

Задача А. Капитал

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

Чтобы преумножить свой капитал, Елена Андреевна придумала новую схему заработка.

Пусть сейчас ее капитал составляет a рублей. Тогда, путем хитрых махинаций, она может превратить данную сумму в новую, размером b . Но на число b накладываются ограничения: его сумма цифр должна быть равна числу a , и количество цифр не должно превосходить n .

Помогите Елене Андреевне как можно больше преумножить свой капитал — найти максимальное такое число b !

Формат входного файла

В единственной строке входного файла дано два числа a, n ($1 \leq a \leq 10^6; 1 \leq n \leq 10^5$) — требуемая сумма цифр и максимальное количество цифр в числе b .

Формат выходного файла

Выведите максимальное число b , которое удовлетворяет описанным ограничениям.

Если такого числа не существует, выведите -1 .

Пример

<code>capital.in</code>	<code>capital.out</code>
8 1	8
14 2	95

Задача В. Стеллаж с книгами

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

Уже больше года Фома пытается добиться согласия Тани и во всем ей помогает. Сегодня она попросила его помочь поставить книги на ее новый стеллаж.

Стеллаж представляет собой таблицу из n строк и m столбцов, на пересечении каждой строки и столбца можно поставить ровно одну книгу. У Тани есть книги с 5 разными цветами корешков. Она попросила Фому поставить их на стеллаж так, что в любом квадрате 3×3 в «кресте» стояли книги всех пяти различных цветов, а в «рамке» стояли книги не более четырех различных цветов (под «крестом» будем понимать все клетки квадрата 3×3 , кроме угловых, а под «рамкой» все клетки квадрата, кроме центральной). Можно считать, что книг с каждым цветом корешка у Тани бесконечно много.

Фома не силен в таких задачах, но ему очень нужно помочь Тане. Подскажите ему, как расставить так книги, или скажите, что так сделать нельзя.

Формат входного файла

В первой и единственной строке входного файла дано два натуральных числа n, m ($1 \leq n, m \leq 100$).

Формат выходного файла

В выходной файл выведите nm чисел. j -е число в i -й строке ($1 \leq i \leq n; 1 \leq j \leq m$) означает цвет соответствующей книги (цвет - натуральное число от 1 до 5 включительно).

Если существует несколько ответов, выведите любой. Если ответа не существует, выведите -1.

Пример

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

Задача С. Хобби

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

Любимое хобби Льва Романовича — соединять точки на плоскости. Однако за последнее время это занятие успело ему наскучить, и он решил разнообразить свое свободное время. А именно, теперь он хочет соединять точки так, чтобы ни одна пара отрезков не пересекалась. Отрезки называются пересекающимися, если они имеют хотя бы одну общую точку (в том числе начало или конец).

Учитель информатики узнал про это хобби и попросил написать вас программу, которая строит максимальное количество отрезков по заданному наборе точек по описанным выше правилам.

Формат входного файла

В первой строке входного файла дано число n ($1 \leq n \leq 1000$) — количество точек. В следующих n строках дано описание координат i -ой точки: x_i, y_i ($-10^5 \leq x_i, y_i \leq 10^5$). Гарантируется, что никакие две точки не совпадают.

Формат выходного файла

В качестве ответа на задачу выведите в первой строке число k — максимальное количество отрезков. В следующих k строках выведите описание i -го отрезка: пару чисел a_i, b_i ($1 \leq a_i, b_i \leq n, a_i \neq b_i$) — номера точек. Точки нумеруются с единицы в том порядке, в котором они представлены во входных данных.

Пример

<code>hobby.in</code>	<code>hobby.out</code>
8	4
2 2	1 5
2 5	2 3
3 3	4 8
4 5	6 7
4 2	
5 6	
7 4	
6 0	

Задача D. Урок физкультуры

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

На очередном уроке физкультуры Фома придумал интересное занятие.

После того как школьники выстроились в шеренгу, он дал каждому свой персональный номер. Теперь он просит школьников разбиться на группы, чтобы выполнялись условия:

- Каждая группа — это подотрезок шеренги. То есть школьники в группе стоят в шеренге подряд.
- Группы не пересекаются.
- В каждой группе последовательность персональных номеров школьников либо нестрого возрастает, либо нестрого убывает.

Перебрав несколько способов, Фома заинтересовался, сколько их может быть всего. Он просит вас написать программу, которая вычислит, сколько есть способов разбиения школьников на такие группы.

