

Задача А. Среднее арифметическое

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

У студента Владислава есть структура данных «мультимножество». В отличие от множества, мультимножество может содержать несколько экземпляров одного и того же элемента.

Владислав выполняет в заданном порядке n операций одного из следующих двух видов:

- добавить число x в мультимножество;
- удалить один экземпляр числа x из мультимножества (при этом гарантируется, что число x есть в мультимножестве хотя бы в единственном экземпляре).

После каждой операции ему нужно вывести среднее арифметическое всех чисел в мультимножестве (среднее арифметическое пустого мультимножества считается равным нулю). К сожалению, у него возникли проблемы с этим, и похоже, что он не справится вовремя. Вам надо снова помочь ему и вывести правильные ответы.

Формат входных данных

В первой строке содержится единственное число n ($1 \leq n \leq 200000$) — количество операций.

В каждой из следующих n строк дан сначала вид операции: «+» для операции добавления и «-» для операции удаления, а затем целое число x ($0 \leq x \leq 10^9$).

Формат выходных данных

Выведите n строк, в i -й строке должно содержаться одно число с плавающей точкой — среднее арифметическое всех чисел в мультимножестве после i -й операции. Абсолютная или относительная погрешность для каждого из ответов не должна превышать 10^{-9} .

Примеры

стандартный ввод	стандартный вывод
9	1.0
+ 1	1.5
+ 2	1.66666666667
+ 2	1.25
+ 0	1.6
+ 3	1.75
- 1	2.2
+ 4	2.25
- 2	3.0
- 0	
5	999999001.0
+ 999999001	999999001.5
+ 999999002	999999002.0
+ 999999003	999999002.5
+ 999999004	999999003.0
+ 999999005	

Задача В. Удвоения

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

У студента Владислава есть структура данных «множество». Эта структура данных примечательна тем, что содержит каждый свой элемент в единственном экземпляре, и при попытке добавить в множество элемент, который там уже есть, ничего не происходит.

Изначально Владислав добавил в свое множество n чисел $a_1, \dots, a_n$. Теперь он собирается 10^{10} раз выполнить следующую операцию: извлечь из множества минимальный элемент, и добавить в множество этот элемент, умноженный на 2.

Владислав боится, что к концу конкурса не успеет проделать эти операции, поэтому вы должны сообщить ему, сколько чисел будет в множестве после того, как он все же закончит свою процедуру.

Формат входных данных

В первой строке содержится целое число n ($1 \leq n \leq 200000$) — изначальное количество чисел в множестве.

Во второй строке содержится n целых чисел от 1 до 10^9 — изначальные элементы множества $a_1, \dots, a_n$.

Формат выходных данных

Выведите единственное число — итоговый размер множества.

Примеры

стандартный ввод	стандартный вывод
3 2 3 4	2
3 20 10 5	1

Задача С. Скобочки

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

Правильная скобочная последовательность определяется следующим образом:

- пустая строка — правильная скобочная последовательность;
- если s — правильная скобочная последовательность, то $\langle s \rangle$ — тоже правильная скобочная последовательность;
- если s и t — правильные скобочные последовательности, то $\langle st \rangle$ — тоже правильная скобочная последовательность.

Дана скобочная последовательность. Гарантируется, что ее длина четная, а количество открывающих и закрывающих скобок в ней совпадает. За одну операцию вы можете выбрать две различные позиции i и j , и поменять местами символы на этих позициях. За какое минимальное количество таких операций можно превратить данную скобочную последовательность в правильную?

Формат входных данных

В единственной строке содержится непустая скобочная последовательность, она состоит только из символов «(» и «)», а ее длина не превышает 10^6 .

Формат выходных данных

Выведите единственное число — минимальное количество операций.

Примеры

стандартный ввод	стандартный вывод
))((	1
(())	0

Задача D. Без одного символа

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

Школьник Вася придумал строку s длины n . Она ему очень понравилась, поэтому ему показалось, что память об этой строке обязательно должна быть увековечена на заборе, мимо которого он каждый день проходит по дороге в школу.

Вася понимает, что писать эту строку как есть слишком рискованно, поэтому в i -й из следующих n дней он будет записывать на заборе строку без i -го символа. Т. е. в первый день он запишет строку $s_2s_3 \dots s_n$, во второй — строку $s_1s_3s_4 \dots s_n$, а в последний, n -й день — строку $s_1s_2 \dots s_{n-1}$.

