

Задача 1А Кто ходит в гости по утрам

Имя входного файла: k.in

Имя выходного файла: k.out

Ограничение по времени: 2 секунды

Ограничение по памяти: 256 мегабайт

В Чудесном Лесу живут N различных персонажей, у каждого из которых есть свой собственный домик. Следуя заветам одного из самых известных лесных персонажей, Винни-Пуха, каждый житель считает необходимым проснуться с утра пораньше, умыться, одеться и пойти в гости к кому-либо. Разумеется, чтобы поступить не просто мудро, а очень мудро и не потратить слишком много времени на дорогу, персонаж отправится не к кому-нибудь, а к своему соседу, то есть к тому из жителей, домик которого находится к данному персонажу на наименьшем возможном расстоянии. Нетрудно понять, что хозяина этого домика не окажется дома, поскольку он тоже воспользуется правилом Винни-Пуха. Лишь по этой причине некому будет ни крикнуть “Ура!”, ни обрадоваться гостям. Если вдруг окажется, что несколько домиков расположены на минимальном расстоянии от персонажа, то он выберет для похода в гости домик с наименьшим номером. Ваша задача – определить какие персонажи собираются у каждого домика.

Формат входного файла

В первой строке задается количество персонажей N ($2 \leq N \leq 100\,000$). В каждой из последующих N строк задаются по два числа – координаты точки на плоскости, в которой расположен домик соответствующего персонажа. Все координаты – целые неотрицательные числа, не превосходящие 10^9 .

Формат выходного файла

Выведите N строк. i -ая строка должна содержать число i , за которым следует двоеточие и далее в порядке возрастания номера персонажей, которые придут в гости в i -ый домик.

Пример

k.in	k.out
6	1: 2 3
0 0	2: 1
1 0	3:
0 1	4: 5
3 3	5: 4 6
2 2	6:
3 1	

Задача 1В Ёлка на новый год

Имя входного файла: spruce.in

Имя выходного файла: spruce.out

Ограничение по времени: 2 секунды

Ограничение по памяти: 256 мегабайт

В городе Манерске готовятся к проведению нового года. На главной площади города нужно поставить большую красивую ёлку. Ответственное задание по выбору ёлки поручили Васе, известному сотруднику какого-то там исследовательского института.

Лесничий по имени Андрей знает, как Вася выбирает деревья. Конечно, он хочет найти самую высокую ёлку. Но лес слишком большой, и Вася не хочет заблудиться в нём. Поэтому перед тем, как идти исследовать деревья, он спросит у Андрея, какую бы ёлку ему спилить и унести на площадь. Потом Вася залезет на эту ёлку и посмотрит, что растёт в окружке. С ёлки высоты h метров виден весь лес в квадрате $2h$ метров \times $2h$ метров, с центром в этой ёлке.

Если Вася увидит ёлку большей высоты, чем та, на которой он сейчас, он перейдёт к такой ёлке (самой высокой из тех, которые увидит) и заберётся на неё. Это будет продолжаться, пока Вася не окажется на ёлке, с которой он не увидит более высоких ёлок. Тогда он слезет с этой ёлки, спилит её и отнесёт на площадь.

Лесничий Андрей знает всё про ёлки и лес: координаты и высоту каждой из n ёлок. Помогите ему сохранить лес – сделать так, чтобы Вася спилит ёлку как можно меньшей высоты.

Формат входного файла

Входной файл состоит из нескольких тестов. В первой строке каждого теста содержится число n ($1 \leq n \leq 50\,000$). Далее следуют n строк по три числа в каждой. В $(i+1)$ -й строке содержатся числа x_i , y_i и h_i . Это координаты и высота

i -й ёлки. Все числа во входном файле целые и не превосходят 10^9 по модулю. В одном тесте нету двух ёлок одинаковой высоты.

Входной файл завершается строкой с одним нулём, которую обрабатывать не нужно. Сумма n по всем тестам не превосходит 50 000.

Формат выходного файла

Для каждого теста выведите два числа на отдельной строке: номер ёлки, которую Андрей должен показать Васе, и высоту ёлки, которая будет в результате спилена. Ёлки нумеруются в том порядке, в котором они даны во входном файле, начиная с единицы, отдельно для каждого теста.

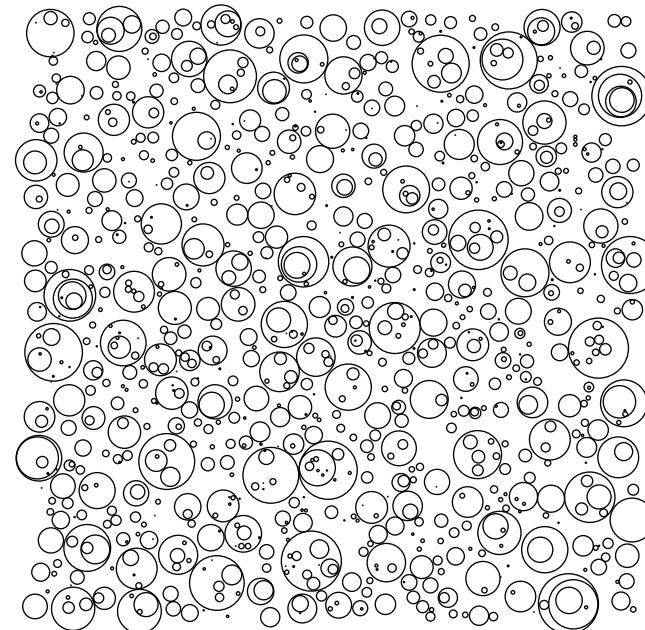
Пример

spruce.in	spruce.out
3	
-1 1 2	2 3
-1 -1 3	
0 3 4	
0	

Задача 1С Circles

Имя входного файла: circles.in
Имя выходного файла: circles.out
Ограничение по времени: 5 seconds
Ограничение по памяти: 64 MB

There are n circles located on the plane. Circles may have common points, but for any two circles their intersection is either a point, or one of the two circles.



