

Задача А. Проблема сапожника

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

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 входных данных

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 выходных данных

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: 64 мегабайта

Вы прекрасно знаете, что в ЛКШ.Зима 2017 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 входных данных

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 выходных данных

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. Планирование заданий

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

Имеется некоторое множество заданий и один исполнитель. На выполнение одного задания уходит единица времени. Задания можно выполнять начиная с момента времени 0. У каждого задания есть две характеристики: d_i и w_i . Если задание не было выполнено к моменту времени d_i , взимается штраф в размере w_i . Требуется минимизировать суммарный штраф.

Формат входных данных

Первая строка входного файла содержит натуральное число n — количество заданий ($1 \leq n \leq 1000$). Следующие n строк содержат по два натуральных числа, разделенных пробелом — d_i и w_i ($0 \leq d_i, w_i \leq 1000$).

Формат выходных данных

Выведите одно число — минимальный суммарный штраф.

Примеры

<code>schedule.in</code>	<code>schedule.out</code>
2 1 1 1 2	1
1 0 5	5

Задача D. Алиса и яблоки

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

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 входных данных

В первой строке вводятся натуральные числа n и s ($1 \leq n \leq 1000$, $1 \leq s \leq 100000$) — число яблок и начальный рост Алисы. В следующих n строках вводятся пары натуральных чисел a_i , b_i , не больших 1000.

Формат выходных данных

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 Е. Разница последовательностей

Имя входного файла: `setdiff.in`
Имя выходного файла: `setdiff.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Вам заданы две строго возрастающие последовательности A и B . Найдите все элементы последовательности A , не содержащиеся в последовательности B .

Формат входных данных

В первой строке входного файла записано число n – длина последовательности A . Во второй строке записаны через пробелы элементы последовательности A . В третьей строке записано число m – длина последовательности B . Во четвертой строке записаны через пробелы элементы последовательности B . Элементы обеих последовательностей записаны в возрастающем порядке, элементы внутри каждой последовательности различны (но могут быть совпадающие элементы из разных, иначе задача была бы странной :) и не превосходят 10^9 по модулю. $1 \leq n, m \leq 5 \cdot 10^5$

Формат выходных данных

Выведите через пробелы все элементы последовательности A , не содержащиеся в последовательности B .

Примеры

<code>setdiff.in</code>	<code>setdiff.out</code>
5 1 2 3 4 5 5 0 3 5 6 7	1 2 4
3 1 2 3 3 1 2 3	

Задача F. Город Че

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

В центре города Че есть пешеходная улица — одно из самых популярных мест для прогулок жителей города. По этой улице очень приятно гулять, ведь вдоль улицы расположено n забавных памятников.

Девочке Маше из города Че нравятся два мальчика из ее школы, и она никак не может сделать выбор между ними. Чтобы принять окончательное решение, она решила назначить обоим мальчикам свидание в одно и то же время. Маша хочет выбрать два памятника на пешеходной улице, около которых мальчики будут ее ждать. При этом она хочет выбрать такие памятники, чтобы мальчики не увидели друг друга. Маша знает, что из-за тумана мальчики увидят друг друга только в том случае, если они будут на расстоянии не более r метров.

Маше заинтересовалась, а сколько способов есть выбрать два различных памятника для организации свиданий.

Формат входных данных

В первой строке входного файла находятся два целых числа n и r ($2 \leq n \leq 300\,000$, $1 \leq r \leq 10^9$) — количество памятников и максимальное расстояние, на котором мальчики могут увидеть друг друга.

Во второй строке задано n положительных чисел d_1, \dots, d_n , где d_i — расстояние от i -го памятника до начала улицы. Все памятники находятся на разном расстоянии от начала улицы. Памятники приведены в порядке возрастания расстояния от начала улицы ($1 \leq d_1 < d_2 < \dots < d_n \leq 10^9$).

Формат выходных данных

Выведите одно число — число способов выбрать два памятника для организации свиданий.

Примеры

<code>che.in</code>	<code>che.out</code>
4 4 1 3 5 8	2

Замечание

В приведенном примере Маша может выбрать памятники 1 и 4 или памятники 2 и 4.

Задача G. Дюбели и сверла

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

Петя хочет повесить картину на стену. Для этого ему нужно просверлить в стене дырку, вбить в нее дюбель и вкрутить в него саморез. Петя покопался в кладовке и нашел n сверел и m дюбелей. Петя хочет найти сверло и дюбель одного радиуса. Однако, таких может не быть, в этом случае он хочет подобрать сверло и дюбель так, чтобы разность их диаметров была как можно меньше. Помогите Пете.

Формат входных данных

В первой строке входного файла заданы целые числа n и m ($1 \leq n, m \leq 10^5$). Во второй строке заданы n целых чисел — диаметры сверел. В следующей строке заданы m целых чисел — диаметры дюбелей. Диаметры заданы в неубывающем порядке, все диаметры — числа от 1 до 10^9 .

Формат выходных данных

Выведите минимальную возможную разницу диаметров сверла и дюбеля.

Примеры

<code>drill.in</code>	<code>drill.out</code>
3 2 1 8 15 5 6	2
3 3 1 3 5 3 4 6	0

Задача N. Еще одна строковая задача

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Строка называется бинарной, если она состоит только из символов 0 и 1.

Строка v называется подстрокой строки w , если она имеет ненулевую длину, и ее можно прочитать, начиная с некоторой позиции, в строке w . Например, у строки 010 есть шесть подстрок: 0, 1, 0, 01, 10, 010. Две подстроки считаются различными, если их позиции вхождения различны. Другими словами, каждую подстроку нужно учитывать столько раз, сколько она встречается.

Дана бинарная строка s . Ваша задача — найти количество ее подстрок, содержащих ровно k единиц.

Формат входных данных

В первой строке записано единственное целое число k ($0 \leq k \leq 10^6$). Во второй строке записана непустая бинарная строка s . Длина s не превосходит 10^6 символов.

Формат выходных данных

Выведите одно целое число — количество подстрок данной строки, содержащих ровно k единиц.

Примеры

<code>stdin</code>	<code>stdout</code>
1 1010	6