

Задача А. Рассеянность [D]

Имя входного файла: `absent.in` или стандартный ввод
Имя выходного файла: `absent.out` или стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На построении рассеянного батальона была дана команда “На первый-второй рассчитайсь”. В полном соответствии со своим названием, батальон не совсем справился с этой задачей. Вы должны выяснить, сколько ошибок сделали бойцы.

Формат входного файла

В первой строке вводится число солдат $n \leq 5000$. Во второй строке n чисел — единиц или двоек, обозначающих, что сказал каждый солдат.

Формат выходного файла

Выведите одно число: количество солдат, которые сделали ошибку.

Пример

<code>absent.in</code> или стандартный ввод	<code>absent.out</code> или стандартный вывод
7 1 2 1 2 1 2 2	1
6 2 1 2 1 2 1	6

Задача В. Важные числа [D, C']

Имя входного файла: `number.in` или стандартный ввод
Имя выходного файла: `number.out` или стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Артур играет сам с собой в такую игру: каждое утро он выбирает три числа, которые называет “главными числами дня”. После этого, все числа, которые он встречает, он проверяет на то, являются ли они “стоящими внимания”. Число считается “стоящим внимания”, если оно делится ровно на два из “главных чисел”.

Сегодня Артур загадал довольно большие числа в качестве главных, поэтому ему лень самому проверять все числа. Напишите программу, которая ему поможет.

Формат входного файла

В первой строке введены главные числа сегодняшнего дня. Во второй — одно число, про которое нужно проверить, является ли оно стоящим внимания. Все четыре числа натуральные и не превосходят 10^9 .

Формат выходного файла

Выведите YES, если число стоит внимания. В противном случае выведите NO.

Пример

<code>number.in</code> или стандартный ввод	<code>number.out</code> или стандартный вывод
3 5 7 42	YES
3 5 7 100	NO

Примечание

В первом примере число 42 делится на 3 и на 7, т. е. ровно на два из трёх главных чисел. Во втором примере число 100 делится только на одно главное число — на 5.

Задача С. Найдите два числа [С']

Имя входного файла: `find.in` или стандартный ввод
Имя выходного файла: `find.out` или стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Мальчик Петя принёс домой массив чисел. Петя не говорит нам, откуда он взял этот массив. Но Петя утверждает, что там, откуда он его принёс, бывают только массивы, в которых найдутся два числа, сумма которых составляет 1000. Причём такая пара чисел в массиве всегда есть ровно одна.

Теперь Петя хочет найти эти два числа в массиве. Что самое обидное, Петя не говорит нам, зачем ему это нужно. Но мы всё равно поможем ему и проведём за него поисковые работы.

Формат входного файла

Первая строка содержит число N — количество чисел в массиве ($2 \leq N \leq 1000$). Во второй строке подаются N чисел — элементы массива. Элементы массива разделены пробелами. Все элементы массива по модулю не превышают 10^9 .

Формат выходного файла

Выведите через пробел два числа этого массива, сумма которых составляет 1000. Выводить числа надо в том же порядке, в котором они стоят во входных данных. Гарантируется, что требуемая пара чисел существует и единственна.

Пример

<code>find.in</code> или стандартный ввод	<code>find.out</code> или стандартный вывод
6 100 200 300 400 500 600	400 600

Задача D. Склад [C]

Имя входного файла: `store.in` или стандартный ввод
Имя выходного файла: `store.out` или стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Склад представляет собой здание с f этажами (с 1 по f) и r комнатами на каждом этаже (на каждом этаже комнаты нумеруются от 1 до r).

На склад привезли новую партию товаров, упакованных в коробки. Для каждой коробки сказано, в какую комнату на каком этаже ее надо ставить.

Охранник хочет узнать, какие комнаты нуждаются в особой охране. Он считает, что комната нуждается в особой охране, если в ней лежит строго больше, чем k коробок.

Формат входного файла

В первой строке вводятся два числа — количество этажей f ($1 \leq f \leq 100$) и количество комнат на каждом этаже r ($1 \leq r \leq 100$).

Во второй строке два числа — общее количество n ($n \leq 500$) привезенных на склад коробок и число k ($0 < k \leq 20$) коробок, после которого комната требует особой охраны.

Далее следуют n строк в каждой из которых написаны два числа: номер этажа и номер комнаты, куда ставится очередная коробка.

Формат выходного файла

Одно число — количество комнат, нуждающихся в особой охране.