Find the total area covered by at least one circle.

Формат входного файла

The first line of the input file contains integer number n ($1 \leq n \leq 100\,000$). The following n lines contain three integers each and describe circles. The i -th circle is described by coordinates of its center x_i and y_i and its radius r_i ($-10^6 \leq x_i, y_i \leq 10^6$, $1 \leq r_i \leq 10^6$).

Формат выходного файла

Output one real number: the total area covered by at least one circle. Your answer must have absolute or relative error of at most 10^{-9} .

Пример

circles.in	circles.out
4	
2 2 2	28.2743338823081391
2 2 1	
5 2 1	
5 5 2	

Задача 1Д Футбол

Имя входного файла:	<code>football.in</code>
Имя выходного файла:	<code>football.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 МВ

На футбольном поле размером $x \times y$ находятся n футболистов. Они уже очень устали и стоят на месте, но ждут, куда упадет мяч, чтобы побежать к нему. Футболист бежит к мячу в том случае, если мяч упал к этому футболисту ближе, чем к любому другому футболисту. Требуется определить для каждого футболиста границы зоны, при попадании в которую он побежит к мячу, если известно, что она представляет собой многоугольник.

Формат входного файла

В первой строке входного файла заданы три целых числа x, y и n ($2 \leq x, y \leq 10^5$, $1 \leq n \leq 1000$). Следующие n строк содержат целые координаты футболистов x_i, y_i ($0 < x_i < x, 0 < y_i < y$). Никакие два футболиста не стоят в одной точке.

Формат выходного файла

В выходной файл выведите n строк. В каждой из строк первое число — количество вершин зоны k_i , далее k_i чисел — координаты вершин x_{ij}, y_{ij} в порядке обхода против часовой стрелки, начиная с самой нижней из самых левых вершин зоны. вещественные числа выводите с максимальной точностью.

Пример

<code>football.in</code>	<code>football.out</code>
4 4 4	4 0 0 2 0 2 2 0 2
1 1	4 0 2 2 2 2 4 0 4
1 3	4 2 0 4 0 4 2 2 2
3 1	4 2 2 4 2 4 4 2 4
3 3	

Задача 1Е Футбол

Имя входного файла:	<code>football.in</code>
Имя выходного файла:	<code>football.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 МВ

На футбольном поле размером $x \times y$ находятся n футболистов. Они уже очень устали и стоят на месте, но ждут, куда упадет мяч, чтобы побежать к нему.

Футболист бежит к мячу в том случае, если мяч упал к этому футболисту ближе, чем к любому другому футболисту. Требуется определить для каждого футболиста границы зоны, при попадании в которую он побежит к мячу, если известно, что она представляет собой многоугольник.

Формат входного файла

В первой строке входного файла заданы три целых числа x, y и n ($2 \leq x, y \leq 10^5$, $1 \leq n \leq 30000$). Следующие n строк содержат целые координаты футболистов x_i, y_i ($0 < x_i < x, 0 < y_i < y$). Никакие два футболиста не стоят в одной точке.

Формат выходного файла

В выходной файл выведите n строк. В каждой из строк первое число — количество вершин зоны k_i , далее k_i чисел — координаты вершин x_{ij}, y_{ij} в порядке обхода против часовой стрелки, начиная с самой нижней из самых левых вершин зоны. вещественные числа выводите с максимальной точностью.

Пример

<code>football.in</code>	<code>football.out</code>
4 4 4	4 0 0 2 0 2 2 0 2
1 1	4 0 2 2 2 2 4 0 4
1 3	4 2 0 4 0 4 2 2 2
3 1	4 2 2 4 2 4 4 2 4
3 3	

Задача 2А Функция

Имя входного файла:	<code>function.in</code>
Имя выходного файла:	<code>function.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

$$\text{Вычислите функцию: } f(n) = \begin{cases} 1 & \text{если } n \leq 2 \\ f(\lfloor 6 * n / 7 \rfloor) + f(\lfloor 2 * n / 3 \rfloor) & \text{если } n \bmod 2 = 1 \\ f(n - 1) + f(n - 3) & \text{если } n \bmod 2 = 0 \end{cases}$$

Формат входного файла

Входные данные содержат натуральное число n ($1 \leq n \leq 10^{12}$).

Формат выходного файла

Выведите значение функции по модулю 2^{32} .

Пример

function.in	function.out
7	10

Задача 2В Котята с пирожками

Имя входного файла: kittens.in
 Имя выходного файла: kittens.out
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 64 мегабайт

Однажды n котят решили покушать пирожков. Однако котят много, поэтому им непросто выбрать начинку, которая всех порадует. Известно, что группа из не более чем k котят всегда может прийти к консенсусу, а вот большая группа обязательно разобъётся на две, принципиально несогласные друг с другом. Котята — существа справедливые, поэтому размеры этих групп будут отличаться не более чем на один.

Группы, размер которых всё ещё окажется больше k после такого разделения, продолжат спорить и разделяться на меньшие по тому же принципу. Определите, сколько групп котят в итоге отправятся за пирожками.

Формат входного файла

В единственной строке входного файла записаны целые числа n и k — количество котят и критический размер группы, соответственно ($1 \leq n, k \leq 10^{18}$). Числа записаны без ведущих нулей.

Формат выходного файла

Выведите единственное целое число — итоговое количество групп. Число должно быть также записано без ведущих нулей.

Пример

kittens.in	kittens.out
17 4	5

Задача 2С Самый длинный путь

Имя входного файла: path.in
 Имя выходного файла: path.out
 Ограничение по времени: 0.5 секунд
 Ограничение по памяти: 256 мебибайт

В данном ориентированном графе найдите самый длинный путь такой, что

каждая вершина графа встречается в нём не более одного раза.

Формат входного файла

В первой строке входного файла заданы через пробел два целых числа n и m ($1 \leq n \leq 22, 0 \leq m \leq 1000$). В следующих m строках заданы рёбра графа в формате $u_i v_i$ — номера начальной и конечной вершин ребра i , соответственно. Граф может содержать петли и кратные рёбра.

