

Задача А. Автостопом по галактике

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

Путешественник хочет посетить все планеты галактики, при этом он хочет завершить свое путешествие как можно раньше. Он знает, что количество лет, которое он потратит на перелёт от одной планеты до другой, равно отношению массы конечной планеты к массе начальной. При этом он может начать и закончить свое путешествие на любой планете и совершать перелёт между любыми двумя планетами.

Формат входного файла

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 500$) — количество планет в галактике. В i -й из следующих n строк записано целое число m_i — масса i -й планеты ($1 \leq m_i \leq 3000$).

Формат выходного файла

Выведите минимальное количество лет, которое понадобится путешественнику, не менее чем с тремя знаками после десятичной точки.

Пример

galaxy.in	galaxy.out
3	1.166667
3 1 2	

Примечание. Минимальное время равно $2/3 + 1/2 \approx 1,66667$.

Задача В. Такси

Имя входного файла: taxi.in
Имя выходного файла: taxi.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайт

После затянувшегося совещания директор фирмы решил заказать такси, чтобы развезти сотрудников по домам. Он заказал N машин — ровно столько, сколько у него сотрудников. Но когда они подъехали, оказалось, что у каждого водителя такси свой тариф за 1 километр.

Каждый сотрудник сказал директору, сколько километров ему нужно проехать до дома. Разные сотрудники должны сесть в разные такси. Теперь директор хочет определить, какой из сотрудников на каком такси должен поехать домой, чтобы суммарные затраты на такси (а их несет фирма) были минимальны.

Формат входного файла

Сначала во входном файле записано натуральное число N ($1 \leq N \leq 1000$) — количество сотрудников компании (совпадающее с количеством вызванных машин такси). Далее записано N чисел, задающих расстояния в километрах от работы до домов сотрудников компании (первое число — для первого сотрудника, второе — для второго и т.д.). Все расстояния — положительные целые числа, не превышающие 1000. Далее записано еще N чисел — тарифы за проезд одного километра в такси (первое число — в первой машине такси, второе — во второй и т.д.). Тарифы выражаются положительными целыми числами, не превышающими 10000.

Формат выходного файла

В выходной файл выведите N чисел — оптимальное распределение сотрудников по такси. Первым выведите номер такси, в которое должен сесть первый сотрудник, вторым — номер такси, в которое должен сесть второй и т.д. Если есть несколько вариантов рассадки сотрудников, при которых затраты минимальны, выведите любой из них.

Пример

taxi.in	taxi.out
3 10 20 30 50 20 30	1 3 2
5 10 20 1 30 30 3 3 3 2 3	1 2 3 5 4

Задача С. Тестирующая система

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

Юный программист Саша написал свою первую тестирующую систему. Он так обрадовался тому, что она скомпилировалась, что решил пригласить школьных друзей на свой собственный контекст.

Но в конце тура выяснилось, что система не умеет сортировать команды в таблице результатов. Помогите Саше реализовать эту сортировку.

Команды упорядочиваются по правилам ACM:

- по количеству решённых задач в порядке убывания;
- при равенстве количества решённых задач — по штрафному времени в порядке возрастания;
- при прочих равных — по номеру команды в порядке возрастания.

Формат входного файла

Первая строка содержит натуральное число n ($1 \leq n \leq 100\,000$) — количество команд, участвующих в контексте. В i -й из следующих n строк записано количество решённых задач S ($1 \leq S \leq 100$) и штрафное время T ($1 \leq T \leq 100\,000$) команды с номером i .

Формат выходного файла

В выходной файл выведите n чисел — номера команд в отсортированном порядке.

Пример

ejudge.in	ejudge.out
5 3 50 5 720 1 7 0 0 8 500	5 2 1 3 4

Задача D. Число

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

Вася написал на длинной полоске бумаги большое число и решил похвастаться своему старшему брату Пете этим достижением. Но только он вышел из комнаты, чтобы позвать брата, как его сестра Катя вбежала в комнату и разрезала полоску бумаги на несколько частей. В результате на каждой части оказалось одна или несколько идущих подряд цифр.

Теперь Вася не может вспомнить, какое именно число он написал. Только помнит, что оно было очень большое. Чтобы утешить младшего брата, Петя решил выяснить, какое максимальное число могло быть написано на полоске бумаги перед разрезанием. Помогите ему!

Формат входного файла

Входной файл содержит одну или более строк, каждая из которых содержит последовательность цифр. Количество строк во входном файле не превышает 100, каждая строка содержит от 1 до 100 цифр. Гарантируется, что хотя бы в одной строке первая цифра отлична от нуля.

Формат выходного файла

Выведите в выходной файл одну строку — максимальное число, которое могло быть написано на полоске перед разрезанием.

Примеры

number.in	number.out
2	66220004
20	
004	
66	
3	3

Задача Е. Лексикографический порядок

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

Будем считать, что одно натуральное число лексикографически меньше другого, если таковы их записи в десятичной системе счисления. Вам необходимо найти k -е по порядку число в лексикографически отсортированном множестве натуральных чисел от 1 до n включительно.

Формат входного файла

Первая строка входного файла содержит два числа n и k ($1 \leq k \leq n \leq 10^6$).

Формат выходного файла

Выведите единственное число — k -й в лексикографическом порядке элемент множества натуральных чисел от 1 до n .

Пример

lexsort.in	lexsort.out
10 2	10