a означает переворот подотрезка элементов a_l, a_{l+1}, \dots, a_r в последовательности, то есть a_l меняется местами с a_r , a_{l+1} с a_{r-1} , и т. д..

Вам дано число n . Вы должны найти i, j, k с помощью некоторых запросов.

За один запрос вы можете выбрать два целых числа l и r ($1 \leq l \leq r \leq n$) и узнать количество инверсий на подотрезке $[l, r]$ последовательности a . Жюри скажет вам количество пар (i, j) таких, что $l \leq i < j \leq r$ и $a_i > a_j$.

Найдите числа i, j, k , выбранные жюри, сделав не более 40 запросов.

Жюри фиксирует числа i, j и k до запуска вашей программы и не меняет их в зависимости от ваших запросов.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится единственное целое число t ($1 \leq t \leq 100$) — количество наборов входных данных. Далее следуют наборы входных данных.

В единственной строке для каждого набора входных данных дано целое число n ($4 \leq n \leq 10^9$). После его считывания, вы должны перейти к процессу взаимодействия с программой жюри и делать запросы. После того, как вы дали ответ, вы должны:

- Закончить работу программы, если это был последний набор входных данных.
- Иначе начать работу со следующим набором.

Протокол взаимодействия

Чтобы спросить количество инверсий на отрезке $[l, r]$, выведите «? l r», где ($1 \leq l \leq r \leq n$). Вы можете сделать не более 40 таких запросов в каждом тестовом случае. В ответ вы получите целое число x .

- Если $x = -1$, ваша программа сделала некорректный запрос или превысила допустимое количество запросов для данного набора. В таком случае ваша программа должна немедленно завершиться (иначе решение может получить произвольный вердикт вместо **Неправильный ответ**).
- Иначе x равен количеству инверсий на подотрезке $[l, r]$ последовательности a .

Чтобы дать ответ, выведите «! i j k», где i, j, k — числа, которые, как вы считаете, загадало жюри. После этого вы должны перейти к следующему набору или завершить программу.

После вывода запроса или ответа не забудьте вывести перевод строки и сбросить буфер выходного потока. В противном случае вы получите вердикт Решение «зависло». Для сброса буфера используйте:

- `fflush(stdout)` или `cout.flush()` в C++;
- `System.out.flush()` в Java;
- `flush(output)` в Pascal;

- `stdout.flush()` в Python;
- смотрите документацию для других языков.

Взломы

Чтобы сделать взлом, используйте следующий формат:

Первая строка содержит единственное целое число t ($1 \leq t \leq 100$) — количество наборов входных данных.

Каждая из следующих t строк содержит четыре целых числа n, i, j, k ($4 \leq n \leq 10^9$, $1 \leq i < j < k \leq n, j - i > 1$).

Пример

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

Замечание

В первом наборе входных данных $i = 1, j = 3, k = 5$, поэтому последовательность a равна $[2, 1, 5, 4, 3]$.

Во втором наборе входных данных $i = 2, j = 4, k = 5$, поэтому последовательность a равна $[1, 3, 2, 5, 4]$.

Задача E. Игра с камнями

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Боб решил отдохнуть от домашних заданий по матанализу и придумал для себя игру.

Игра происходит на последовательности куч камней, которая может быть представлена как массив s_1, \dots, s_k , где s_i — количество камней в i -й куче. За один ход Боб выбирает две соседние непустые кучи i и $i + 1$ и убирает по одному камню из каждой. Если куча стала пустой, то её соседи **не становятся соседними**. Игра заканчивается, когда у Боба нет доступных ходов. Боб считает, что он выиграл, если в конце игры все кучи стали пустыми.

Мы считаем последовательность куч выигрышной, если Боб может начать с неё и выиграть после некоторой последовательности ходов.

Вам дана последовательность a_1, \dots, a_n , посчитайте количество выигрышных подотрезков a . Другими словами, найдите количество отрезков $[l, r]$ ($1 \leq l \leq r \leq n$) таких, что a_l, a_{l+1}, \dots, a_r является выигрышной последовательностью куч.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится единственное целое число t ($1 \leq t \leq 3 \cdot 10^5$) — количество наборов входных данных. Далее следуют описания наборов входных данных.