Формат выходного файла

В первой строке выходного файла выведите длину искомого пути l . Во второй строке выведите $l + 1$ число через пробел — вершины пути в порядке обхода. Если оптимальных ответов несколько, можно вывести любой из них.

Примеры

path.in	path.out
3 3 1 2 2 3 3 1	2 1 2 3
4 6 1 2 2 1 2 3 2 4 3 2 4 2	2 1 2 4
5 3 3 2 2 2 1 5	1 3 2

Задача 2D Кривые зеркала

Имя входного файла: mirror.in
 Имя выходного файла: mirror.out
 Ограничение по времени: 0.5 секунды
 Ограничение по памяти: 64 Мб

За один залп BFG-9000 уничтожает монстров на трех соседних балкончиках. Балкончики при этом не разрушаются. А монстры погибают. (N -й балкончик

Конкурс группы А1.
МатМех, Весенний семестр 2011

соседствует с первым). После залпа оставшиеся в живых монстры наносят Леониду (главному герою романа) повреждения — по одной единице каждый. Далее следует новый залп и так до тех пор, пока все монстры не погибнут. Требуется определить минимальные повреждения, которые может понести Леонид.

Формат входного файла

Первая строка содержит целое число N , количество балкончиков, на которых монстры заняли круговую оборону ($3 \leq N \leq 30$). Во второй строке даны N целых чисел — количество монстров на каждом балкончике (на каждом не менее 1 и не более 100).

Формат выходного файла

Выведите минимальное количество единиц повреждений.

Пример

mirror.in	mirror.out
7	9
3 4 2 2 1 4 1	

Задача 2Е Сортировка вручную

Имя входного файла: sort.in

Имя выходного файла: sort.out

Ограничение по времени: 2 секунды

Ограничение по памяти: 256 мегабайт

Пришло время расставить книжки на полке. На ней стоят n книг, однозначно пронумерованных целыми числами от 1 до n . Требуется расставить их в порядке возрастания номеров. Вы прекрасно знаете, что *быстрая сортировка* и *сортировка вставками* — быстрые методы сортировки, однако их непросто исполнять вручную. Они эффективны для компьютеров, а не для людей.

Поэтому вы решили сортировать книги, вставляя i -ю книгу на i -ю позицию. Сколько операций вам придётся выполнить для успешной сортировки?

Вот два примера операции:

- $1 \ 3 \ 4 \ 5 \ 2 \rightarrow 1 \ 2 \ 3 \ 4 \ 5$, если мы вставляем 2 на свое место
- $1 \ 3 \ 4 \ 5 \ 2 \rightarrow 1 \ 4 \ 3 \ 5 \ 2$, если мы вставляем 3 на свое место

Формат входного файла

В первой строке входного файла содержится целое число n ($1 \leq n \leq 20$) — количество книг на полке.

Во второй строке записано n различных целых чисел от 1 до n — номера книг в том порядке, в котором они расставлены вначале.

Формат выходного файла

Выведите единственное целое число — минимальное количество операций указанного вида, которое придётся выполнить для сортировки.

Примеры

sort.in	sort.out
3	0
1 2 3	
3	1
2 1 3	
3	2
3 2 1	
9	5
7 2 9 4 6 8 5 3 1	

Задача 2F Перестановки

Имя входного файла: perms.in

Имя выходного файла: perms.out

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Саша и Федя играют в интересную игру. У них есть n кубиков, на которых написаны различные числа от 1 до n . Ребята нарисовали на бумаге n клеточек в ряд и играют по следующим правилам.

Сначала первый игрок выставляет некоторые кубики на клеточки, затем второй игрок выставляет на свободные клетки оставшиеся кубики. После этого первый игрок делает следующие действия: он смотрит, какое число написано на последнем кубике (пусть это число a) и после этого переставляет последние a кубиков в обратном порядке. Эти действия первый игрок повторяет до тех пор, пока последним не станет кубик с числом 1.

Например, пусть у ребят пять кубиков. Если первый игрок поставил второй и третий кубик на третье и пятое место: «..3.2», то второй игрок может расставить оставшиеся кубики так: «41352». В этом случае первому игроку потребуется сделать пять действий: «41325», «52314», «54132», «54123», «54321», после чего игра закончится.

Конкурс группы А1.
МатМех, Весенний семестр 2011

Сейчас первым ходил Саша. Помогите Феде расставить кубики так, чтобы Саша сделал максимально возможное количество действий.

Формат входного файла

Во входном файле содержится число n ($1 \leq n \leq 25$). Следующие n чисел задают расположение кубиков после хода Саши. Число 0 означает, что клетка свободна, число от 1 до n — номер кубика, который стоит в этой клетке. Во входном файле не более 10 нулей.

Формат выходного файла

На первой строке выходного файла выведите максимальное количество действий, которое придется сделать Саше.

На второй строке выведите n чисел от 1 до n , где i -е число означает номер кубика, стоящего в i -й клетке после хода Феди. Если оптимальных решений несколько, выведите любое.

Пример

perms.in	perms.out
5	5
0 0 3 0 2	4 1 3 5 2
2	1
0 0	1 2

Задача 3А ДНК роботов

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

Последние достижения в технологии синтеза ДНК позволили провести эксперимент по созданию биороботов.

Для облегчения задачи создания ПО для управления роботами было принято решение, что их ДНК будет состоять из $M = 2^n$ символов для некоторого $n \geq 2$. Кроме этого, по техническим причинам это будет не обычная строка, а циклическая, то есть её можно начинать читать с любой позиции.

Одной из целей эксперимента является изучение мутаций биороботов. В результате продолжительных наблюдений было найдено много различных видов роботов. Для понимания процесса мутации учёным необходимо решить следующую задачу. Для ДНК двух роботов требуется определить коэффициент их похожести. Он вычисляется, как максимальное количество совпадающих символов

при наилучшем совмещении этих ДНК. Чем больше символов совпадает, тем лучше совмещение.

Требуется написать программу, которая найдёт наилучшее совмещение двух ДНК.

