

Задача А. Ахтунг!

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

Ахтунг! Механическую няню кто-то включил, и теперь она бежит за Пином, пытаясь окружить его лаской и заботой! Во дворе Пина есть n бункеров, и он рад бы спрятаться в одном из них, но в каждом бункере у него есть очень важное и очень срочное дело.

Поэтому Пин просит вас составить маршрут его передвижения между бункерами, чтобы он смог посетить каждый ровно один раз, и вернуться в начало. При этом начать Пин может с любого бункера.

Кроме того, так как Пин сам писал программу перехвата для няни и собирал ее двигатель, то он точно знает, что будет пойман, если будет бежать от одного бункера к другому не по прямой, либо если он дважды пробежит через одно и то же место, то есть пересечёт или коснётся отрезка пути, который он уже пробежал.

Формат входного файла

Во входном файле дано описание двора Пина.

В первой строке входного файла находится одно целое число n ($1 \leq n \leq 100\,000$) — количество бункеров во дворе Пина.

В следующих n строках записано по два целых числа x_i и y_i ($|x_i|, |y_i| \leq 10^9$) — координаты входа в i -ый бункер. Вход в бункер настолько мал по сравнению с размерами двора, что считается точкой. Никакие два бункера не лежат в одной точке.

Формат выходного файла

В выходной файл выведите перестановку из n чисел — номера бункеров в порядке их посещения Пином, либо 'No solution', если не существует маршрута, по которому Пин может пробежать и не быть пойманным няней.

Примеры

<code>achtung.in</code>	<code>achtung.out</code>
4 0 0 0 1 1 0 1 1	1 2 4 3
3 0 0 0 1 0 2	No solution

Задача В. Печеньки

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

Как-то раз Бараш решил угостить Ньюшу овсяным печеньем. Ньюша с удовольствием согласилась, но предложила поделить их как можно честнее. Так получилось, что у Бараша было $2n$ печенек, и было бы логично разделить их поровну, но тут обнаружилось, что печеньки имеют разный вес.

И Ньюша предложила следующую игру. Игра состоит из n ходов, в ходе которых игроки берут по одной печеньке в некотором порядке. На первом ходу первой берёт печеньку Ньюша. После каждого хода Бараш и Ньюша взвешивают уже выбранные печеньки и тот, у кого на конец хода они весят меньше, берёт в следующий раз первым. В случае равенства весов в конце хода, порядок остается прежним.

Несмотря на то, что в конце каждый игрок получит одинаковое количество печенек, их суммарный вес может различаться. Ньюша просит подсчитать вас максимальный суммарный вес печенек, который она может гарантированно набрать своими ходами, независимо от ходов Бараша.

Формат входного файла

Первая строка входного файла содержит единственное число n ($1 \leq n \leq 7$) — сколько печенек в конце игры получают и Ньюша, и Бараш.

Вторая строка входного файла содержит $2n$ натуральных чисел a_i ($1 \leq a_i \leq 100$) — веса печенек в граммах, записанные в неубывающем порядке.

Формат выходного файла

В выходной файл выведите единственное число — максимальный суммарный вес печенек, который может гарантированно получить Ньюша.

Примеры

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

Задача С. Покраска забора

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

В один прекрасный день Совунья решила покрасить забор вокруг своего дерева на северо-западе. Когда-то давно его красили, потом краска облезала, потом его снова красили, потом краска снова облезала... В общем, к настоящему моменту забор совершенно потерял свой первоначальный вид.

Сейчас забор представляет собой n пронумерованных досок, каждая имеет цвет от 1 до k . Для начала Совунья решила выбрать цвет, который будут иметь все доски после завершения покраски. После того, как она сделает это, она будет каждый день красить в этот цвет все доски на каком-то отрезке, то есть с номерами не меньше l и не больше r . Совунья хочет так выбрать цвет, чтобы покрасить забор за наименьшее количество дней.

Казалось бы, что может быть проще? Однако, есть два правила, которые Совунья будет соблюдать при покраске. Первое правило заключается в том, что за день она может покрасить не больше s досок (она, хоть и обожает спорт, уже не может очень много заниматься физическим трудом). Второе правило — краску надо экономить, поэтому нельзя красить в какой-то цвет доску, которая уже имеет этот цвет. Соответственно, если на той части забора (всех досках с номерами от l до r), которую Совунья собралась красить в какой-то цвет, есть хотя бы одна доска этого цвета, то весь отрезок красить в этот цвет нельзя, нужно выбирать другой.