Пример

<code>store.in</code> или стандартный ввод	<code>store.out</code> или стандартный вывод
2 3	1
4 2	
1 1	
1 1	
1 1	
2 3	

Задача Е. Корректность почтового адреса [С]

Имя входного файла: email.in или стандартный ввод
Имя выходного файла: email.out или стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Разработчикам приложений время от времени приходится собирать регистрационные данные от пользователей. При этом некоторые данные принято автоматически проверять на корректность. Мы предлагаем вам проверить на корректность адрес электронной почты.

Будем считать, что корректный адрес устроен так: LOCAL@DOMAIN. Мы считаем, что он состоит из части LOCAL, которая стоит до знака @, и части DOMAIN, которая стоит после него. Часть LOCAL состоит из одного или более *слов*, которые отделяются друг от друга ровно одной точкой. Часть DOMAIN состоит из двух или более *слов*, которые отделяются друг от друга ровно одной точкой. Каждое *слово* — это последовательность символов, которая может состоять из маленьких и больших латинских букв, десятичных арабских цифр, а также дефиса «-» и знака подчёркивания «_». Каждое *слово* состоит хотя бы из одного символа.

Формат входного файла

В единственной строке входного файла дана последовательность символов с кодами от 32 до 127. В этой последовательности не более 100 символов.

Формат выходного файла

Выведите YES, если содержимое входного файла является корректным адресом электронной почты. В противном случае выведите NO.

Пример

email.in или стандартный ввод	email.out или стандартный вывод
postmaster@example.com	YES
***@gmail.com	NO

Задача F. Петя и полоска [B]

Имя входного файла: `segment.in`
Имя выходного файла: `segment.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

У Пети есть полоска бумаги, разделенная на N клеток. Он хочет раскрасить каждую клетку в синий, красный или зеленый цвет.

Кроме этого, Пете интересны одноцветные отрезки. Петя называет одноцветным отрезком несколько подряд идущих клеток, раскрашенных в один цвет и ограниченных с обеих сторон клетками другого цвета или границами полоски.

Петя хочет, чтобы все синие одноцветные отрезки имели длину A клеток, все красные одноцветные отрезки имели длину B клеток, а все зеленые одноцветные отрезки имели длину C клеток.

Пете интересно, сколькими способами он может раскрасить полоску таким образом. Помогите ему и вычислите это количество. Поскольку оно может быть очень большим, выведите его по модулю $10^9 + 7$.

Формат входного файла

В первой строке записано четыре целых числа, разделенных пробелами — N , A , B и C ($1 \leq A, B, C, N \leq 10^6$, $A, B, C \leq N$).

Формат выходного файла

Выведите ответ на задачу по модулю $10^9 + 7$.

Примеры

<code>segment.in</code>	<code>segment.out</code>
3 1 2 3	3
3 1 1 1	12
5 1 1 1	48
3 2 2 2	0

Задача G. Загадочный алгоритм — 1 [B]

Имя входного файла: algo1.in
Имя выходного файла: algo1.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан загадочный алгоритм, принимающий на вход число N и $N - 1$ пар целых чисел из промежутка от 1 до N включительно, и выводящий последовательность чисел от 1 до N некоторой длины. Реализации алгоритма на нескольких языках программирования доступны для скачивания на странице

<http://lksh.ru/sis/2013/vstupit/>.

На вход вашей программе подается число N , число P и последовательность из P чисел. Вы должны придумать такой набор из $N - 1$ пары чисел (с учетом порядка), чтобы загадочный алгоритм, получив на вход число N и этот набор пар чисел, вывел данную последовательность.

Гарантируется, что ответ всегда существует.

Гарантируется, что во входной последовательности чисел каждое число от 1 до N встречается хотя бы один раз.

Формат входного файла

В первой строке записано два целых числа, разделенных пробелом — N и P — длина последовательности, выведенной алгоритмом ($1 \leq N \leq 100000$, $1 \leq P \leq 200000$).

Во второй строке — последовательность из P целых чисел, выведенная алгоритмом.

Формат выходного файла

Выведите $N - 1$ пар целых чисел, каждое из которых лежит в интервале от 1 до N — искомый набор в требуемом порядке. Каждую пару выведите на отдельной строке, числа внутри пары разделяйте пробелом.

Примеры

algo1.in	algo1.out
2 3 1 2 1	1 2
3 5 1 2 1 3 1	1 2 1 3
3 5 1 3 1 2 1	1 3 1 2

Задача Н. Загадочный алгоритм — 2 [В]

Имя входного файла: algo2.in
Имя выходного файла: algo2.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан некоторый алгоритм, принимающий на вход массив целых чисел и выводящий одно целое число. Реализации алгоритма на нескольких языках программирования доступны для скачивания на странице <http://lksh.ru/sis/2013/vstupit/>.