Формат входного файла

В первой строке входного файла записано одно число M ($4 \leq M \leq 131072$). В следующих двух строках записаны ДНК двух роботов. Обе ДНК — строки, состоящие ровно из M символов из множества {'A', 'C', 'G', 'T'}.

Формат выходного файла

В выходной файл выведите два числа — максимальное количество совпадающих символов и значение оптимального сдвига — неотрицательное количество символов второй ДНК, которые необходимо перенести из конца строки в её начало для достижения наилучшего совмещения.

Пример

robots.in	robots.out
16 ACGTACGTACGTACGT CGTACGTACGTACGT	15 1

Задача 3Д Уравнение

Имя входного файла: equation.in
Имя выходного файла: equation.out
Ограничение по времени: 10 секунды
Ограничение по памяти: 256 мегабайт

Дано уравнение вида $X^N + Y^N \equiv Z^N \pmod{M}$.

Требуется для фиксированных N и M найти количество различных решений этого уравнения. Решением назовём такую тройку натуральных чисел (X, Y, Z) , что выполняется:

- $1 \leq X \leq Y < M$
- $1 \leq Z < M$
- $X^N + Y^N \equiv Z^N \pmod{M}$

Формат входного файла

В единственной строке входного файла записаны числа N и M ($1 \leq N \leq 7^7$, $1 \leq M \leq 7^7$).

Формат выходного файла

В выходной файл выведите одно число — ответ на задачу.

Примеры

equation.in	equation.out
1 3	2
2 4	5
3 5	8

Задача 3F АВЛ-деревья

Имя входного файла: **avl.in**

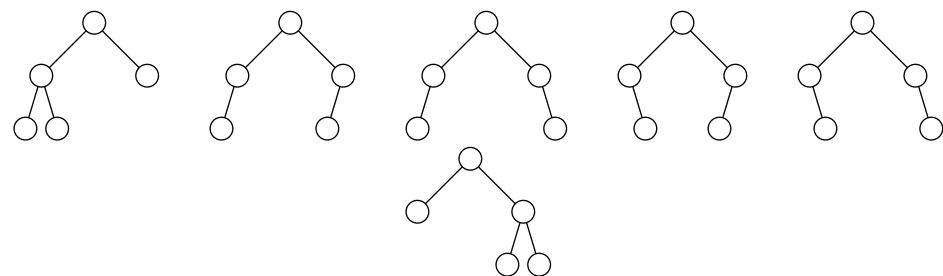
Имя выходного файла: **avl.out**

Ограничение по времени: 5 секунд

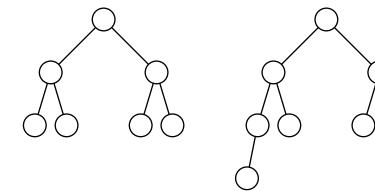
Ограничение по памяти: 256 мегабайт

АВЛ-дерево — сбалансированное по высоте двоичное дерево поиска: для каждой его вершины высота её двух поддеревьев различается не более чем на 1. АВЛ-деревья названы по первым буквам фамилий их изобретателей, Г. М. Адельсона-Вельского и Е. М. Ландиса.

Для фиксированного количества вершин может существовать несколько АВЛ-деревьев. Например, существует шесть АВЛ-деревьев, состоящих из пяти вершин.



Также деревья с одинаковым количеством вершин могут иметь различную высоту. Например, существуют деревья из семи вершин с высотами 2 и 3 соответственно.



Требуется по заданным n и h найти количество АВЛ-деревьев, состоящих из n вершин и имеющих высоту h . Так как ответ может быть очень большим, требуется найти остаток от деления искомого количества на 786433.

Формат входного файла

Во входном файле даны числа n и h ($1 \leq n \leq 65535$, $0 \leq h \leq 15$).

Формат выходного файла

Выполните одно число — остаток от деления количества АВЛ-деревьев, состоящих из n вершин и имеющих высоту h , на 786433.

Пример

avl.in	avl.out
7 3	16

786433 — простое число, $786433 = 3 \cdot 2^{18} + 1$.

Задача 3Н Раздвоение

Имя входного файла: **real.in**

Имя выходного файла: **real.out**

Ограничение по времени: 2 секунды

Ограничение по памяти: 256 мегабайт

Обозначим две последовательности действительных чисел $x(k)$ и $y(k)$. Определим последовательность комплексных чисел $z(k)$: $z(k) = x(k) + iy(k)$.

Пусть $FFT_N(k, z) = \sum_{n=0}^{N-1} z_n e^{2\pi i kn/N}$. Аналогичным образом определяются $FFT_N(k, x)$ и $FFT_N(k, y)$.

Требуется по вычисленным значениям $FFT_N(k, z)$ восстановить значения $FFT_N(k, x)$ и $FFT_N(k, y)$.

Формат входного файла

В первой строке входного файла записано целое число N ($1 \leq N \leq 2^{30}$, N является степенью двойки). Далее следуют целые неотрицательные числа A, B, C, D, E, F , не превосходящие 1000. Для экономии времени ввода значения $FFT_N(k, z)$ нужно будет вычислять по следующим формулам:

$$FFT_N(k, z).real = ((A + B \cdot k) \text{ xor } (C \cdot k)) \cdot 10^{-3},$$

$$FFT_N(k, z).imag = ((D + E \cdot k) \text{ xor } (F \cdot k)) \cdot 10^{-3},$$

где $FFT_N(k, z).real$ и $FFT_N(k, z).imag$ — действительная и мнимая части соответственно.

Затем дано число M — количество запросов ($1 \leq M \leq 10^5$). Далее следуют M целых чисел q_j ($0 \leq q_j < N$).

Формат выходного файла

В выходной файл выведите M строк. В j -ой строке — значения $FFT_N(q_j, x)$ и $FFT_N(q_j, y)$. Значения должны отличаться от правильных не более, чем на 10^{-4} .

