

## Задача А. Проблема сапожника

Имя входного файла: `cobbler.in`  
Имя выходного файла: `cobbler.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В некоей воинской части есть сапожник. Рабочий день сапожника длится  $K$  минут. Заведующий складом оценивает работу сапожника по количеству починенной обуви, независимо от того, насколько сложный ремонт требовался в каждом случае. Дано  $n$  сапог, нуждающихся в починке. Определите, какое максимальное количество из них сапожник сможет починить за один рабочий день.

### Формат входного файла

В первой строке вводятся числа  $K$  (натуральное, не превышает 1000) и  $n$  (натуральное, не превышает 500). Затем идет  $n$  чисел — количество минут, которые требуются, чтобы починить  $i$ -й сапог (времена — натуральные числа, не превосходят 100).

### Формат выходного файла

Выведите одно число — максимальное количество сапог, которые можно починить за один рабочий день.

### Примеры

<code>cobbler.in</code>	<code>cobbler.out</code>
10 3 6 2 8	2
3 2 10 20	0

## Задача В. Выбор заявок

Имя входного файла: `request.in`  
Имя выходного файла: `request.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вы прекрасно знаете, что в ЛКШ.Зима 2016 лекции читают лучшие преподаватели мира. К сожалению, лекционных аудиторий у нас не так уж и много, поэтому каждый преподаватель составил список лекций, которые он хочет прочитать ЛКШатам. Чтобы ЛКШата, утром идя на завтрак, увидели расписание лекций, необходимо его составить прямо сейчас. И без вас нам здесь не справиться.

У нас есть список заявок от преподавателей на лекции для одной из аудиторий. Каждая заявка представлена в виде временного интервала  $[s_i, f_i)$  — время начала и конца лекции. Лекция считается открытым интервалом, то есть какая-то лекция может начаться в момент окончания другой, без перерыва. Необходимо выбрать из этих заявок такое подмножество, чтобы суммарно выполнить максимальное количество заявок. Учтите, что одновременно в лекционной аудитории, конечно же, может читаться лишь одна лекция.

### Формат входного файла

В первой строке вводится натуральное число  $N$ , не более 1000 — общее количество заявок на лекции. Затем вводится  $N$  строк с описаниями заявок — по два числа в каждом  $s_i$  и  $f_i$ . Гарантируется, что  $s_i < f_i$ . Время начала и окончания лекции — натуральные числа, не превышают 1440 (в минутах с начала суток).

### Формат выходного файла

Выведите одно число — максимальное количество заявок, которые можно выполнить.

### Примеры

<code>request.in</code>	<code>request.out</code>
1 5 10	1
3 1 5 2 3 3 4	2

## Задача С. Распечатка условий

Имя входного файла: `printing.in`  
Имя выходного файла: `printing.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Популярность окружной олимпиады по информатике растёт год от года. При этом организаторы должны заранее распечатать как условия задач, так и другие материалы олимпиады (анкеты, памятки и т.п.). В этом году они оценили объём печатной продукции в  $N$  листов.

Фирма, готовая размножить печатные материалы, предлагает следующие финансовые условия. Один лист она печатает за  $A_1$  рублей, 10 листов — за  $A_2$  рублей, 100 листов — за  $A_3$  рублей, 1000 листов — за  $A_4$  рублей, 10000 листов — за  $A_5$  рублей, 100000 листов — за  $A_6$  рублей, 1000000 листов — за  $A_7$  рублей. При этом не гарантируется, что один лист в более крупном заказе обойдется дешевле, чем в более мелком. И даже может оказаться, что для любой партии будет выгодно воспользоваться тарифом для одного листа.

Печать конкретного заказа производится или путем комбинации нескольких тарифов, или путем заказа более крупной партии. Например, 980 листов можно распечатать, заказав печать 9 партий по 100 листов плюс 8 партий по 10 листов, сделав 98 заказов по 10 листов, 980 заказов по 1 листу или заказав печать 1000 (или даже 10000 и более) листов, если это окажется выгоднее. Требуется по заданному объёму заказа в листах  $N$  определить минимальную сумму денег в рублях, которой будет достаточно для выполнения заказа.