Сколько различных строк будет написано на заборе?

Формат входных данных

Во входных данных содержится строка s . Она состоит из строчных латинских букв, а ее длина n находится в промежутке от 2 до 200000.

Формат выходных данных

Выведите единственное целое число — сколько различных строк получится у Васи.

Примеры

стандартный ввод	стандартный вывод
abca	4
zzz	1

Задача Е. Кубики

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

Есть трехмерное пространство из кубиков, размера $x \times y \times z$. Некоторые из этих кубиков пустые, а некоторые заполнены и образуют фигуру (не обязательно связную).

Вам даны три проекции этой фигуры (вид слева, вид спереди и вид сверху). Требуется построить фигуру из максимально возможного числа кубиков, которая давала бы точно такие же проекции.

Формат входных данных

В первой строке содержатся три целых числа x, y, z ($1 \leq x, y, z \leq 100$) — линейные размеры пространства (если смотреть спереди, то x — это ширина, y — глубина, а z — высота).

Затем дан вид слева (z строк по y символов), а после него пустая строка.

Затем дан вид спереди (z строк по x символов), а после него пустая строка.

Затем дан вид сверху (y строк по x символов).

Каждый символ — либо «#», что соответствует заполненной клетке в проекции, либо «.», что соответствует пустой клетке. Гарантируется, что во входных данных есть хотя бы один символ «#».

Формат выходных данных

В первой строке выведите «YES» или «NO», в зависимости от того, существует ли такая фигура.

В случае положительного ответа выведите описание фигуры. Оно должно состоять из z блоков, каждый блок должен описывать кубики фигуры на одной и той же высоте, в порядке сверху вниз. В каждом блоке должно быть y строк по x символов. Блоки должны быть отделены друг от друга пустой строкой.

Примеры

стандартный ввод	стандартный вывод
4 3 2 ### #.# #### #### #### #.# ###.	YES #### #.# ###. #### ###.
3 3 3 ### ### ### ### #.# ### ### ### ###	YES ### ### ### #.# #.# #.# ### ### ###
3 3 3 #.. .#. ..# .#. ..# #.. ..# #.. .#.	NO
3 3 3 #.. .#. ..# .#. #.. ..# .#. #.. ..#	YES .#. #..#

Задача F. Фальшивая монета

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

У вас есть n монет. Вы точно знаете, что одна из них фальшивая и отличается по весу, но неизвестно, на сколько и в какую сторону.

Вы хотите с помощью серии взвешиваний определить, какая же из монет фальшивая. У вас есть весы с двумя чашами. На каждую чашу весов можно класть сколько угодно монет (но количества на первой и второй чашах должны совпадать).

Протокол взаимодействия

Это интерактивная задача. Ваша программа должна общаться с программой жюри, используя для этого стандартные потоки ввода и вывода.

В самом начале вашей программе сообщается единственное число n ($3 \leq n \leq 1000$) — количество монет.

После этого вы можете сделать не более 9 взвешиваний. Чтобы провести взвешивание, нужно вывести три строки. В первой строке должен содержаться символ «?», а затем через пробел целое положительное число k — количество монет, которое вы положите на каждую чашу весов. Во второй строке должны содержаться k чисел — номера монет, которые вы положите на первую чашу весов. В третьей строке аналогичным образом должны содержаться k номеров монет, которые вы положите на вторую чашу весов. Все выведенные номера монет во второй и третьей строках должны быть в промежутке от 1 до n и быть различны.

В ответ на такой запрос вам приходит результат взвешивания — один из трех символов «<», «=» или «>».

После того, как вы однозначно установите фальшивую монету, выведите символ «!», а затем через пробел целое число от 1 до n — ее номер. После этого ваша программа должна завершиться.

Пример

стандартный ввод	стандартный вывод
6	? 3 1 2 3 4 5 6
>	? 2 1 4 2 5
<	? 1 3 4
=	! 2

Замечание

Обратите внимание, что каждое выведенное вами сообщение должно завершаться переводом строки. Также после вывода каждого сообщения ваша программа должна очищать потоковый буфер, чтобы выведенная вами информация дошла до программы жюри: например, это делают вызовы «`fflush(stdout)`» или «`cout.flush()`» в C++, «`System.out.flush()`» в Java, «`Console.Out.Flush()`» в C#, «`flush(output)`» в Pascal, «`sys.stdout.flush()`» в Python.

