

Задача А. Фибоначчиева последовательность

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

Последовательность чисел $a_1, a_2, \dots, a_i, \dots$ называется Фибоначчиевой, если для всех $i \geq 3$ верно, что $a_i = a_{i-1} + a_{i-2}$, то есть каждый член последовательности (начиная с третьего) равен сумме двух предыдущих.

Ясно, что задавая различные числа a_1 и a_2 мы можем получать различные такие последовательности, и любая Фибоначчиева последовательность однозначно задается двумя своими первыми членами.

Будем решать обратную задачу. Вам будет дано число N и два члена последовательности: a_N и a_{N+1} . Вам нужно написать программу, которая по их значениям найдет a_1 и a_2 .

Формат входных данных

Вводятся число N и значения двух членов последовательности: a_N и a_{N+1} ($1 \leq N \leq 30$, члены последовательности — целые числа, по модулю не превышающие 100)

Формат выходных данных

Выведите два числа — значения первого и второго членов этой последовательности.

Примеры

<code>fib.in</code>	<code>fib.out</code>
4 3 5	1 1

Задача В. Параграф

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

Дана строка, являющаяся параграфом в тексте. Текст необходимо отформатировать так, чтобы длина каждой строки не превосходила числа M , слова при этом не разрывать.

Формат входных данных

На вход программе сначала подается число $0 < M \leq 255$. В следующей строке находится исходный текст. Длина слов в нем не превышает M , слова разделены ровно одним пробелом.

Формат выходных данных

Выведите разбиение этого текста на строки длиной не более чем M символов (слово переносится на следующую строку, только если в текущей строке его разместить уже невозможно). Новая строка не должна начинаться с пробела.

Примеры

<code>par.in</code>	<code>par.out</code>
7 один два три четыре	один два три четыре

Задача С. Шарики

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

В одной компьютерной игре игрок выставляет в линию шарiki разных цветов. Когда образуется непрерывная цепочка из трех и более шариков одного цвета, она удаляется из линии. Все шарики при этом сдвигаются друг к другу, и ситуация может повториться.

Напишите программу, которая по данной ситуации определяет, сколько шариков будет сейчас уничтожено. Естественно, непрерывных цепочек из трех и более одноцветных шаров в начальный момент может быть не более одной.

Формат входных данных

Даны количество шариков в цепочке (не более 1000) и цвета шариков (от 0 до 9, каждому цвету соответствует свое целое число).

Формат выходных данных

Требуется вывести количество шариков, которое будет уничтожено.

Примеры

<code>balls.in</code>	<code>balls.out</code>
5 1 3 3 3 2	3
10 3 3 2 1 1 1 2 2 3 3	10

Задача D. Парк аттракционов

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

В городе π недавно построили парк аттракционов, в котором есть павильон игровых автоматов. Каждый из автоматов рассчитан на одного человека. В программе олимпиады планируется посещение этого павильона.

Перед организаторами встала сложная задача — составить расписание игры участников олимпиады на автоматах таким образом, чтобы каждый из N участников олимпиады смог поиграть на каждом из автоматов, и при этом автобус, увозящий участников из парка олимпиады, смог бы отправиться к месту проживания как можно раньше.

Время перемещения участников между автоматами, а также между автобусом и павильоном считается равным нулю. Каждый из участников в любой момент времени может как играть на автомате, так и ждать своей очереди, например, гуляя по парку. Для каждого из M ($M \leq N$) автоматов известно время игры на нём t_i ($1 \leq i \leq M$). Прервать начатую игру на автомате невозможно. Автобус привозит всех участников олимпиады в парк одновременно в нулевой момент времени.

Требуется написать программу, которая по заданным числам N , M и t_i определяет оптимальное расписание игры на автоматах для каждого из участников.