### Формат входного файла

На вход программе сначала подается число  $N$  ( $1 \leq N \leq 2 \times 10^9$ ) — количество листов в заказе. В следующих 7 строках ввода находятся натуральные числа  $A_1, A_2, A_3, A_4, A_5, A_6, A_7$  соответственно ( $1 \leq A_i \leq 10^6$ ).

### Формат выходного файла

Выведите одно число — минимальную сумму денег в рублях, которая нужна для выполнения заказа. Гарантируется, что правильный ответ не будет превышать  $2 \times 10^9$ .

### Примеры

printing.in	printing.out
980 1 9 90 900 1000 10000 10000	882
980 1 10 100 1000 900 10000 10000	900

## Задача D. Последовательность

Имя входного файла: `sequence.in`  
Имя выходного файла: `sequence.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дана последовательность натуральных чисел  $a_1, a_2, \dots, a_n$ , и известно, что  $a_i \leq i$  для любого  $1 \leq i \leq n$ . Требуется определить, можно ли разбить элементы последовательности на две части таким образом, что сумма элементов в каждой из частей будет равна половине суммы всех элементов последовательности.

### Формат входного файла

В первой строке входного файла находится одно целое число  $n$  ( $1 \leq n \leq 40\,000$ ). Во второй строке находится  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq i$ ).

### Формат выходного файла

В первую строку выходного файла выведите количество элементов последовательности в любой из получившихся двух частей, а во вторую строку через пробел номера этих элементов. Если построить такое разбиение невозможно, выведите -1.

### Примеры

<code>sequence.in</code>	<code>sequence.out</code>
3	1
1 2 3	3

## Задача Е. Алиса и яблоки

Имя входного файла: `apples.in`  
Имя выходного файла: `apples.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Алисе в стране чудес попались  $n$  волшебных яблок. Про каждое яблоко известно, что после того, как его съешь, твой рост сначала уменьшится на  $a_i$  сантиметров, а потом увеличится на  $b_i$  сантиметров. Алиса очень голодная и хочет съесть все  $n$  яблок, но боится, что в какой-то момент ее рост станет равным нулю или еще меньше, и она пропадет совсем. Помогите ей узнать, можно ли съесть яблоки в таком порядке, чтобы в любой момент времени рост Алисы был больше нуля.

### Формат входного файла

В первой строке вводятся натуральные числа  $n$  и  $s$ , не более 1000 — число яблок и начальный рост Алисы. В следующих  $n$  строках вводятся пары натуральных чисел  $a_i, b_i$ , не больших 1000.

### Формат выходного файла

Если яблоки съесть нельзя, выведите число -1. Иначе выведите  $n$  чисел — номера яблок, в том порядке, в котором их нужно есть.

### Примеры

<code>apples.in</code>	<code>apples.out</code>
3 5 2 3 10 5 5 10	1 3 2
3 5 2 3 10 5 5 6	-1

## Задача F. Индикатор

Имя входного файла: `indic.in`  
Имя выходного файла: `indic.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Недавно Вася приобрел настольный калькулятор с жидкокристаллическим индикатором. Этот индикатор отображает  $n$  цифр с помощью  $n$  одинаковых элементов.



Отметим, что каждый элемент содержит семь полосок, каждая из которых может быть либо белой, либо черной. В частности, при отображении цифры «1» черными являются две полоски.

Вася — очень любознательный мальчик, поэтому он хочет узнать, какое максимальное и минимальное  $n$ -значное число может быть отображены на индикаторе его нового калькулятора так, чтобы черными были ровно  $k$  полосок.

Напишите программу, которая найдет ответ на Васин вопрос. Учитывайте при этом, что числа не могут содержать ведущие нули.

### Формат входного файла

Первая строка входного файла содержит два целых числа:  $n$  и  $k$  ( $1 \leq n \leq 100$ ,  $1 \leq k \leq 700$ ).

### Формат выходного файла

В первой строке выходного файла выведите минимальное число, во второй строке выходного файла выведите максимальное число. Если указанным образом не может быть представлено ни одно число, выходной файл должен содержать одну строку `NO SOLUTION`.

### Примеры

<code>indic.in</code>	<code>indic.out</code>
5 15	10117 97111
10 1	NO SOLUTION

## Задача G. Коммерческий калькулятор

Имя входного файла: `calc.in`  
Имя выходного файла: `calc.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Фирма OISAC выпустила новую версию калькулятора. Этот калькулятор берет с пользователя деньги за совершаемые арифметические операции. Стоимость каждой операции в долларах равна 5% от числа, которое является результатом операции.

