

Содержание

Must have	2
Задача 8А. Декодирование [0.1 sec, 256 mb]	2
Задача 8В. Кодирование [0.1 sec, 256 mb]	3
Задача 8С. Золотой песок [0.1 sec, 256 mb]	5
Обязательные задачи	6
Задача 8D. Число [0.1 sec, 256 mb]	6
Задача 8Е. Различные слагаемые [0.1 sec, 256 mb]	7
Задача 8F. Белоснежка и n гномов [0.2 sec, 256 mb]	8
Задача 8G. Коробки [0.2 sec, 256 mb]	9
Дополнительные задачи	10
Задача 8H. Коробки [0.2 sec, 256 mb]	10
Задача 8I. Эльфы и олени [0.2 sec, 256 mb]	11
Задача 8J. Автомойка [0.3 sec, 256 mb]	12

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же `stdin`), вывести ответ нужно в **стандартный поток вывода** (он же `stdout`).

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) **переопределение стандартного аллокатора** ускорит вашу программу.

Обратите внимание на GNU C++ компиляторы с суффиксом `inc`, они позволяют пользоваться **дополнительной библиотекой**. Под ними можно сдать **вот это**.

Must have

Задача 8А. Декодирование [0.1 sec, 256 mb]

В этой задаче по набору беспрефиксных кодов букв и строке, закодированной с помощью этих кодов так, как это описано в задаче “Кодирование”, нужно восстановить исходную строку.

Формат входных данных

В первой строке входного файла заданы два целых числа k и l через пробел — количество различных букв, встречающихся в строке, и размер получившейся закодированной строки, соответственно. В следующих k строках записаны коды букв в формате “<letter>: <code>”. Ни один код не является префиксом другого. Буквы могут быть перечислены в любом порядке. В качестве букв могут встречаться лишь строчные буквы латинского алфавита; каждая из этих букв встречается в строке хотя бы один раз. Наконец, в последней строке записана закодированная строка.

Исходная строка и коды всех букв непусты. Заданный код таков, что закодированная строка имеет минимальный возможный размер.

Формат выходных данных

В первой строке выходного файла выведите строку s . Она должна состоять из строчных букв латинского алфавита. Гарантируется, что длина правильного ответа не превосходит 100 000 символов.

Примеры

stdin	stdout
1 1 a: 0 0	a
4 14 a: 0 b: 10 c: 110 d: 111 01001100100111	abacabad

Замечание

Кстати, есть такая штука: `scanf("%c: %s")`.

Задача 8В. Кодирование [0.1 сек, 256 mb]

Известно, что в текстах практически на любом языке, имеющем алфавит, одни буквы систематически встречаются чаще, чем другие. Например, в типичном тексте на английском языке частота буквы ‘e’ больше двенадцати процентов, а частота буквы ‘z’ меньше одного процента. В русском языке буква ‘o’ имеет частоту более девяти процентов, а буква ‘ъ’ — менее одной десятой процента.

Такие различия между буквами можно использовать, чтобы хранить тексты в сжатом виде. Сопоставим каждой букве какой-то непустой двоичный код; частым буквам сопоставим коды из малого количества битов, а редким — из большого количества битов. Строку будем записывать как последовательность кодов букв в ней без разделителей. Коды букв организуем таким образом, чтобы ни один код буквы не начинался с кода какой-то другой буквы; такие коды называются *беспрефиксными*. К примеру, если мы закодируем букву ‘e’ кодом 10, то все остальные буквы должны иметь либо код 0, либо код, начинающийся на 00, 01 или 11. Это нужно для того, чтобы в закодированном тексте не требовалось ставить разделители между буквами: при такой организации кодов любая закодированная строка восстанавливается однозначно.

