

Содержание

1	Задача А. Сумма двух [1 секунда, 256 mb]	2
2	Задача В. Два числа [1 секунда, 256 mb]	3
3	Задача С. Самая большая буква [1 секунда, 256 mb]	4
4	Задача D. Разложение числа [1 секунда, 256 mb]	5
5	Задача Е. Коллекционер [1.5 секунды, 256 mb]	6
6	Задача F. Разбиения на слагаемые [1 секунда, 256 mb]	7
7	Задача G. Компоненты связности [1 секунда, 256 mb]	8
8	Задача H. Общий предок [1 секунда, 256 mb]	9
9	Задача I. Сумма трёх [3 секунды, 256 mb]	10
10	Задача J. Неявный Ключ [1 секунда, 256 mb]	11
11	Задача K. Быстрое пересечение множеств [1 секунда, 256 mb]	12
12	Задача L. Скалярное произведение [1 секунда, 256 mb]	13
13	Задача M. Лабиринт знаний [1 секунда, 256 mb]	14
14	Задача N. Короля — в угол 3 [1 секунда, 256 mb]	15
15	Задача O. Части плоскости [1 секунда, 256 mb]	17

1 Задача А. Сумма двух [1 секунда, 256 mb]

Формат входных данных

В первой строке входного файла расположены два целых числа A и B , не превосходящих 1 000 по модулю.

Формат выходных данных

Ваша программа должна выдавать в выходной файл одно число — сумму чисел A и B .

Примеры

sum.in	sum.out
2 3	5
17 -18	-1

2 Задача В. Два числа [1 секунда, 256 mb]

Даны два целых числа A и B ($1 \leq A, B \leq 100$). Найдите два таких целых числа X и Y , что выполнено равенство $AX + BY = 1$.

Формат входных данных

Во входном файле записаны два числа A и B , разделённые пробелом.

Формат выходных данных

В выходной файл выведите два числа X и Y , разделённые пробелом. Требуется, чтобы выполнялись неравенства $|X| \leq 10\,000$, $|Y| \leq 10\,000$. Если правильных ответов несколько, разрешается вывести любой из них. Если же таких чисел не существует, выведите вместо них два нуля.

Примеры

numbers.in	numbers.out
2 3	2 -1
4 6	0 0
100 51	-5075 9951

3 Задача С. Самая большая буква [1 секунда, 256 mb]

Дана строка. Найдите в ней букву, алфавитный номер которой — самый большой среди букв этой строки.

Формат входных данных

В первой строке входного файла задана строка длины от 1 до 100 символов, включительно. Строка состоит из строчных букв латинского алфавита.

Формат выходных данных

В первой строке выходного файла выведите одну букву — ту букву строки, алфавитный номер которой максимален.

Примеры

greatest.in	greatest.out
aab	b
dabux	y

4 Задача D. Разложение числа [1 секунда, 256 mb]

Напишите программу, которая по данному натуральному числу n выводит все его простые натуральные делители с учетом кратности.

Формат входных данных

Программа получает на вход одно целое число n ($1 \leq n < 2^{31}$).

Формат выходных данных

Программа должна вывести все простые натуральные делители числа n с учетом кратности в порядке неубывания.

Примеры

prime.in	prime.out
6	2 3

5 Задача Е. Коллекционер [1.5 секунды, 256 mb]

В Байтландии за всю её историю было выпущено 15000 различных почтовых марок. Известный коллекционер почтовых марок планирует собрать полную коллекцию марок Байтландии. Какое-то количество марок (возможно, с дубликатами) у него есть на данный момент). По заданному списку марок, имеющихся в наличии, вычислить, какое минимальное количество марок коллекционер должен докупить, чтобы коллекция стала полной.

Формат входных данных

Входной файл состоит из двух строк. В первой строке задано одно целое число n ($1 \leq n \leq 3\,000\,000$) — количество имеющихся на данный момент у коллекционера экземпляров байтландских марок. Во второй строке заданы n целых чисел, каждое из которых лежит в интервале $[1, 15000]$ — номер марки, представленной соответствующим экземпляром.

Формат выходных данных

Одно число — минимальное количество марок, которое коллекционер должен докупить, чтобы коллекция стала полной.

Примеры

collection.in	collection.out
8 3 6 2 2 4 6 3 7	14995

6 Задача F. Разбиения на слагаемые [1 секунда, 256 mb]

Перечислите все разбиения целого положительного числа N ($1 \leq N \leq 40$) на целые положительные слагаемые. Разбиения должны обладать следующими свойствами:

1. Слагаемые в разбиениях идут в невозрастающем порядке.
2. Разбиения перечисляются в лексикографическом порядке.

Пример

partition.in	partition.out
4	1 1 1 1 2 1 1 2 2 3 1 4

7 Задача G. Компоненты связности [1 секунда, 256 mb]

Вам задан неориентированный граф с N вершинами и M ребрами ($1 \leq N \leq 20\,000$, $1 \leq M \leq 200\,000$). В графе отсутствуют петли и кратные ребра.