Примеры

real.in	real.out
2	1.0 0.0 0.0 0.0
1000 0 0 0 0 0	1.0 0.0 0.0 0.0
2	
0 1	
4	0.000 0.000 0.500 0.000
0 100 300 500 100 200	0.504 0.140 0.516 0.176
4	0.656 0.000 0.812 0.000
0 1 2 3	0.504 -0.140 0.516 -0.176
1048576	540.737 -1587.741 1589.778
999 998 997 996 995 994	539.689
3	2404.809 531.421 1359.578
17 239239 2011	1569.751
	3678.277 -523.243 526.382
	3664.887

Задача ЗК Дуэль

Имя входного файла:	duel.in
Имя выходного файла:	duel.out
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Двое дуэлянтов решили выбрать в качестве места проведения поединка тёмную аллею. Вдоль этой аллеи растёт n деревьев и кустов. Расстояние между соседними объектами равно одному метру. Дуэль решили проводить по следующим правилам. Некоторое дерево выбирается в качестве стартовой точки. Затем два дерева, находящихся на одинаковом расстоянии от исходного, отмечаются как места для стрельбы. Дуэлянты начинают движение от стартовой точки в противоположных направлениях. Когда соперники достигают отмеченных деревьев, они разворачиваются и начинают стрелять друг в друга.

Дана схема расположения деревьев вдоль аллеи. Требуется определить количество способов выбрать стартовую точку и места для стрельбы согласно правилам дуэли.

Формат входного файла

Во входном файле содержится одна строка, состоящая из символов ‘0’ и ‘1’ — схема аллеи. Деревья обозначаются символом ‘1’, кусты — символом ‘0’. Длина строки не превосходит 100000 символов.

Формат выходного файла

Выполните количество способов выбрать стартовую точку и места для стрельбы согласно правилам дуэли.

Примеры

duel.in	duel.out
101010101	4
101001	0

В первом примере возможны следующие конфигурации дуэли (стартовое дерево и деревья для стрельбы выделены жирным шрифтом): **101010101**, **101010101**, **101010101** и **101010101**.

Задача 3Л Произведение

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

Требуется найти произведение двух целых чисел.

Формат входного файла

В каждой из двух строк входного файла записано целое число, состоящее не более чем из 239000 цифр.

Формат выходного файла

В выходной файл выведите одно целое число — произведение этих чисел.

Примеры

multiply.in	multiply.out
2	4
2	
-1	-1
1	

Задача 4А Переворачивания

Имя входного файла: reverse.in
Имя выходного файла: reverse.out
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мебибайт

Учитель физкультуры школы с углубленным изучением предметов уже давно научился считать суммарный рост всех учеников, находящихся в ряду на позициях от l до r . Но дети играют с ним злую шутку. В некоторый момент дети на позициях l по r меняются местами. Учитель заметил, что у детей не очень богатая фантазия, поэтому они всегда «переворачивают» этот отрезок, т. е. l меняется с r , $l + 1$ меняется с $r - 1$ и так далее. Но учитель решил не ругать детей за их хулиганство, а все равно посчитать суммарный рост на всех запланированных отрезках.

Формат входного файла

В первой строке записано два числа n и m ($1 \leq n, m \leq 200\,000$) — количество детей в ряду и количество событий, произошедших за все время. Во второй строке задано n натуральных чисел — рост каждого школьника в порядке следования

в ряду. Рост детей не превосходит $2 \cdot 10^5$. Далее в m строках задано описание событий: три числа q, l, r в каждой строке ($0 \leq q \leq 1$, $1 \leq l \leq r \leq n$). Число q показывает тип события: 0 показывает необходимость посчитать и вывести суммарный рост школьников на отрезке $[l, r]$; 1 показывает то, что дети на отрезке $[l, r]$ «перевернули» свой отрезок. Все числа во входном файле целые.

Формат выходного файла

Для каждого события типа 0 выведите единственное число на отдельной строке — ответ на этот запрос.

Пример

reverse.in	reverse.out
5 6	15
1 2 3 4 5	9
0 1 5	8
0 2 4	7
1 2 4	10
0 1 3	
0 4 5	
0 3 5	

Задача 4В И снова сумма...

Имя входного файла: sum2.in
Имя выходного файла: sum2.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Реализуйте структуру данных, которая поддерживает множество S целых чисел, с которым разрешается производить следующие операции:

- $add(i)$ — добавить в множество S число i (если он там уже есть, то множество не меняется);
- $sum(l, r)$ — вывести сумму всех элементов x из S , которые удовлетворяют неравенству $l \leq x \leq r$.

Формат входного файла

Исходно множество S пусто. Первая строка входного файла содержит n — количество операций ($1 \leq n \leq 300\,000$). Следующие n строк содержат операции. Каждая операция имеет вид либо « $+ i$ », либо « $? l r$ ». Операция « $? l r$ » задает запрос $sum(l, r)$.

Если операция « $+ i$ » идет во входном файле в начале или после другой операции « $+$ », то она задает операцию $add(i)$. Если же она идет после запроса «?», и результат этого запроса был y , то выполняется операция $add((i + y) \bmod 10^9)$.

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до 10^9 .

Формат выходного файла

Для каждого запроса выведите одно число — ответ на запрос.

Пример

sum2.in	sum2.out
6	3
+ 1	
+ 3	7
+ 3	
? 2 4	
+ 1	
? 2 4	

Задача 4C Вставка ключевых значений

Имя входного файла: key.in

Имя выходного файла: key.out

Ограничение по времени: 2 секунды

Ограничение по памяти: 256 MB

Вас наняла на работу компания MacroHard, чтобы вы разработали новую структуру данных для хранения целых ключевых значений.

Эта структура выглядит как массив A бесконечной длины, ячейки которого нумеруются с единицы. Изначально все ячейки пусты. Единственная операция, которую необходимо поддерживать — это операция $Insert(L, K)$, где L — положение в массиве, а K — некоторое положительное целое ключевое значение.

Операция выполняется следующим образом:

- Если ячейка $A[L]$ пуста, то присвоить $A[L] := K$.
- Если ячейка $A[L]$ непуста, выполнить $Insert(L + 1, A[L])$, а затем присвоить $A[L] := K$.