От того, какие коды букв мы выберем, зависит размер закодированной строки. Например, если в строке “abacabad” мы закодируем буквы как “a: 00”, “b: 01”, “c: 10” и “d: 11”, то закодированная строка будет иметь вид “00 01 00 10 00 01 00 11” (группы цифр, соответствующие разным буквам, разделены для удобства восприятия), и размер её будет равен 16. Если же мы выберем другие беспрефиксные коды “a: 0”, “b: 10”, “c: 110” и “d: 111”, то закодированная строка будет иметь вид “0 10 0 110 0 10 0 111”, и её размер будет равен 14.

Дана строка s . Сопоставьте каждой её букве двоичный код так, чтобы коды букв были беспрефиксными, и при этом закодированная ими строка s имела минимальный возможный размер.

Формат входных данных

В первой строке входного файла задана непустая строка s . Она состоит из строчных букв латинского алфавита. Длина этой строки не превосходит 100 000 символов.

Формат выходных данных

В первой строке выходного файла выведите два числа k и l через пробел — количество различных букв, встречающихся в строке, и размер получившейся закодированной строки, соответственно. В следующих k строках запишите коды букв в формате “<letter>: <code>”. Ни один код не должен являться префиксом другого. Буквы могут быть перечислены в любом порядке. Наконец, в последней строке запишите закодированную строку.

Если ответов с минимальным значением l несколько, можно вывести любой из них.

Примеры

stdin	stdout
a	1 1 a: 0 0
abacabad	4 14 a: 0 b: 10 c: 110 d: 111 01001100100111

Замечание

Коды удобно восстановить, обойдя дерево кодов.

Задача 8С. Золотой песок [0.1 sec, 256 mb]

Во время ограбления магазина вор обнаружил N ящичков с золотым песком. В ящичек под номером i песок имеет стоимость v_i и вес w_i . Чтобы унести награбленное, вор использует рюкзак. Требуется определить наибольшую суммарную стоимость песка, который может унести грабитель, если грузоподъемность рюкзака ограничена величиной W .

Из ящичков можно пересыпать любое количество песка, тогда отношение стоимости отсыпанного песка к стоимости всего ящичка будет равна отношению объема пересыпанного песка к объему всего ящичка.

Формат входных данных

В первой строке входного файла записаны два числа — N и W ($1 \leq N \leq 1000$, $0 \leq W \leq 2 \cdot 10^6$). Далее следует N строк по два целых числа в каждой. В i -ой строке записана стоимость v_i и вес w_i песка в i -ом ящичке. Все числа неотрицательные и не превосходят $2 \cdot 10^6$.

Формат выходных данных

Выведите искомую максимальную стоимость с точностью до 3 знаков после запятой.

Пример

stdin	stdout
3 50 60 20 100 50 120 30	180.000

Замечание

Компаратор работает. Стандартная теория работает.

Обязательные задачи

Задача 8D. Число [0.1 sec, 256 mb]

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 разрежала 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 входных данных

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 выходных данных

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

Примеры

stdin	stdout
2 20 004 66	66220004
3	3

Замечание

Компаратор работает. Стандартная теория работает.

Задача 8Е. Различные слагаемые [0.1 сек, 256 mb]

В этой задаче требуется найти разбиение натурального числа n на как можно большее количество различных натуральных слагаемых.

Формат входных данных

В первой строке входного файла записано натуральное число n ($1 \leq n \leq 1\,000\,000\,000$).

Формат выходных данных

В первой строке выходного файла выведите k — количество слагаемых. Во второй строке выведите k чисел через пробел — сами слагаемые. Если разбиений с максимальным количеством слагаемых несколько, можно вывести любое из них. Слагаемые можно выводить в любом порядке.

Примеры

stdin	stdout
4	2 1 3
6	3 1 2 3

Задача 8F. Белоснежка и n гномов [0.2 sec, 256 mb]

«Ну не гномы, а наказание какое-то!», — подумала Белоснежка, в очередной раз пытаюсь уложить гномов спать. Одного уложишь — другой уже проснулся! И так всю ночь.

