

Задача А. А+В

Имя входного файла: `aplusb.in`
Имя выходного файла: `aplusb.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

Для начала найдите сумму двух целых чисел.

Формат входных данных

В первой и единственной строке входного файла два числа — A и B , по модулю не превышающие 10^9 .

Формат выходных данных

В первой и единственной строке выходного файла одно число, равное сумме A и B .

Примеры

<code>aplusb.in</code>	<code>aplusb.out</code>
7 4	11

Задача В. Классический рюкзак с восстановлением

Имя входного файла: `backpack.in`
Имя выходного файла: `backpack.out`
Ограничение по времени: 0.5 секунда
Ограничение по памяти: 256 мегабайт

Дано n предметов массой m_0, \dots, m_{n-1} и стоимостью c_0, \dots, c_{n-1} соответственно.

Ими наполняют рюкзак, который выдерживает вес не более M . Какую наибольшую стоимость могут иметь предметы в рюкзаке?

Формат входных данных

В первой строке вводится натуральное число n , не превышающее 100, и натуральное число M , не превышающее 10 000.

Во второй строке вводятся n натуральных чисел m_i , не превышающих 100.

Во третьей строке вводятся n натуральных чисел c_i , не превышающих 100.

Формат выходных данных

В первой строке выходного файла выведите одно целое число: наибольшую возможную стоимость рюкзака. Во второй строке выведите количество предметов, которое необходимо взять, чтобы набрать максимальную стоимость. В третьей строке выведите через пробел номера этих предметов. Нумерация предметов начинается с нуля. Если оптимальных наборов предметов несколько, выведите любой из них. Номера предметов можно выводить в произвольном порядке.

Примеры

<code>backpack.in</code>	<code>backpack.out</code>
4 6	13
2 4 1 2	3
7 2 5 1	0 2 3

Задача С. Паросочетание

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

Двудольным графом называется неориентированный граф (V, E) , $E \subseteq V \times V$ такой, что его множество вершин V можно разбить на два множества A и B , для которых $\forall (e_1, e_2) \in E$ $e_1 \in A$, $e_2 \in B$ и $A \cup B = V$, $A \cap B = \emptyset$.

Паросочетанием в двудольном графе называется любой набор его несмежных рёбер, то есть такой набор $S \subseteq E$, что для любых двух рёбер $e_1 = (u_1, v_1)$, $e_2 = (u_2, v_2)$ из S $u_1 \neq u_2$ и $v_1 \neq v_2$.

Ваша задача — найти максимальное паросочетание в двудольном графе, то есть паросочетание с максимально возможным числом рёбер.

Формат входных данных

В первой строке записаны два целых числа n и m ($1 \leq n, m \leq 250$), где n — число вершин в множестве A , а m — число вершин в B .

Далее следуют n строк с описаниями рёбер — i -я вершина из A описана в $(i + 1)$ -й строке файла. Каждая из этих строк содержит номера вершин из B , соединённых с i -й вершиной A . Гарантируется, что в графе нет кратных ребер. Вершины в A и B нумеруются независимо (с единицы). Список завершается числом 0.

Формат выходных данных

Первая строка выходного файла должна содержать одно целое число l — количество рёбер в максимальном паросочетании. Далее следуют l строк, в каждой из которых должны быть два целых числа u_j и v_j — концы рёбер паросочетания в A и B соответственно.

Пример

<code>pairs.in</code>	<code>pairs.out</code>
2 2	2
1 2 0	1 1
2 0	2 2

Задача D. Префикс-функция

Имя входного файла: `prefix-function.in`
Имя выходного файла: `prefix-function.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Дана непустая строка S , длина которой N не превышает 10^6 . Будем считать, что элементы строки нумеруются от 1 до N .

Требуется для всех i от 1 до N вычислить её префикс-функцию $\pi[i]$.

Формат входных данных

Одна строка длины N , $0 < N \leq 10^6$, состоящая из маленьких латинских букв.

Формат выходных данных

Выведите N чисел — значения префикс-функции для каждой позиции, разделённые пробелом.

Примеры

<code>prefix-function.in</code>	<code>prefix-function.out</code>
<code>abracadabra</code>	<code>0 0 0 1 0 1 0 1 2 3 4</code>

Задача E. Range Variation Query

Имя входного файла: `rvq.in`
Имя выходного файла: `rvq.out`
Ограничение по времени: 0.5 секунда
Ограничение по памяти: 64 мегабайта

В начальный момент времени последовательность a_n задана следующей формулой: $a_n = n^2 \bmod 12345 + n^3 \bmod 23456$.

Требуется много раз отвечать на запросы следующего вида:

- найти разность между максимальным и минимальным значениями среди элементов a_i, a_{i+1}, \dots, a_j ;
- присвоить элементу a_i значение j .

Формат входных данных

Первая строка входного файла содержит натуральное число k — количество запросов ($1 \leq k \leq 100\,000$). Следующие k строк содержат запросы, по одному на строке. Запрос номер i описывается двумя целыми числами x_i, y_i .

Если $x_i > 0$, то требуется найти разность между максимальным и минимальным значениями среди элементов a_{x_i}, \dots, a_{y_i} . При этом $1 \leq x_i \leq y_i \leq 100\,000$.

Если $x_i < 0$, то требуется присвоить элементу $a_{|x_i|}$ значение y_i . В этом случае $-100\,000 \leq x_i \leq -1$ и $|y_i| \leq 100\,000$.

Формат выходных данных

Для каждого запроса первого типа в выходной файл требуется вывести одну строку, содержащую разность между максимальным и минимальным значениями на соответствующем отрезке.

Примеры

rvq.in	rvq.out
7	34
1 3	68
2 4	250
-2 -100	234
1 5	1
8 9	
-3 -101	
2 3	

Задача F. Пересечение отрезков

Имя входного файла: `segments.in`
Имя выходного файла: `segments.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Даны два отрезка: AB и CD . Определите, какое множество точек является пересечением этих отрезков.

Формат входных данных

Программа получает на вход восемь целых чисел, по абсолютной величине не превосходящих 10^4 — координаты точек A, B, C, D . Точки могут совпадать (в том числе могут совпадать и концы одного отрезка).

Формат выходных данных

Если указанные отрезки не пересекаются, то выведите строку «`Empty`». Если отрезки пересекаются в одной точке, то выведите два числа — координаты точки пересечения. Если пересечением является отрезок, то выведите четыре числа — координаты двух концов отрезка в лексикографическом порядке (то есть сначала нужно вывести ту точку, у которой меньше координата x , а если у них равны координаты x , то ту, у которой меньше координата y). Все числа следует выводить с точностью не менее 6 знаков после запятой.

Примеры

<code>segments.in</code>	<code>segments.out</code>
0 0 9 9 9 5 0 5	5.0000000000 5.0000000000
0 0 9 9 15 15 7 7	7.0000000000 7.0000000000 9.0000000000 9.0000000000
0 0 9 9 10 10 10 10	Empty