Определите компоненты связности заданного графа.

Формат входных данных

Граф задан во входном файле следующим образом: первая строка содержит числа N и M . Каждая из следующих M строк содержит описание ребра — два целых числа из диапазона от 1 до N — номера концов ребра.

Формат выходных данных

На первой строке выходного файла выведите число L — количество компонент связности заданного графа. На следующей строке выведите N чисел из диапазона от 1 до L — номера компонент связности, которым принадлежат соответствующие вершины. Компоненты связности следует занумеровать от 1 до L произвольным образом.

Пример

connect.in	connect.out
4 2	2
1 2	1 1 2 2
3 4	

8 Задача Н. Общий предок [1 секунда, 256 mb]

Дано подвешенное дерево с корнем в 1-й вершине и M запросов вида “найти у двух вершин наименьшего общего предка”.

Формат входных данных

В первой строке файла записано одно число N — количество вершин. В следующих $N - 1$ строках записаны числа. Число x на строке $2 \leq i \leq n$ означает, что x — отец вершин i . ($x < i$). На следующей строке число M . Следующие M строк содержат запросы вида (x, y) — найти наименьшего предка вершин x и y . Ограничения: $1 \leq N \leq 5 \cdot 10^4, 0 \leq M \leq 5 \cdot 10^4$.

Формат выходных данных

M ответов на запросы.

Пример

lca.in	lca.out
5	1
1	1
1	
2	
3	
2	
2 3	
4 5	

9 Задача I. Сумма трёх [3 секунды, 256 mb]

Даны три массива целых чисел A, B, C и целое число S .

Найдите такие i, j, k , что $A_i + B_j + C_k = S$.

Формат входных данных

На первой строке число S ($1 \leq S \leq 10^9$). Следующие три строки содержат описание массивов A, B, C в одинаковом формате: первое число задает длину n соответствующего массива ($1 \leq n \leq 15\,000$), затем заданы n целых чисел от 1 до 10^9 — сам массив.

Формат выходных данных

Если таких i, j, k не существует, выведите единственное число -1 . Иначе выведите на одной строке три числа i, j, k . Элементы массивов нумеруются с нуля. Если ответов несколько, выведите лексикографически минимальный.

Примеры

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

10 Задача J. Неявный Ключ [1 секунда, 256 mb]

Научитесь быстро делать две операции с массивом:

- `add i x` — добавить после i -го элемента x ($0 \leq i \leq n$)
- `del i` — удалить i -й элемент ($1 \leq i \leq n$)

Формат входных данных

На первой строке n_0 и m ($1 \leq n_0, m \leq 10^5$) — длина исходного массива и количество запросов. На второй строке n_0 целых чисел от 0 до $10^9 - 1$ — исходный массив. Далее m строк, содержащие запросы. Гарантируется, что запросы корректны: например, если просят удалить i -й элемент, он точно есть.

Формат выходных данных

Выведите конечное состояние массива. На первой строке количество элементов, на второй строке сам массив.

Примеры

<code>implicitkey.in</code>	<code>implicitkey.out</code>
<code>3 4</code>	<code>3</code>
<code>1 2 3</code>	<code>9 2 8</code>
<code>del 3</code>	
<code>add 0 9</code>	
<code>add 3 8</code>	
<code>del 2</code>	

11 Задача К. Быстрое пересечение множеств [1 секунда, 256 мб]

Даны N множеств. Множества занумерованы целыми числами от 1 до N . Для каждого множества $i = 1..N$ нужно найти такое множество $j = 1..N, j \neq i$, что их непохожесть минимальна. Непохожестью двух множеств A и B называется количество элементов, присутствующих ровно в одном из множеств A и B .

Формат входных данных

На первой строке целое число N от 2 до 10^4 — количество множеств. Далее собственно множества. Каждое множество задается следующим образом: сперва целое число k от 0 до 32 — размер множества, далее k целых чисел от 0 до 31 — элементы множества. Все элементы множества различны.

Формат выходных данных

Выведите N строк, в i -й строке выведите номер j — номер множества, которое вы считаете наименее непохожим на i -е), и собственно “непохожесть” данных множеств. Если для некоторого i существует несколько оптимальных j , выведите любое.

Пример

intersectsets.in	intersectsets.out
6	2 2
6 1 2 3 4 5 6	3 0
4 1 2 3 4	2 0
4 1 2 3 4	1 2
6 0 1 2 3 4 5	6 3
4 31 30 29 28	5 3
3 1 30 31	

12 Задача L. Скалярное произведение [1 секунда, 256 mb]

Даны два вектора: $v_1 = (x_1, x_2, \dots, x_n)$ и $v_2 = (y_1, y_2, \dots, y_n)$. Скалярным произведением этих векторов называется значение, вычисляемое по формуле: $x_1y_1 + x_2y_2 + \dots + x_ny_n$.