На этом калькуляторе требуется вычислить сумму  $N$  натуральных чисел (числа известны). Нетрудно заметить, что от того, в каком порядке мы будем складывать эти числа, иногда зависит, в какую сумму денег нам обойдется вычисление суммы чисел (тем самым, оказывается нарушен классический принцип "от перестановки мест слагаемых сумма не меняется":-) ).

Например, пусть нам нужно сложить числа 10, 11, 12 и 13. Тогда если мы сначала сложим 10 и 11 (это обойдется нам в \$1.05), потом результат - с 12 (\$1.65), и затем - с 13 (\$2.3), то всего мы заплатим \$ 5, если же сначала отдельно сложить 10 и 11 (\$1.05), потом - 12 и 13 (\$1.25) и, наконец, сложить между собой два полученных числа (\$2.3), то в итоге мы заплатим лишь \$4.6.

Напишите программу, которая будет определять, за какую минимальную сумму денег можно найти сумму данных  $N$  чисел.

### Формат входного файла

Во входном файле записано число  $N$  ( $2 \leq N \leq 10^5$ ). Далее идет  $N$  натуральных чисел, которые нужно сложить, каждое из них не превышает 10000.

### Формат выходного файла

В выходной файл выведите, сколько денег нам потребуется на нахождение суммы этих  $N$  чисел. Результат должен быть выведен с двумя знаками после десятичной точки.

### Примеры

<code>calc.in</code>	<code>calc.out</code>
4 10 11 12 13	4.60
2 1 1	0.10

## Задача Н. Сушка

Имя входного файла: `drying.in`  
Имя выходного файла: `drying.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Тетя Люба только что постирала все белье —  $n$  вещей — и теперь перед ней стоит непростая задача — как его высушить за минимальное время. Сразу после стирки,  $i$ -я постиранная вещь имеет влажность  $a_i$ . Если она сушится на веревке, то за минуту ее влажность уменьшается на 1, а если на батарее — то на  $k$  (если влажность была меньше  $k$ , то она становится равной 0).

Причем веревок у тети Любы много (хватает для одновременной сушки всех вещей), а батарея только одна, причем такая маленькая, что на ней нельзя сушить две вещи одновременно. Тетя Люба может каждую минуту выбрать ровно одну вещь и повесить ее сушиться на батарею. За какое минимальное время она сможет высушить все вещи?

### Формат входного файла

Первая строка содержит число  $n$  ( $1 \leq n \leq 100\,000$ ). Вторая строка содержит числа  $a_i$  ( $1 \leq a_i \leq 10^9$ ). Третья строка содержит число  $k$  ( $1 \leq k \leq 10^9$ ).

### Формат выходного файла

Выведите одно число — минимальное число минут, за которые можно высушить все вещи.

### Пример

drying.in	drying.out
3 2 3 9 5	3
3 2 3 6 5	2



## Задача I. Инверсии и Сережа

Имя входного файла: john.in  
Имя выходного файла: john.out  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Недавно Сережа изучил понятие инверсии. Инверсией в последовательности чисел  $s_k$  называется пара  $s_i, s_j$ , такая что  $i < j$  и  $s_i > s_j$ .

Петя дал Сереже  $n$  карточек. На каждой карточке написано два числа: одно красное и одно синее. Сережа хочет применить свои знания об инверсиях к этому набору карточек.

Сережа хочет выложить перед собой карточки в таком порядке, чтобы количество инверсий было как можно меньше. При этом он считает число инверсий между числами одного цвета. Таким образом, он хочет выложить карточки в таком порядке, чтобы если рассмотреть отдельно красные числа в этом порядке и отдельно синие числа в этом порядке, суммарное число инверсий было как можно меньше.

### Формат входного файла

Первая строка входного файла содержит число  $n$  — число карточек в наборе ( $1 \leq n \leq 100\,000$ ). Следующие  $n$  строк описывают карточки, по одной на строке,  $i$ -я строка содержит целые числа  $r_i$  и  $b_i$  ( $1 \leq r_i, b_i \leq 10^9$ ) — соответственно красное и синее число на  $i$ -й карточке.

### Формат выходного файла

Выведите одно число — минимальное возможное число одноцветных инверсий.