По заданной последовательности из N целых чисел L_1, L_2, \dots, L_N вам необходимо вывести содержимое этого массива после выполнения следующей последовательности операций:

$Insert(L_1, 1)$
 $Insert(L_2, 2)$
...
 $Insert(L_N, N)$

Формат входного файла

В первой строке входного файла содержится N — число операций $Insert$ и M — максимальный номер позиции, которую можно использовать в операции $Insert$. ($1 \leq N \leq 131\,072$, $1 \leq M \leq 131\,072$).

В следующей строке даны N целых чисел L_i , которые описывают операции $Insert$ ($1 \leq L_i \leq M$).

Формат выходного файла

Выведите содержимое массива после выполнения данной последовательности операций $Insert$. На первой строке выведите W — номер последней несвободной позиции в массиве. Далее выведите W целых чисел — $A[1], A[2], \dots, A[W]$. Для пустых ячеек выводите нули.

Пример

key.in	key.out
5 4 3 3 4 1 3	6 4 0 5 2 3 1

Задача 4D Fast Refactoring

Имя входного файла: refactor.in

Имя выходного файла: refactor.out

Ограничение по времени: 2 seconds

Ограничение по памяти: 256 Mebibytes

It is known that many modern IDEs, such as Eclipse, have a wide support of refactorings — actions which can make a batch of automatic modifications by just hitting one button. Consider one kind of refactoring: a renaming. The renaming is a process of keeping the code consistent after changing a name of an identifier. After the rename refactoring is launched, IDE looks for references to the renamed identifier, and applies the same changes to them. In this problem, we will modify the source code only with rename refactorings.

We can do the following actions:

- Replace all occurrences of the word under cursor with a new word.
- Remove one word under cursor (with the space symbol after that word). Other occurrences of the same word are untouched.

- Print a code fragment that starts from a specified position and has a specified length.

A word is said to be “under cursor” if the cursor points at a symbol of the word or at the space immediately after the word.

The source code is just a single line which consists of Latin letters, digits and single spaces.

Формат входного файла

The first line of the input file contains the source string S . Its length does not exceed 10^5 . S consists of Latin letters, digits and spaces. S does not start or end with a space. The second line of the input file contains the number of queries Q ($1 \leq Q \leq 10^5$). The following Q lines contain descriptions of queries, one line per query. Each line satisfies one of the following formats:

- **REPLACE $p_i s_i$**

Denotes a replace query of the word under the p_i position to the new word s_i . The string s_i contains only Latin letters and digits. The length of s_i does not exceed 10^5 .

- **REMOVE p_i**

Denotes a remove query of the word under position p_i .

- **PRINT $b_i l_i$**

Denotes a print query of the fragment of the length l_i ($l_i > 0$) which starts on position b_i .

The symbol positions in all queries are numbered from zero. All positions and intervals do not violate current boundaries of the text, and the total length of the printed fragments does not exceed 10^5 . Also, the total size of the input file does not exceed 1 MiB.

Формат выходного файла

For each print query, you must output a line with the required fragment.

Пример

refactor.in	refactor.out
if a1 equals b then	
5	A2 eq
REPLACE 4 A2	if A2 less b
PRINT 3 5	
REPLACE 12 less	
REMOVE 13	
PRINT 0 12	

Задача 4E Permutations

Имя входного файла: permutations.in
Имя выходного файла: permutations.out
Ограничение по времени: 10 seconds
Ограничение по памяти: 256 Mebibytes

Consider a cyclic alphabet which consists of the first ten letters of the usual Latin alphabet. It is called cyclic because the next letter after ‘a’ is ‘b’, the next after ‘b’ is ‘c’ and so on. The last letter ‘j’ is followed by the first letter ‘a’.

You are given an initial string S which contains only the letters of this cyclic alphabet. You should process queries of three kinds:

- Reverse substring of S from L to R , inclusive.
- For substring of S from L to R , inclusive, replace each character with the D -th next character in the cyclic alphabetical order.
- For substring of S from L to R , inclusive, return the number of distinct permutations of the characters of this substring modulo $10^9 + 7$.

Формат входного файла

The first line of input contains one integer N ($1 \leq N \leq 10^5$) — the length of the string S , followed by the string S on the second line. The string contains only lowercase Latin letters from ‘a’ to ‘j’, inclusive. The third line contains an integer M ($1 \leq M \leq 10^5$) — the number of queries. It is followed by M lines, each of which represents one of the following three types of queries:

- -1 $L R$ ($1 \leq L \leq R \leq N$) — reverse substring from L to R .
- 0 $L R D$ ($1 \leq L \leq R \leq N$, $0 < D \leq N$) — replace each letter with the D -th next letter in the cyclic alphabetical order.
- 1 $L R$ ($1 \leq L \leq R \leq N$) — return the number of distinct permutations of the characters of substring $[L, R]$ of S .

Формат выходного файла

For each query of type “1 $L R$ ”, return the answer modulo $10^9 + 7$.

Конкурс группы А1.
МатМех, Весенний семестр 2011

Пример

permutations.in	permutations.out
6	
abcaabc	
3	
-1 1 6	
0 1 3 1	
1 1 6	180

Задача 4F Intercity Express

Имя входного файла: **intercity.in**

Имя выходного файла: **intercity.out**

Ограничение по времени: 5 seconds

Ограничение по памяти: 256 MB

Andrew is developing the a system for train ticket sales. He is going to test it on Intercity Express line that connects two large cities and has $n - 2$ intermediate stations, so there are a total of n stations numbered from 1 to n .

Intercity Express train has s seats numbered from 1 to s . In test mode the system has access to a database that contains already sold tickets in direction from station 1 to station n and needs to answer questions whether it is possible to sell a ticket from station a to station b and if so, what is the minimal number of seat that is vacant on all segments between a and b . Initially the system will have read only access, so even if there is a vacant seat, it should report so, but should not modify the data to report it reserved.

