

## Содержание

<b>Обязательные задачи</b>	<b>3</b>
Задача 14А. Декодирование [0.5 sec, 256 mb]	3
Задача 14В. Кодирование [0.5 sec, 256 mb]	4
Задача 14С. Различные слагаемые [0.5 sec, 256 mb]	5
Задача 14D. Отрезки [0.5 sec, 256 mb]	6
Задача 14Е. Число [0.5 sec, 256 mb]	7
Задача 14F. Золотой песок [0.5 sec, 256 mb]	8
Задача 14G. Скалярное произведение [0.5 sec, 256 mb]	9
Задача 14Н. Белоснежка и $n$ гномов [0.5 sec, 256 mb]	10
<b>Дополнительные задачи</b>	<b>11</b>
Задача 14I. $F2 \parallel C_{max}$ [0.5 sec, 256 mb]	11
Задача 14J. Коробки [0.5 sec, 256 mb]	12
Задача 14K. Коробки [0.5 sec, 256 mb]	13
Задача 14L. Эльфы и олени [0.5 sec, 256 mb]	14
Задача 14M. Car Wash [0.5 sec, 256 mb]	15

---

### **Пример работы с файлами.**

Если вы не умеете читать/выводить данные, или открывать файлы, воспользуйтесь примерами. <http://acm.math.spbu.ru/~sk1/algo/sum/>

### **Пример работы с файлами.**

В некоторых задачах большой ввод и вывод. Про ввод-вывод в C++:

[http://acm.math.spbu.ru/~sk1/algo/input-output/cpp\\_common.html](http://acm.math.spbu.ru/~sk1/algo/input-output/cpp_common.html)

Имеет смысл пользоваться супер быстрым вводом-выводом. Две версии:

[http://acm.math.spbu.ru/~sk1/algo/input-output/io\\_export.cpp.html](http://acm.math.spbu.ru/~sk1/algo/input-output/io_export.cpp.html)

[http://acm.math.spbu.ru/~sk1/algo/input-output/fread\\_write\\_export.cpp.html](http://acm.math.spbu.ru/~sk1/algo/input-output/fread_write_export.cpp.html)

### **Выделение памяти.**

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

<http://acm.math.spbu.ru/~sk1/algo/memory.cpp.html>

## Обязательные задачи

### Задача 14А. Декодирование [0.5 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

decoding.in	decoding.out
1 1 a: 0 0	a
4 14 a: 0 b: 10 c: 110 d: 111 01001100100111	abacabad

### Задача 14В. Кодирование [0.5 sec, 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

encoding.in	encoding.out
a	1 1 a: 0 0
abacabad	4 14 a: 0 b: 10 c: 110 d: 111 01001100100111

### Задача 14С. Различные слагаемые [0.5 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

sumdiff.in	sumdiff.out
4	2 1 3
6	3 1 2 3

### Задача 14D. Отрезки [0.5 sec, 256 mb]

Даны отрезки на прямой. Какое максимальное количество отрезков можно выбрать так, чтобы никакие два из них не пересекались? Отрезки считаются открытыми.

#### Формат входных данных

В первой строке входного файла задано целое число  $n$  ( $1 \leq n \leq 100\,000$ ). В следующих  $n$  строках описаны отрезки;  $i$ -я из этих строк содержит два целых числа  $l_i$  и  $r_i$  через пробел — координаты начала и конца отрезка ( $1 \leq l_i < r_i \leq 10^9$ ).

#### Формат выходных данных

В первой строке выходного файла выведите одно число — максимальное количество выбранных отрезков.