Формат входного файла

В первой строке входного файла задано число n ($1 \leq n \leq 5000$) — количество школьников. Во второй строке содержатся n чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — персональный номер каждого школьника.

Формат выходного файла

В единственной строке выходного файла выведите одно число — количество способов разделить школьников на группы. Ответ может быть очень большой, поэтому нужно вывести его по модулю $10^9 + 7$.

Пример

<code>lesson.in</code>	<code>lesson.out</code>
4 1 2 3 4	8
5 1 2 2 2 1	15
5 1 2 1 2 1	8

Задача Е. Фома и занимательная математика

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

Сегодня утром Фома искал Таню по всей школе, но никак не мог найти. Тогда он начал заходить по очереди во все кабинеты.

В один момент, зайдя в кабинет математики, Фома обнаружил на доске какие-то каракули: $a^x + b^x + c^x = d \wedge x \in \{1, 2, 3\}$. Тут же на помощь был вызван «Усач». Начал он с того, что объяснил, что « \wedge » — логическое «и», а « \in » означает принадлежность множеству. Дальше Валя начал рассказывать, что с этим можно делать, но как только Фома понял, что вместо a , b , c и d можно подставить любые числа, он стал спрашивать у Вали корень получившегося уравнения.

Валя не любит лишний раз общаться с Олегом Евгеньевичем, поэтому просит вас помочь. Напишите программу, которая по заданным a , b , c и d , определит корень уравнения.

Формат входного файла

В единственной строке входного файла даны четыре числа a , b , c и d ($-500 \leq a, b, c \leq 500$; $-10^9 \leq d \leq 10^9$) — параметры уравнения.

Формат выходного файла

В единственной строке выходного файла выведите одно число x — корень уравнения. Если однозначно его найти нельзя, выведите -1 .

Пример

<code>numbers.in</code>	<code>numbers.out</code>
1 1 1 3	-1
1 2 3 6	1
1 2 3 7	-1
1 -1 -2 6	2

Задача F. Димас и конструктор

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

Леха «Псих» подарил Димасу набор конструктора. Несмотря на возраст, он сильно обрадовался подарку, ведь в Батайске таких игрушек никогда не было.

Частью набора был забор, который больше всего понравился Димасу, так что он решил его модернизировать, используя другие детали из набора. Все детали в наборе одинаковые, а забор представляет из себя стоящие в ряд n деталей, на каждую из которых поставлено еще сколько-то деталей. Назовем такой «столбик» сегментом забора. Высота сегмента вычисляется как количество деталей в нем. Модернизация состоит в том, что Димас ставит детали сверху на некоторые сегменты забора. При этом он не трогает детали, которые уже стояли. Теперь Димас хочет понять, какую наибольшую высоту забора он может получить. Высота забора совпадает с высотой наименьшего сегмента.

Димас никогда не слышал о программировании, поэтому просит вас помочь ему с этой задачей.

Формат входного файла

В первой строке входного файла задано два числа n и k ($1 \leq n \leq 10^5; 1 \leq k \leq 10^9$) — ширина стены в деталях и количество лишних деталей соответственно. Во второй строке содержатся n чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — высота каждого сегмента.

Формат выходного файла

В единственной строке выходного файла выведите одно число h — максимальную высоту забора, которую может получить Димас после модернизации.

Пример

<code>repair.in</code>	<code>repair.out</code>
3 3 1 2 3	3
4 6 1 2 2 1	3

Задача G. Последовательность

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

Снова, снова эти школьники занимаются чем-то неподобающим на уроках физкультуры! Сегодня они совсем обезумели и притащили на физкультуру ноутбуки, и начали обсуждать какие-то раунды какого-то CodeHorses.

Фома не совсем понял, что происходит, и на всякий случай объявил, что любой желающий может бросить ему вызов: в молодости Фома был боксером, и он был уверен, что положит любого смельчака в первом же раунде. Но когда ему выдали ноутбук, Фома понял, что что-то неладно, и попросил вас о помощи. Задача, которую ему нужно решить, звучит так:

Есть последовательность a из n целых чисел $a_1 \dots a_n$. Вам нужно найти такую последовательность b из $n + 2$ целых чисел $b_0 \dots b_{n+1}$, что $a_i = b_{i-1} + b_i + b_{i+1}$.

Формат входного файла

В первой строке входного файла дано одно целое число n ($1 \leq n \leq 3 \cdot 10^4$).

Во второй строке входного файла даны n целых чисел a_i ($-3 \cdot 10^4 \leq a_i \leq 3 \cdot 10^4$).