### Примеры

john.in	john.out
3 10 3 20 2 30 1	3
4 2 2 5 25 2 1 10 9	1

## Задача J. Свиновоз

Имя входного файла: `journey.in`  
Имя выходного файла: `journey.out`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 мегабайт

Фермер Паша работает на свиноферме около города А. Он хочет съездить в гости к своему другу, который живет в городе В. Во время поездки он планирует посетить  $n$  деревень, расположенных вдоль дороги и продать жителям этих деревень свинину — не просто же так ехать.

Он взял  $n$  туш свиней и планирует продавать по одной туше в каждой из деревень. Известно, что жители  $j$ -й деревни готовы купить тушу свиньи по цене  $p_j$  рублей за килограмм. При этом  $j$ -я деревня располагается на расстоянии  $d_j$  километров от города вдоль дороги, по которой поедет фермер Паша.

Известна масса каждой туши, при этом фермер Паша знает, что если везти мясо, то на каждый килограмм свинины необходимо потратить дополнительные  $t$  рублей на километр на бензин.

Помогите фермеру Паше решить, в какой деревне какую тушу продать, чтобы максимизировать прибыль.

### Формат входного файла

Первая строка входного файла содержит числа  $n$  ( $1 \leq n \leq 1000$ ) и  $t$  ( $1 \leq t \leq 10^9$ ). Вторая строка содержит  $n$  целых чисел  $w_i$  ( $1 \leq w_i \leq 10^9$ ) — массы туш. Третья строка содержит  $n$  целых чисел  $d_j$  ( $1 \leq d_j \leq 10^9$ ) — расстояния до деревень от города А. Четвертая строка содержит  $n$  целых чисел  $p_j$  ( $1 \leq p_j \leq 10^9$ ) — цены на свинину в деревнях.

### Формат выходного файла

Выведите  $n$  чисел,  $j$ -е выведенное число должно быть номером туши, которую следует продать в  $j$ -й деревне. Фермер Паша обязан продать ровно одну тушу в каждой деревне, даже если в итоге окажется в убытке. В конце концов, жители не должны голодать.

### Пример

journey.in	journey.out
3 1 10 20 15 10 20 30 50 70 60	3 2 1

## Задача К. Магазин

Имя входного файла: `shop.in`  
Имя выходного файла: `shop.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

У Билла большая семья: трое сыновей, девять внуков. И всех надо кормить. Поэтому Билл раз в неделю ходит в магазин.

Однажды Билл пришел в магазин и увидел, что в магазине проводится акция под названием «каждый  $k$ -й товар бесплатно». Изучив правила акции, Билл выяснил следующее. Пробив на кассе товары, покупатель получает чек. Пусть в чеке  $n$  товаров, тогда  $n/k$  округленное вниз самых дешевых из них достаются бесплатно.

Например, если в чеке пять товаров за 200, 100, 1000, 400 и 100 рублей, соответственно, и  $k = 2$ , то бесплатно достаются оба товара по 100 рублей, всего покупатель должен заплатить 1600 рублей.

Билл уже выбрал товары, и направился к кассе, когда сообразил, что товары, которые он хочет купить, можно разбить на несколько чеков, и благодаря этому потратить меньше денег.

Помогите Биллу выяснить, какую минимальную сумму он сможет заплатить за выбранные товары, возможно разбив их на несколько чеков.

### Формат входного файла

Первая строка входного файла содержит два целых числа  $n$ ,  $k$  ( $1 \leq n \leq 100\,000$ ,  $2 \leq k \leq 100$ ) — количество товаров, которые хочет купить Билл и параметр акции «каждый  $k$ -й товар бесплатно».

Следующая строка содержит  $n$  целых чисел  $a_i$  ( $1 \leq a_i \leq 10\,000$ ) — цены товаров, которые покупает Билл.

### Формат выходного файла

Выведите в выходной файл одно число — минимальную сумму, которую должен заплатить Билл за товары.

### Примеры

<code>shop.in</code>	<code>shop.out</code>
5 2 200 100 1000 400 100	1300

В приведенном примере Билл может разбить товары на два чека: в один чек пойдут товары за 1000 и за 400 рублей, товар за 400 рублей в этом чеке достанется бесплатно, а в другой — остальные товары, в нем бесплатно достанется один товар за 100 рублей. Итого Биллу придется заплатить 1300 рублей.