У Белоснежки n гномов, и все они очень разные. Она знает, что для того, чтобы уложить спать i -го гнома нужно a_i минут, и после этого он будет спать ровно b_i минут. Помогите Белоснежке узнать, может ли она получить хотя бы минутку отдыха, когда все гномы будут спать, и если да, то в каком порядке для этого нужно укладывать гномов спать.

Например, пусть есть всего два гнома, $a_1 = 1, b_1 = 10, a_2 = 10, b_2 = 20$. Если Белоснежка сначала начнет укладывать первого гнома, то потом ей потребуется целых 10 минут, чтобы уложить второго, а за это время проснется первый. Если же она начнет со второго гнома, то затем она успеет уложить первого и получит целых 9 минут отдыха.

Формат входных данных

Первая строка входного файла содержит число n ($1 \leq n \leq 10^5$), вторая строка содержит числа a_1, a_2, \dots, a_n , третья — числа b_1, b_2, \dots, b_n ($1 \leq a_i, b_i \leq 10^9$).

Формат выходных данных

Выведите в выходной файл n чисел — порядок, в котором нужно укладывать гномов спать. Если Белоснежке отдохнуть не удастся, выведите число -1 . Если решений несколько, выведите любое.

Пример

stdin	stdout
2 1 10 10 20	2 1
2 10 10 10 10	-1
3 1 4 1 5 3 4	2 1 3

Замечание

Компаратор работает. Стандартная теория работает.

Задача 8G. Коробки [0.2 sec, 256 mb]

У Васи в комнате очень много коробок, которые валяются в разных местах. Васина мама хочет, чтобы он прибрался. Свободного места в комнате мало и поэтому Вася решил собрать все коробки и поставить их одну на другую.

К сожалению, это может быть невозможно. Например, если на картонную коробку с елочными украшениями положить что-то железное и тяжелое, то вероятно следующий Новый год придется встречать с новыми игрушками.

Вася взвесил каждую коробку и оценил максимальный вес который она может выдержать. Помогите ему определить какое наибольшее количество коробок m он сможет поставить одну на другую так, чтобы для каждой коробки было верно, что суммарный вес коробок сверху не превышает максимальный вес, который она может выдержать.

Формат входных данных

Первая строка входного файла содержит целое число n ($1 \leq n \leq 1250$) — количество коробок в комнате. Каждая следующая из n строк содержит два целых числа w_i и c_i ($1 \leq w_i \leq 10^5, 1 \leq c_i \leq 10^9$), где w_i — это вес коробки с номером i , а c_i — это вес который она может выдержать.

Формат выходных данных

В выходной файл выведите одно число — ответ на задачу.

Пример

stdin	stdout
3 10 11 20 100 30 10	3
3 11 11 20 100 30 10	2

Замечание

Задача с практики.

Дополнительные задачи

Задача 8Н. Коробки [0.2 sec, 256 mb]

У Васи в комнате очень много коробок, которые валяются в разных местах. Васина мама хочет, чтобы он прибрался. Свободного места в комнате мало и поэтому Вася решил собрать все коробки и поставить их одну на другую.

К сожалению, это может быть невозможно. Например, если на картонную коробку с елочными украшениями положить что-то железное и тяжелое, то вероятно следующий Новый год придется встречать с новыми игрушками.

Вася взвесил каждую коробку и оценил максимальный вес который она может выдержать. Помогите ему определить какое наибольшее количество коробок m он сможет поставить одну на другую так, чтобы для каждой коробки было верно, что суммарный вес коробок сверху не превышает максимальный вес, который она может выдержать.

Формат входных данных

Первая строка входного файла содержит целое число n ($1 \leq n \leq 10^5$) — количество коробок в комнате. Каждая следующая из n строк содержит два целых числа w_i и c_i ($1 \leq w_i \leq 10^5$, $1 \leq c_i \leq 10^9$), где w_i — это вес коробки с номером i , а c_i — это вес которой она может выдержать.

Формат выходных данных

В выходной файл выведите одно число — ответ на задачу.