Help Andrew to test his system by writing a program that would answer such questions.

Формат входного файла

The first line of the input file contains n — the number of stations, s — the number of seats and m — the number of already sold tickets ($2 \leq n \leq 10^9$, $1 \leq s \leq 100\,000$, $0 \leq m \leq 100\,000$). The following m lines describe tickets, each ticket is described by c_i , a_i , and b_i — the seat that the owner of the ticket occupies, the station from which the ticket is sold, and the station to which the ticket is sold ($1 \leq c_i \leq s$, $1 \leq a_i < b_i \leq n$).

The following line contains q — the number of queries ($1 \leq q \leq 100\,000$). A special value p must be maintained when reading queries. Initially $p = 0$. The following $2q$ integers describe queries. Each query is described with two numbers: x_i and y_i ($x_i < y_i$). To get cities a and b between which the seat availability is requested use the following

formulae: $a = x_i + p$, $b = y_i + p$. The answer to the query is 0 if there is no seat that is vacant on each segment between a and b , or the minimal number of seat that is vacant.

After answering the query, assign the answer for the query to p .

Формат выходного файла

For each query output the answer to it.

Пример

intercity.in	intercity.out
5 3 5	1
1 2 5	2
2 1 2	2
2 4 5	3
3 2 3	0
3 3 4	2
10	0
1 2 1 2 1 2 2 3 -2 0	0
2 4 1 3 1 4 2 5 1 5	0
	0

Note that actual queries are (1,2), (2,3), (3,4), (4,5), (1,3), (2,4), (3,5), (1,4), (2,5), (1,5).

Задача 5А Две строки

Имя входного файла: **2strings.in**

Имя выходного файла: **2strings.out**

Ограничение по времени: 3 секунды

Ограничение по памяти: 64 мегабайт

Вам заданы две строки длиной не более 50 000 символов. Назовем строку хорошей, если она удовлетворяет условию, что если дописать ее в конец самой себе достаточно много раз, то в полученной строке будут содержаться в качестве подстрок обе заданные строки.

Например, для строк `ababa` и `bab` строка `ab` является хорошей — действительно, дописав ее в конец себе два раза, мы получим строку `ababab`, которая содержит обе заданные строки в качестве подстрок.

Для двух заданных строк найдите самую короткую хорошую строку.

Формат входного файла

Входной файл содержит две заданные строки. Строки состоят из символов с ASCII-кодами от 33 до 127. Длина каждой из них не превышает 50 000.

Формат выходного файла

Выведите в выходной файл ответ на задачу. Если существует несколько различных оптимальных хороших строк, то выведите любую.

Пример

2strings.in	2strings.out
ababa	
bab	ab

Задача 5В Общая подстрока

Имя входного файла: common.in
Имя выходного файла: common.out
Ограничение по времени: 0.4 секунды
Ограничение по памяти: 16 МБ

Заданы две строки, состоящие из 0 и 1. Рассмотрим все строки, которые являются подстроками обеих данных строк. Найдите среди них k -ую в лексикографическом порядке.

Строка S меньше строки T в лексикографическом порядке, если выполняется одно из двух условий:

- S является префиксом T ;
- существует i , не превышающее длину строк S и T , такое что для $j < i$ выполняется $S[j] = T[j]$ и $S[i] < T[i]$.

Формат входного файла

Первые две строки входного файла содержат заданные строки, длиной не более 4000 символов каждая. Третья строка содержит целое положительное число k , не превышающее количества общих подстрок двух заданных строк.

Формат выходного файла

Выведите в выходной файл k -ую в лексикографическом порядке общую подстроку заданных строк.

Пример

common.in	common.out
0100	
0010	01
3	

Задача 5С Конкатенация

Имя входного файла: concat.in
Имя выходного файла: concat.out
Ограничение по времени: 0.1 секунда
Ограничение по памяти: 16 мегабайт

Дана строка S , состоящая из строчных букв латинского алфавита. Рассмотрим строку $T(S)$, представляющую собой конкатенацию всех подстрок S в лексикографическом порядке.

Например, если $S = aba$, ее подстроки — это $\{a, b, a, ab, ba, aba\}$, подстроки в лексикографическом порядке — это $\{a, a, ab, aba, b, ba\}$, и таким образом, $T(S) = aaabababba$.

Найдите i -й символ строки $T(S)$.

Формат входного файла

Ввод состоит из одного или более тестов.

Каждый тест начинается строкой, содержащей натуральное число m , задающим число запросов. Следующая строка содержит строку S ($1 \leq |S| \leq 5\,000$). Следующая строка содержит m целых чисел a_i ($1 \leq a_i \leq |T(S)|$), задающих запросы.

Ввод будет завершён тестом с $m = 0$, который не требуется обрабатывать.

Сумма m по всем тестам во вводе не превысит 5 000.

Сумма длин всех строк S также не превысит 5 000.

Формат выходного файла

Для каждого теста выведите строку из m символов: ответы на запросы. Следуйте формату вывода, указанному в примере, как можно точнее.

Пример

concat.in	concat.out
10	Case #1: aaabababba
aba	Case #2: x
1 2 3 4 5 6 7 8 9 10	
1	
x	
1	
0	

Задача 5Д Ненокку

Имя входного файла: nenokku.in
Имя выходного файла: nenokku.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Очень известный автор не менее известной книги решил написать продолжение своего произведения. Он писал все свои книги на компьютере, подключенным к интернету. Из-за такой неосторожности мальчику Ненокку удалось получить доступ к еще ненаписанной книге. Каждый вечер мальчик залазил на компьютер писателя и записывал на свой компьютер новые записи. Ненокку, записав на свой компьютер очередную главу, заинтересовался, а использовал ли хоть раз писатель слово "книга". Но он не любит читать книги (он лучше полазает в интернете), и поэтому он просит вас узнать есть ли то или иное слово в тексте произведения. Но естественно его интересует не только одно слово, а достаточно много.

Формат входного файла

В каждой строчке входного файла записано одна из двух записей.