Формат входных данных

В первой строке входного файла содержатся два числа: N и M ($1 \leq M \leq N \leq 100$). Во второй строке заданы M целых чисел t_i ($1 \leq t_i \leq 100$), каждое из которых задаёт время игры на i -м автомате ($1 \leq i \leq M$). Числа в строке разделяются одиночными пробелами.

Формат выходных данных

В первой строке необходимо вывести одно число — минимально возможное время отправления автобуса из парка аттракционов. Далее необходимо вывести N расписаний игр на автоматах, по одному для каждого из участников. Каждое расписание описывается в $(M + 1)$ строках, первая из которых — пустая, а далее следуют M строк, описывающих автоматы в порядке их посещения этим участником. Посещение автомата описывается двумя целыми числами: номером автомата j ($1 \leq j \leq M$) и временем начала игры участника на этом автомате.

Примеры

<code>attract.in</code>	<code>attract.out</code>
2 1 2	4 1 0 1 2
3 2 2 1	6 1 0 2 2 1 2 2 4 2 0 1 4

Задача Е. Хитроумное устройство

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

Петя нашел на чердаке старый телеграфный аппарат и приделал к нему хитроумное устройство, которое может печатать на телеграфной ленте определенное слово (обозначим его X). Петино устройство может напечатать на ленте это слово сколько угодно раз. Петя может заставить аппарат напечатать на ленте и любое другое сообщение, но для этого ему нужно разобрать свое хитроумное устройство, и после этого он уже не сможет печатать сообщение X . А самое главное, что напечатать даже один символ другого сообщения потребует от Пети больше усилий, чем напечатать на ленте слово X с помощью хитроумного устройства.

Петя хочет сделать так, чтобы всем казалось, что ему по телеграфу пришло сообщение Z . Для этого он может (строго в этой последовательности):

- сколько угодно раз напечатать сообщение X
- разобрать хитроумное устройство и посимвольно напечатать еще что-нибудь (назовем это Y)
- оторвать и выбросить начало ленты так, чтобы на оставшейся ленте было напечатано в точности сообщение Z

Поскольку набирать отдельные символы сообщения Y довольно сложно, Петя хочет, чтобы в сообщении Y было как можно меньше символов.

Формат входных данных

В первой строке вводится слово X , которое Петя может печатать с помощью хитроумного устройства сколько угодно раз. Во второй строке вводится сообщение Z , которое хочет получить Петя. Каждое сообщение состоит только из маленьких латинских букв и имеет длину не более 100 символов.

Формат выходных данных

Выведите минимальное по длине сообщение Y , которое Пете придется допечатать вручную.

Примеры

<code>device.in</code>	<code>device.out</code>
mama amamam	m
ura mura	mura
computer comp	comp
ejudge judge	
m mmm	

Задача F. Футбольный турнир

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

В лагере прошел турнир по футболу. Теперь Руслану Назировичу необходимо подвести результаты и составить итоговую таблицу. Турнир прошел в один круг, т.е. каждая команда сыграла с каждой ровно один раз. За победу команда получала 3 очка, за ничью 1 очко, а за поражение 0 очков. Кроме того в таблице учитывается разница пропущенных и забитых мячей.

Требуется упорядочить команды по невозрастанию набранных очков, а при равенстве очков, раньше в таблице должна идти команда, у которой разница забитых и пропущенных мячей больше.

Формат входных данных

В первой строке входного файла вводится единственное число N ($1 \leq N \leq 1000$) — количество команд, участвовавших в турнире.

Во второй строке записаны N неотрицательных целых чисел — количество очков, набранных каждой командой.

В третьей строке содержатся N чисел — разница забитых и пропущенных мячей каждой команды.

Формат выходных данных

В выходной файл выведите N чисел — номера команд в порядке, в котором они будут записаны в итоговой таблице. Если количество очков и разница забитых/пропущенных мячей для каких-то команд совпадают, то их можно выводить в произвольном порядке.

Пример

<code>football.in</code>	<code>football.out</code>
5	2 3 5 1 4
5 12 5 0 5	
-2 8 2 -6 2	