Первая строка описания каждого набора входных данных содержит единственное целое число n ($1 \leq n \leq 3 \cdot 10^5$).

Вторая строка описания каждого набора входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$).

Гарантируется, что сумма n по всем наборам входных данных не превосходит $3 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных выведите единственное целое число — ответ на задачу.

Пример

стандартный ввод	стандартный вывод
6	1
2	0
2 2	4
3	2
1 2 3	1
4	3
1 1 1 1	
4	
1 2 2 1	
4	
1 2 1 2	
8	
1 2 1 2 1 2 1 2	

Замечание

В первом наборе входных данных Боб не может выиграть с подотрезками длины 1, так как нет ни одной пары соседних куч для массива длины 1.

Во втором наборе входных данных каждый подотрезок не является выигрышным.

В четвертом наборе входных данных подотрезок $[1, 4]$ выигрышный, потому что Боб может сделать ходы со следующими парами соседних кучек: $(2, 3)$, $(1, 2)$, $(3, 4)$. Другой выигрышный подотрезок это $[2, 3]$.

Задача F. Странная НОП

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам даны n строк s_1, s_2, \dots, s_n , состоящих из строчных и заглавных латинских букв. Более того, каждый символ в каждой строке встречается **не более двух раз**. Найдите самую длинную общую подпоследовательность этих строк.

Строка t является подпоследовательностью строки s , если t может быть получена из s удалением нескольких (возможно, ни одного или всех) символов.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится единственное целое число t ($1 \leq t \leq 5$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка описания каждого набора входных данных содержит единственное целое число n ($2 \leq n \leq 10$) — количество строк.

В следующих n строках содержатся строки s_i . Все s_i непустые, состоят из строчных и заглавных латинских букв, и никакой символ не встречается ни в одной строке более чем два раза.

Формат выходных данных

Для каждого набора входных данных выведите ответ в двух строках.

В первой строке выведите длину самой длинной общей подпоследовательности.

Во второй — саму подпоследовательность. Если существует несколько самых длинных общих подпоследовательностей, выведите любую из них.

Пример

стандартный ввод	стандартный вывод
4	1
2	A
ABC	0
CBA	
2	3
bacab	ace
defed	3
3	coc
abcde	
aBcDe	
ace	
2	
codeforces	
technocup	

Замечание

В первом наборе входных данных одна из самых длинных общих подпоследовательностей это «A». Нет ни одной общей подпоследовательности длины 2.

Во втором наборе входных данных множества символов строк не пересекаются, поэтому никакая непустая строка не может быть общей подпоследовательностью.

Задача G. Допустимые отрезки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Вам даны n различных точек p_1, p_2, \dots, p_n на плоскости, а также положительное целое число R .

Найдите количество пар индексов (i, j) таких, что $1 \leq i < j \leq n$, и для всех возможных k ($1 \leq k \leq n$) расстояние от точки p_k до отрезка, образованного точками p_i и p_j , не больше R .

Формат входных данных

Первая строка содержит два целых числа n, R ($1 \leq n \leq 3000, 1 \leq R \leq 10^5$) — количество точек на плоскости и максимальное возможное расстояние от точки до отрезка.

Каждая из следующих n строк содержит два целых числа x_i, y_i ($-10^5 \leq x_i, y_i \leq 10^5$), которые задают i -ю точку $p_i = (x_i, y_i)$. Все точки различны.

Гарантируется, что ответ на задачу не изменится при изменении значения параметра R на величину не более 10^{-2} .

Формат выходных данных

Выведите единственное целое число — количество подходящих пар (i, j) .

Примеры

стандартный ввод	стандартный вывод
4 2 0 1 0 -1 3 0 -3 0	1
3 3 1 -1 -1 -1 0 1	3

Замечание

В первом примере подходит единственная пара точек $(-3, 0), (3, 0)$. Расстояние до отрезка между этими точками от точек $(0, 1)$ и $(0, -1)$ равно 1, что меньше $R = 2$.

Во втором примере подходят все возможные пары точек.