Пустые строки в примере приведены лишь для удобства, чтобы было лучше понятно, в каком

порядке выводятся сообщения. При решении задачи вам не нужно выводить пустые строки, и программа жюри тоже не будет выводить пустые строки.

Задача G. Нажатия на кнопки

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

Есть n кнопок. На каждой кнопке есть дисплей, на котором отображается число. Изначально на i -й кнопке отображается число a_i . Если нажать на i -ю кнопку, число на ней уменьшится на 1, а числа на всех остальных кнопках увеличатся на 1.

Вам нужно, чтобы на i -й кнопке отображалось число b_i . Можете ли вы этого добиться, и если да, на какие кнопки надо нажимать?

Формат входных данных

В первой строке содержится целое число n ($1 \leq n \leq 200000$) — количество кнопок.

Вторая строка содержит n целых чисел a_i ($-10^9 \leq a_i \leq 10^9$) — числа, которые изначально отображаются на кнопках.

Третья строка содержит n целых чисел b_i ($-10^9 \leq b_i \leq 10^9$) — числа, которые требуется получить.

Формат выходных данных

Если достигнуть требуемой конфигурации не получится, выведите единственное число -1 .

Иначе выведите n целых чисел $c_1, \dots, c_n$, где c_i — количество раз, которое надо нажать на i -ю кнопку.

Примеры

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

Задача Н. Саратовская дилемма

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

Каждый год лучшие команды Самарского университета ездят в Саратов на четвертьфинал чемпионата мира по программированию ICPC. Как правило, вся делегация снимает одну большую квартиру, тут-то и возникает дилемма.

В квартире есть a односпальных и b двуспальных кроватей. В делегации n человек, и среди них есть люди, которые согласны спать с кем-либо вместе на двуспальной кровати, а есть люди, которые против этого.

Зная эти данные, распределите спальные места среди участников делегации.

Формат входных данных

В первой строке даны три целых числа n , a и b ($1 \leq n \leq 200000$, $0 \leq a + b \leq 200000$) — количество человек, а также количество односпальных и двуспальных кроватей.

Вторая строка содержит n символов «0» и «1» — согласен ли i -й человек спать с кем-то вместе на двуспальной кровати.

Формат выходных данных

В первой строке выведите «YES» или «NO», в зависимости от того, можно ли распределить людей по спальным местам или нет.

В случае положительного ответа выведите еще $a + b$ строк. Первые a строк должны содержать по одному числу — номеру человека, который будет спать на соответствующей односпальной кровати (или 0, если никто не будет спать на этой кровати). Следующие b строк должны содержать по два числа — номера людей, которые будут спать вдвоем на соответствующей двуспальной кровати (если одно или оба спальных места останутся свободными, для этих мест должно быть выведено число 0).

Если возможных ответов несколько, разрешается вывести любой из них.

Примеры

стандартный ввод	стандартный вывод
7 3 2 1111000	YES 5 6 7 1 2 3 4
7 3 2 1011000	NO
5 3 2 10110	YES 0 1 2 3 4 5 0

Задача I. Контесты

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

Кирилл готовится пройти на финал ICPC. Впереди n контестов, и Кирилл уже вычислил, что на i -м из них он сможет увеличить свой рейтинг на a_i . Однако после некоторых контестов Кирилл так сильно устает, что не способен участвовать в нескольких следующих. Конкретно, если он решал i -й контест, он должен будет пропустить как минимум следующие b_i контестов.

Вычислите максимально возможное увеличение рейтинга Кирилла после n контестов.

Формат входных данных

В первой строке содержится целое число n ($1 \leq n \leq 200000$) — количество контестов.

В каждой из следующих n строк даны два целых числа a_i и b_i ($-10^9 \leq a_i \leq 10^9, 0 \leq b_i \leq 200000$) — увеличение рейтинга после i -го контеста, если Кирилл поучаствует в нем, и сколько контестов после i -го Кирилл обязан будет пропустить.

Формат выходных данных

Выведите единственное число — максимально возможное увеличение рейтинга после n контестов.