Пример

stdin	stdout
3 10 11 20 100 30 10	3
3 11 11 20 100 30 10	2

Замечание

Задача с практики.

Задача 81. Эльфы и олени [0.2 sec, 256 mb]

Скоро новый год и Санта-Клаус уже начал готовить свою волшебную оленью упряжку, на которой он развозит подарки детям. Известно, что упряжку везут несколько волшебных оленей, на каждом из которых едут два эльфа.

Но волшебные олени — строптивые животные, поэтому не любые два эльфа могут ехать на любом олене. А именно, каждый олень характеризуется некоторой строптивостью a_i , а каждый эльф — темпераментом b_i . Два эльфа j и k могут ехать на i -м олене в том и только в том случае, если либо $b_j < a_i < b_k$, либо $b_k < a_i < b_j$.

Чтобы его появление было максимально зрелищным, Санта-Клаус хочет, чтобы в его упряжке было как можно больше оленей. Про каждого оленя Санта знает его строптивость, а про каждого эльфа — его темперамент.

Помогите Санте выяснить, какое максимальное количество оленей он сможет включить в упряжку, каких оленей ему следует выбрать, и какие эльфы должны на них ехать.

Формат входных данных

Первая строка входного файла содержит два целых числа m и n — количество оленей и эльфов, соответственно ($1 \leq m, n \leq 100\,000$).

Вторая строка содержит m целых чисел a_i — строптивость оленей ($0 \leq a_i \leq 10^9$). Третья строка содержит n целых чисел b_i — темперамент эльфов ($0 \leq b_i \leq 10^9$).

Формат выходных данных

На первой строке выходного файла выведите одно число k — максимальное количество оленей, которое Санта-Клаус может включить в свою упряжку. На следующих k строках выведите по три целых числа: $d_i, e_{i,1}, e_{i,2}$ — для каждого оленя в упряжке выведите его номер и номера эльфов, которые на нем поедут. Если решений несколько, выведите любое. Выводить эльфов нужно в таком порядке, что $b_{e_{i,1}} < a_{d_i} < b_{e_{i,2}}$.

И эльфы, и олени пронумерованы, начиная с единицы, в том порядке, в котором они заданы во входном файле.

Пример

stdin	stdout
4 6	2
2 3 4 5	1 1 2
1 3 2 2 5 2	3 4 5

Замечание

Привет со ВКОШпа.

Задача 8J. Автомойка [0.3 sec, 256 mb]

Бен – владелец автомойки. Бен предлагает мойку и чистку салона машин. Его автомойка расположена в столице Ранглии, поэтому он часто обслуживает клиентов из столицы. Недавно он получил важный и весьма большой заказ на мойку и чистку n машин. Этот заказ должен быть обработан как можно быстрее!

Предварительное исследование показало, что i -ю машину нужно мыть a_i минут, чистить b_i минут. Мойка и чистка происходят в разных зданиях, поэтому нельзя одну машину одновременно и мыть, и чистить. Порядок этих двух операций для каждой из машин не важен. Также не важен порядок мойки машин и порядок чистки машин. У Бена есть два рабочих – Том и Джери. Том моет, Джери чистит. Требования качества обязывают каждого из рабочих закончить с одной машиной перед тем, как браться за другую.

Помогите Бену минимизировать время обработки заказа и составить расписание работ.

Формат входных данных

На первой строке число машин n ($1 \leq n \leq 10\,000$).

Следующие n строк содержат пары целых чисел a_i, b_i ($1 \leq a_i, b_i \leq 10^5$).

Формат выходных данных

На первой строке выведите время выполнения заказа t .

Следующие n строк должны содержать по два числа – когда Тому начать мыть машину и когда Джери начать чистить её салон.

Если есть несколько оптимальных расписаний, выведите любое.

Пример

stdin	stdout
6	39
10 6	11 26
7 9	4 17
3 8	0 7
1 2	3 15
12 7	27 0
6 6	21 32

Замечание

Гробик. Или нет?