Разрешено переставлять координаты каждого из векторов любым образом. Выберите такие их перестановки, чтобы скалярное произведение двух полученных векторов было минимальным и выведите его значение.

$$1 \leq n \leq 800. \quad -100\,000 \leq x_i, y_i \leq 100\,000.$$

Формат входных данных

Первая строка входного файла содержит единственное целое число t — количество наборов тестовых данных. Далее следуют сами наборы, по три строки в каждом. Первая строка тестового набора содержит единственное целое число n . Две следующие строки содержат по n целых чисел, задающих координаты соответствующего вектора, каждая.

Формат выходных данных

Для каждого набора выведите строку с номером этого набора и ответом на задачу — значением минимального скалярного произведения. Следуйте формату, указанному в примере.

Пример

product.in	product.out
2	Case #1: -25
3	Case #2: 6
1 3 -5	
-2 4 1	
5	
1 2 3 4 5	
1 0 1 0 1	

13 Задача М. Лабиринт знаний [1 секунда, 256 mb]

Участникам сборов подарили билеты на аттракцион “Лабиринт знаний”. Лабиринт представляет собой N комнат, занумерованных от 1 до N , между некоторыми из которых есть двери. Когда человек проходит через дверь, показатель его знаний изменяется на определенную величину, фиксированную для данной двери. Вход в лабиринт находится в комнате 1, выход — в комнате N . Каждый участник сборов проходит лабиринт ровно один раз и набирает некоторое количество знаний (при входе в лабиринт этот показатель равен нулю). Ваша задача — показать наилучший результат.

Формат входных данных

Первая строка входного файла содержит целые числа N ($1 \leq N \leq 2000$) — количество комнат и M ($1 \leq M \leq 10000$) — количество дверей. В каждой из следующих M строк содержится описание двери — номера комнат, из которой она ведет и в которую она ведет (через дверь в лабиринте можно ходить только в одну сторону), а также целое число, которое прибавляется к количеству знаний при прохождении через дверь (это число по модулю не превышает 10000). Двери могут вести из комнаты в нее саму, между двумя комнатами может быть более одной двери.

Формат выходных данных

В выходной файл выведите “:)” — если можно пройти лабиринт и получить неограниченно большой запас знаний, “:(” — если лабиринт пройти нельзя, и максимальное количество набранных знаний в противном случае.

Пример

maze.in	maze.out
2 2	5
1 2 5	
1 2 -5	

14 Задача N. Короля — в угол 3 [1 секунда, 256 mb]

На каждой клетке шахматной доски размеров 8×8 записано целое неотрицательное число. Двое игроков по очереди переставляют короля, перемещая его только вправо, вверх или по диагонали вправо–вверх. Первоначально король стоит в левом нижнем углу. Игра продолжается до тех пор, пока король не окажется в правой верхней клетке доски. Игрок, переставивший короля в некоторую клетку получает от другого игрока денежную сумму, равную числу, записанному в этой клетке. Определите стоимость игры — сумму, которая окажется в конце игры у первого игрока, если первый игрок старается её максимизировать, а второй — минимизировать.

Формат входных данных

На вход программе подается восемь строк, каждая строка содержит восемь целых неотрицательных чисел, не превосходящих 1000. В левом нижнем углу всегда записано число 0.

Формат выходных данных

Выведите единственное число — стоимость игры.

Пример

king3.in	king3.out
0 1 0 1 0 1 0 1	4
0 0 0 0 0 0 0 0	
0 1 0 1 0 1 0 1	
0 0 0 0 0 0 0 0	
0 1 0 1 0 1 0 1	
0 0 0 0 0 0 0 0	
0 1 0 1 0 1 0 1	
0 0 0 0 0 0 0 0	

king3.in	king3.out
0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0	-3
9 9 9 9 9 9 1 9 9 9 9 9 9 1 9 2 9 9 9 9 9 9 1 9 0 9 9 9 9 9 9 9	9

15 Задача О. Части плоскости [1 секунда, 256 mb]

Даны N точек на плоскости. Проведём прямые через каждую пару точек. На сколько частей эти прямые делят плоскость?

Формат входных данных

В первой строке входного файла задано число N — количество точек ($2 \leq N \leq 10$). Следующие N строк содержат по два числа X_i Y_i каждая через пробел — координаты i -й точки ($-100 \leq X_i, Y_i \leq 100$). Никакие две данные точки не совпадают, никакие три не лежат на одной прямой. Все числа во входном файле целые.

Формат выходных данных

В первой строке выходного файла выведите P — количество частей, на которые полученные прямые делят плоскость.

Примеры

parts.in	parts.out
4 0 0 0 1 1 0 1 1	16
3 1 5 2 3 -8 4	7