Примеры

стандартный ввод	стандартный вывод
3 20 0 100 2 30 0	120
3 20 1 100 2 30 0	100

Задача J. Игра с перестановкой

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

Перестановкой из n элементов называется последовательность целых чисел от 1 до n , в которой все числа различны.

Есть перестановка p из n элементов. Два игрока играют q раундов в следующую игру: они одновременно называют случайное число от 1 до n (если их числа совпадают, они делают это еще раз, пока их числа не окажутся различными), а побеждает тот игрок, чье число встречается в перестановке p раньше.

К сожалению, в некоторых случаях определение победителя занимает слишком много времени. Для каждого раунда выведите, какой же игрок победил в этом раунде.

Формат входных данных

В первой строке содержится целое число n ($2 \leq n \leq 200000$) — количество элементов в перестановке.

Вторая строка содержит n различных целых чисел от 1 до n — элементы перестановки $p_1, \dots, p_n$.

В третьей строке содержится целое число q ($1 \leq q \leq 200000$) — количество раундов.

В каждой из следующих q строк содержится два различных целых числа a_j и b_j ($1 \leq a_j, b_j \leq n, a_j \neq b_j$) — числа, названные в j -м раунде первым и вторым игроком соответственно.

Формат выходных данных

Для каждого из q раундов, в зависимости от того, какой игрок в нем победил, выведите «First» или «Second».

Пример

стандартный ввод	стандартный вывод
3	Second
2 3 1	Second
4	First
1 2	First
1 3	
2 1	
2 3	

Задача К. Обход дерева

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

Деревом называется связный граф без циклов. Количество ребер в дереве всегда на 1 меньше, чем количество вершин.

Есть дерево с n вершинами. Вы должны пройти по каждому его ребру ровно один раз. Вы можете начать обход с любой вершины, а также в любой момент телепортироваться из любой вершины в любую другую.

Какое минимальное количество раз придется телепортироваться, чтобы выполнить свой план?

Формат входных данных

В первой строке содержится целое число n ($1 \leq n \leq 200000$) — количество вершин в дереве.

В каждой из следующих $(n - 1)$ строк содержатся по два различных целых числа от 1 до n — пары вершин, соединенных ребрами.

Формат выходных данных

Выведите единственное число — минимальное количество раз, которое придется телепортироваться.

Примеры

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

Задача L. Перевероты

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

Есть две строки s и t . Разрешается любое количество раз выполнять следующую операцию: выбрать две позиции i и j в строке s ($i < j$), и перевернуть подстроку $s_{i...j}$. Т. е. i -й символ поменяется местами с j -м, $(i + 1)$ -й с $(j - 1)$ -м, и т. д.

Можно ли с помощью таких операций превратить s в t ?

Формат входных данных

Во входных данных содержатся две строки s и t , каждая из них непустая, имеет длину до 200000 и состоит из строчных латинских букв.

Формат выходных данных

Выведите «YES» или «NO», в зависимости от того, можно ли превратить s в t или нет.

Примеры

стандартный ввод	стандартный вывод
abac acba	YES
xxx yuu	NO
abcd abc	NO

Задача М. Выпуклая оболочка

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

Многоугольник называется выпуклым, если любая прямая, полностью содержащая одну из его сторон, оставляет остальные точки многоугольника строго по одну сторону от себя.

Выпуклой оболочкой множества точек на координатной плоскости назовем наименьший выпуклый многоугольник, содержащий все заданные точки. При этом какие-то из заданных точек оказываются на границе выпуклой оболочки, а какие-то — строго внутри нее.

Представьте себе доску, в которую вбито — но не по самую шляпку — много гвоздей. Возьмите верёвку, свяжите на ней скользящую петлю (лассо) и набросьте её на доску, а потом затяните. Верёвка окружает все гвозди, но касается она только некоторых, самых внешних. Те гвозди, которых она касается, составляют выпуклую оболочку для всей группы гвоздей. — Википедия

Даны 4 точки. Найдите, сколько из них лежат на границе их выпуклой оболочки.

Формат входных данных

Во входных данных содержатся 4 строки, в каждой из них по два целых числа — координаты x_i, y_i ($-100 \leq x_i, y_i \leq 100$). Гарантируется, что никакие две точки не совпадают, и никакие три не лежат на одной прямой.

Формат выходных данных

Выведите количество точек из заданных, которые лежат на границе их выпуклой оболочки.

Примеры

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