1. ? <слово> (<слово> - это набор не более 50 латинских символов);
2. A <текст> (<текст> - это набор не более 10^5 латинских символов).

1 означает просьбу проверить существование подстроки <слово> в произведении.

2 означает добавление в произведение <текст>.

Писатель только начал работать над произведением, поэтому он не мог написать более 10^5 символов. А входной файл содержит не более 15 мегабайт информации.

Формат выходного файла

Выведите на каждую строчку типа 1 "YES", если существует подстрока <слово>, и "NO" в противном случае. Не следует различать регистр букв.

Пример

nenokku.in	nenokku.out
? love	NO
? is	NO
A Loveis	YES
? love	NO
? WHO	YES
A Whoareyou	
? is	

Задача 5Е Рефрен

Имя входного файла: refrain.in
Имя выходного файла: refrain.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Рассмотрим последовательность n целых чисел от 1 до m . Подпоследовательность подряд идущих чисел называется *рефреном*, если произведение ее длины на количество вхождений в последовательность максимально.

По заданной последовательности требуется найти ее рефрен.

Формат входного файла

Первая строка входного файла содержит два целых числа: n и m ($1 \leq n \leq 150\,000$, $1 \leq m \leq 10$).

Вторая строка содержит n целых чисел от 1 до m .

Формат выходного файла

Первая строка выходного файла должна содержать произведение длины рефrena на количество ее вхождений. Вторая строка должна содержать длину рефrena. Третья строка должна содержать последовательность которая является рефреном.

Пример

refrain.in	refrain.out
9 3	9
1 2 1 2 1 3 1 2 1	3
	1 2 1

Задача 5F Башни

Имя входного файла: towers.in
Имя выходного файла: towers.out
Ограничение по времени: 4 секунды
Ограничение по памяти: 64 мегабайта

Задано число n и последовательность из n чисел. Требуется рассмотреть все возможные циклические сдвиги заданной последовательности, отсортировать их в лексикографическом порядке, и вывести сумму наибольших общих префиксов соседних в этом порядке сдвигов.

Формат входного файла

Входной файл содержит не более 200 тестовых примеров. Каждый тестовый пример состоит из двух строк. Первая из них содержит целое число $1 \leq n \leq 50000$ — количество магических башен. Вторая строка содержит n чисел в интервале от 0 до 100 — заданную последовательность.

После последнего тестового примера вместо числа n идет 0.

Формат выходного файла

Для каждого тестового примера выведите одно число — искомую сумму.

Пример

towers.in	towers.out
11	
12 8 18 18 8 18 18 8 15 15 8	
0	
	13

Задача 5G Подстроки

Имя входного файла: substr.in
Имя выходного файла: substr.out
Ограничение по времени: 0.4 секунды
Ограничение по памяти: 16 мегабайт

Дана строка s . Вам требуется подсчитать количество её различных подстрок. Пустую строку учитывать не следует.

Формат входного файла

В единственной строке входного файла содержится данная строка s , состоящая из строчных латинских букв. Длина строки не превосходит 20 000 символов.

Формат выходного файла

В единственной строке выходного файла выведите единственное число — количество различных подстрок s .

substr.in	substr.out
aaaa	4
abacaba	21

Задача 6A Просто поток

Имя входного файла: flow.in
Имя выходного файла: flow.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дана система из узлов и труб, по которым может течь вода. Для каждой трубы известна наибольшая скорость, с которой вода может протекать через нее. Известно, что вода течет по трубам таким образом, что за единицу времени в каждый узел (за исключением двух — источника и стока) втекает ровно столько воды, сколько из него вытекает. Более того, известно, что для любой пары узлов (включая источник и сток) сумма скоростей течения воды вдоль любого пути, их соединяющего, постоянна для данной пары узлов. Сумма берется таким образом, что если труба представлена в пути против направления движения воды в ней, то соответствующее слагаемое берется со знаком минус.

Ваша задача — найти наибольшее количество воды, которое за единицу времени может протекать между источником и стоком, а также скорость течения воды по каждой из труб.

Трубы являются двусторонними, то есть вода в них может течь в любом направлении. Между любой парой узлов может быть более одной трубы.

Формат входного файла

В первой строке записано натуральное число N — количество узлов в системе ($2 \leq N \leq 100$). Известно, что источник имеет номер 1, а сток номер N . Во второй строке записано натуральное M ($1 \leq M \leq 5000$) — количество труб в системе. Далее в M строках идет описание труб. Каждая труба задается тройкой целых чисел A_i, B_i, C_i , где A_i, B_i — номера узлов, которые соединяет данная труба

$(A_i \neq B_i)$, а C_i ($0 \leq C_i \leq 10000$) — наибольшая допустимая скорость течения воды через данную трубу.

Формат выходного файла

В первой строке выведите наибольшее количество воды, которое протекает между источником и стоком за единицу времени. Далее выведите M строк, в каждой из которых выведите скорость течения воды по соответствующей трубе. Если направление не совпадает с порядком узлов, заданным во входных данных, то выводите скорость со знаком минус. Числа выводите с точностью 10^{-3} .

Пример

flow.in	flow.out
2	4.0000000
2	1.0000000
1 2 1	-3.0000000
2 1 3	

Задача 6В Задача о назначениях 2

Имя входного файла: assignment2.in

Имя выходного файла: assignment2.out

Ограничение по времени: 2 секунды

Ограничение по памяти: 256 мегабайт

Дана целочисленная матрица C размера $n \times n$. Требуется выбрать n ячеек так, чтобы в каждой строке и каждом столбце была выбрана ровно одна ячейка и сумма значений в выбранных ячейках было минимальна.

Формат входного файла

Первая строка входного файла содержит n ($2 \leq n \leq 300$). Каждая из последующих n строк содержит по n чисел: C_{ij} . Все значения во входном файле неотрицательны и не превосходят 10^6 .

Формат выходного файла

Выполните сумму выбранных значений.

Примеры

assignment2.in	assignment2.out
3	3
3 2 1	
1 3 2	
2 1 3	