На вход вашей программе подается число N и результат работы программы P . Выведите любую перестановку чисел от 1 до N , на которой данный алгоритм выведет число P , либо определите, что такой перестановки не существует.

Перестановкой чисел от 1 до N называется последовательность чисел длины N , в которой каждое целое число от 1 до N включительно встречается ровно один раз.

Формат входного файла

В первой строке записано два целых числа, разделенных пробелом — N и P ($1 \leq N \leq 1000$, $0 \leq P \leq 10^9$).

Формат выходного файла

В первой строке выведите YES, если ответ существует, и NO в противном случае.

Если ответ существует, во второй строке выведите N чисел, разделенных пробелами — искомую перестановку.

Примеры

algo2.in	algo2.out
2 1	YES 2 1
2 2	NO

Задача I. Палиндромы [A]

Имя входного файла: palindr.in
Имя выходного файла: palindr.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Строка называется палиндромом, если она одинаково читается как слева направо, так и справа налево. Например, „abba“ — палиндром, а „опах“ — нет. Для строки α будем обозначать $\alpha[i..j]$ ее подстроку длины $j - i + 1$ с i -й по j -ю позицию включительно (позиции нумеруются с 1). Для заданной строки α длины N ($1 \leq N \leq 100\,000$) требуется подсчитать число q пар (i, j) , $1 \leq i < j \leq n$, таких что $\alpha[i..j]$ является палиндромом.

Формат входного файла

Входной файл содержит одну строку α длины N , состоящую из маленьких латинских букв.

Формат выходного файла

В выходной файл выведите искомое число q .

Примеры

palindr.in	palindr.out
aaa	3
abba	2
опах	0

Задача J. K-ый максимум [A]

Имя входного файла: kth.in
Имя выходного файла: kth.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам необходимо реализовать структуру данных, позволяющую отвечать на запросы следующего вида:

- $1\ x$ — добавить в множество число x . Гарантируется, что непосредственно перед выполнением этого запроса в множестве не содержится числа x .
- $-1\ x$ — удалить из множества число x . Гарантируется, что непосредственно перед выполнением этого запроса в множестве содержится число x .
- $0\ x$ — вывести x -ое в порядке убывания число в множестве. Гарантируется, что непосредственно перед выполнением этого запроса в множестве содержится не менее x чисел.

До выполнения всех запросов множество пусто.

Формат входного файла

В первой строке записано целое число N — количество запросов ($1 \leq N \leq 100000$).

В каждой из следующих N строк записано по два целых числа — описание очередного запроса. Запросы необходимо выполнять в том порядке, в каком они заданы во входном файле. Все числа во входном файле не превосходят по модулю 10^9 .

Формат выходного файла

Для каждого запроса вида $0\ x$ выведите одно число — ответ на этот запрос. Ответы нужно выводить в том порядке, в котором соответствующие запросы заданы во входном файле. Каждый ответ нужно вывести на отдельной строке.

Примеры

kth.in	kth.out
9	3
1 1	2
1 2	1
1 3	
0 1	
0 2	
0 3	
-1 1	
-1 2	
-1 3	

Задача К. Точки и рамка [A]

Имя входного файла: `rect.in`
Имя выходного файла: `rect.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На плоскости нарисована ломаная из N вершин. i -ая вершина ломаной имеет координаты (x_i, y_i) .

У Васи есть прямоугольная рамка размером $W \times H$. Он хочет выбрать несколько подряд идущих вершин ломаной и заключить их в рамку. Кроме этого, Вася хочет располагать стороны рамки только параллельно соответствующим осям координат (т.е. сторона длины W должна быть параллельна оси x , а сторона длины H — параллельна оси y).

Более формально, Вася выбирает числа X и Y — координаты левого нижнего угла рамки. Тогда вершина (x_i, y_i) лежит внутри рамки, если $X \leq x_i \leq X + W$ и $Y \leq y_i \leq Y + H$.

Помогите Васе определить, какое наибольшее количество подряд идущих вершин он может заключить в рамку.

Формат входного файла

В первой строке записано три целых числа, разделенных пробелом — N , W и H ($1 \leq N \leq 100000$, $0 \leq W, H \leq 10^9$).

В каждой из следующих N строк записано по два числа — координаты очередной вершины, разделенные пробелом. Все координаты — целые числа, не превосходящие по модулю 10^9 .

Формат выходного файла

Выведите два числа, разделенных пробелом — максимальное количество подряд идущих вершин, которые Вася может заключить в рамку, а также номер первой вершины, начиная с которой, Вася может заключить указанное количество вершин в рамку.

Если вариантов ответа несколько, выведите любой.

Примеры

<code>rect.in</code>	<code>rect.out</code>
3 2 2 0 2 -2 3 -3 2	2 1