#### Примеры

segments.in	segments.out
2 1 2 3 5	2
2 1 3 2 6	1
3 2 4 3 5 1 3	2

**Задача 14Е. Число [0.5 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**

<code>number.in</code>	<code>number.out</code>
2 20 004 66	66220004
3	3

### Задача 14F. Золотой песок [0.5 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

dust.in	dust.out
3 50	180.000
60 20	
100 50	
120 30	



### Задача 14G. Скалярное произведение [0.5 сек, 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.

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

#### 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	

### Задача 14Н. Белоснежка и $n$ гномов [0.5 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

dwarfs.in	dwarfs.out
2 1 10 10 20	2 1
2 10 10 10 10	-1
3 1 4 1 5 3 4	2 1 3

## Дополнительные задачи

### Задача 14I. $F2 \parallel C_{max}$ [0.5 sec, 256 mb]

Имеется множество из  $n$  работ и два станка. Время выполнения  $i$ -й работы на первом станке равно  $a_i$ , время выполнения  $i$ -й работы на втором станке равно  $b_i$ . Каждую работу надо выполнить сначала на первом станке, потом на втором. И на первом, и на втором станке работы можно выполнять в произвольном порядке. Каждый станок в каждый момент времени может выполнять только одну работу.

Минимизируйте  $C_{max}$  — время выполнения последней работы на втором станке.

#### Формат входных данных

В первой строке дано одно целое число  $n$  ( $1 \leq n \leq 100\,000$ ) — количество работ. В следующей строке  $n$  целых чисел от 0 до  $10^6$  — время выполнения работ на первом станке. В следующей строке  $n$  целых чисел от 0 до  $10^6$  — время выполнения работ на втором станке.

#### Формат выходных данных

В первой строке выведите единственное число —  $C_{max}$ . Во второй строке выведите перестановку — порядок выполнения работ на первом станке. В третьей строке выведите перестановку — порядок выполнения работ на втором станке.

#### Примеры

f2cmax.in	f2cmax.out
3	16
1 2 3	1 3 2
5 5 5	1 2 3
2	6
3 2	2 1
1 3	2 1

### Задача 14J. Коробки [0.5 сек, 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

boxes.in	boxes.out
3 10 11 20 100 30 10	3
3 11 11 20 100 30 10	2

### Задача 14К. Коробки [0.5 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 который она может выдержать.

#### Формат выходных данных

В выходной файл выведите одно число — ответ на задачу.

#### Пример

boxes.in	boxes.out
3 10 11 20 100 30 10	3
3 11 11 20 100 30 10	2

### Задача 14L. Эльфы и олени [0.5 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

elves.in	elves.out
4 6	2
2 3 4 5	1 1 2
1 3 2 2 5 2	3 4 5

### Задача 14М. Car Wash [0.5 sec, 256 mb]

Ben is the owner of a car wash. Ben offers car washing and dry-cleaning of the car compartment. His wash is located in the capital of Edgeland, and he often serves clients from government. Recently Ben has received an order of washing and dry-cleaning of  $n$  cars. And this order must be executed as fast as possible!

After preliminary investigation, Ben found out that the  $i$ -th car can be washed in  $a_i$  minutes and cleaned in  $b_i$  minutes. Washing and cleaning are performed at different buildings, so they cannot be performed simultaneously for the same car. The order of the two operations for a car is irrelevant, neither is the order of processing the cars. Ben's best workers Tom and Jerry were called to perform the order. Tom will wash the cars, and Jerry will dry-clean their compartments. Each worker can work with one car at a moment, and due to quality requirements, it is not allowed to switch from one car to another until the one is ready (this condition is independent for each worker, different workers can process cars in any order).

Help Ben to find out how the cars should be processed by the workers, so that all cars were finished as soon as possible.

#### Формат входных данных

The first line of the input file contains  $n$  ( $1 \leq n \leq 10\,000$ ). The following  $n$  lines contain two integer numbers each:  $a_i$  and  $b_i$  ( $1 \leq a_i, b_i \leq 10^5$ ).

#### Формат выходных данных

The first line of the output file must contain  $t$  — the number of minutes after the process has started when all cars will be washed and dry-cleaned. You must minimize this number.

The following  $n$  lines must contain two integer numbers each — the number of minutes after the start of the process when the washing and the dry-cleaning of the corresponding car must start, respectively.

If there are several optimal solutions, output any one.

#### Пример

carwash.in	carwash.out
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