Формат выходного файла

В единственной строке выходного файла выведите $n + 2$ целых числа b_i ($-10^9 \leq b_i \leq 10^9$).

Пример

<code>sequence.in</code>	<code>sequence.out</code>
3 1 1 1	0 0 1 0 0

Задача Н. Игра со строкой

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

Недавно Альберт Моисеевич обновил свой iPad до iOS 8.1. И, недолго думая, он установил новую игру, вышедшую буквально вчера.

Суть игры довольно проста: дана строка s , состоящая из строчных латинских символов. Игроку предлагается за отведенное время найти в ней подстроку, у которой существует максимальное количество непересекающихся вхождений в строку s . Вхождением подстроки t в строку s будем называть пару символов (l, r) , такую, что $l \leq r$ и подстрока строки s с l по r символ включительно равна строке t . Непересекающимися будем называть два вхождения (l_1, r_1) и (l_2, r_2) , такие, что отрезок $[l_1, r_1]$ не пересекается с отрезком $[l_2, r_2]$.

За найденную подстроку игроку начисляется количество очков, равное ее длине. Альберт Моисеевич хочет набрать как можно больше очков, поэтому он просит вас найти подстроку максимальной длины, которая будет удовлетворять описанному условию. Помогите ему!

Формат входного файла

В первой и единственной строке входного дана строка s ($1 \leq |s| \leq 10^5$), состоящая только из строчных латинских букв.

Формат выходного файла

В единственной строке выходного файла выведите максимальную длину подстроки, которая удовлетворяет условию.

Пример

<code>strings.in</code>	<code>strings.out</code>
abacaba	1
abab	2

Задача I. Дерево

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

Недавно на уроке информатики Саша узнала о двоичных деревьях поиска.

Двоичным деревом поиска называется двоичное дерево, в каждой вершине которого написано число, называемое ключом этой вершины. Причем ключ, записанный в вершине, больше всех ключей, записанных в левом поддереве этой вершины, но меньше всех ключей, записанных в ее правом поддереве.

Такая организация чисел позволяет эффективно реализовать структуру данных «множество чисел». То есть мы можем построить двоичное дерево поиска по некоторому множеству чисел, а потом эффективно проверять, принадлежит ли то или иное число этому множеству.

Саша уже реализовала двоичное дерево поиска. Но вдруг она поняла, что хранить множество чисел — не очень интересно. Намного интереснее хранить структуру данных, представляющую собой так называемый «ассоциативный массив». Это такая структура данных, в которой для каждого ключа хранится некоторое другое число, называемое значением этого ключа. То есть, в каждой вершине дерева находится по два числа — ключ и значение, причем это дерево является двоичным деревом поиска по ключам вершин.

Так как Саша уже реализовала двоичное дерево поиска, то она хочет придумать такую структуру ассоциативного массива, при которой ей пришлось бы хранить в вершине только одно число.

Определим такую операцию над двоичным деревом поиска, хранящим пары ключ-значение: Новым ключом левого сына данной вершины становится ключ данной вершины, а новым ключом ее правого сына — ее значение. Если у вершины до выполнения операции не было левого или правого сына, то он появляется и получает соответствующий ключ. Новым ключом корня становится некоторое число T .

Вам дано двоичное дерево поиска, но не даны значения, хранящиеся в вершинах. Выведите любой подходящий набор таких значений (где каждое число целое и лежит в диапазоне от 1 до 10^9) при которых дерево, полученное операцией, описанной выше, будет являться деревом поиска. Или определите, что это невозможно.

Формат входного файла

В первой строке входного файла через пробел даны два числа n и T ($1 \leq n \leq 10^5$, $1 \leq T \leq 10^9$) — количество вершин в исходном дереве и ключ корня в новом дереве.

В следующих n строках перечислена информация о вершинах. На i -ой строке находятся три числа l_i , r_i , a_i ($1 \leq l_i \leq n$, $0 \leq r_i \leq n$, $0 \leq a_i \leq 10^9$) — номер вершины, являющейся левым потомком вершины i , номер вершины, являющейся ее правым потомком и ключ этой вершины. Если у вершины нет какого-то потомка, то вместо соответствующего числа в файле записан 0. Первая вершина является корнем дерева.

Формат выходного файла

Если решения не существует, то в первой строке выходного файла выведите -1. Иначе, выведите n чисел — для каждой вершины выведите значение в ней. Каждое число должно быть целым и лежать в диапазоне от 1 до 10^9

Пример

tree.in	tree.out
3 4 2 3 2 0 0 1 0 0 5	6 3 7