Сейчас Совунья хочет знать, сколько дней у нее уйдет на покраску забора.

Формат входного файла

В первой строке входного файла даны три числа: n , k и s ($1 \leq n, k, s \leq 100\,000$) — количество досок в заборе, количество различных возможных цветов и максимальная длина раскрашиваемого отрезка.

Вторая строка содержит n целых чисел c_i ($1 \leq c_i \leq k$), разделённых пробелами — цвета забора.

Формат выходного файла

Выведите одно целое число — минимальное количество дней, за которое Совунья покрасит забор.

Примеры

<code>fence.in</code>	<code>fence.out</code>
6 3 2 1 2 3 3 1 2	2

Задача D. Бикфордов шнур

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

В прекрасный солнечный день Ёжик прогуливался по своей любимой полянке. Однако, он увидел нечто необычное, чего на полянке раньше не было. Это нечто издали казалось похожим на клубок верёвок. Однако, подойдя поближе, распутав его и изучив материал, из которого он изготовлен, Ёжик пришёл к выводу, что это — бикфордов шнур.

Внимательно посмотрев на него, Ёжик пришёл к выводу, что, похоже что, тому, кто его сделал, было скучно и у него было много свободного времени, так как шнур был запутан, и на нём было завязано огромное количество узлов! После кропотливого пересчёта, выяснилось, что всего имеется n узлов и m соединений между узлами. Каждое из них соединяет два узла и горит равномерно. При поджигании узла загораются все соединения, которые его содержат.

Однако, Ёжику находка не понравилась, и он решил уничтожить её. Для этого он решил использовать так кстати завалившуюся у него в кармане спичку. Он может поджечь ей один из узлов, и вся эта конструкция начинает гореть. Ёжик — занятой смешарик, и хочет, чтобы эта конструкция сгорела как можно быстрее. Помогите ему определить, какой узел для этого надо поджечь.

Формат входного файла

Первая строка входного файла содержит два целых числа n и m ($2 \leq n \leq 100, 1 \leq m \leq \frac{n \cdot (n-1)}{2}$) — количество узлов и верёвок между узлами, соответственно.

Следующие m строк содержат описания соединений между узлами — три целых числа a_i, b_i и t_i ($1 \leq a_i, b_i \leq n, a_i \neq b_i, 1 \leq t_i \leq 1000$) — номера узлов, которые соединяет i -ая верёвка и за сколько секунд эта верёвка полностью сгорит, будучи подожжённой с одного из концов.

Каждую пару узлов соединяет максимум одна верёвка. Гарантируется, что весь шнур может сгореть, если поджечь любой его узел.

Формат выходного файла

В входной файл выведите пару чисел — номер узла, который надо поджечь, и за какое время при этом сгорит весь шнур.

Примеры

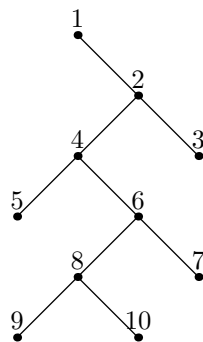
<code>fuse.in</code>	<code>fuse.out</code>
4 4 1 2 3 2 3 4 1 3 5 1 4 1	2 4

Задача Е. Лямбда-растение

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

Недавно перед домом Лосяша упал метеорит. На следующее утро Лосяш обнаружил, что в его палисаднике выросло новое, неизвестное ему, растение. Шли дни, растение росло, и однажды ночью Лосяш увидел, что некоторые его части светятся.

А именно, растение состоит из большого количества шарообразных клубней, некоторые из которых соединены стебельками. После тщательного анализа Лосяш установил, что клубни соединены следующим образом — если сопоставить каждому клубню номер, то клубень с номером один, соответствующий корню растения, будет соединен с клубнем номер два, клубень два - с номерами один, три и четыре, а клубень i с номером больше двух, соединен с $i - 1$, если i нечетно, либо с $i - 2$, $i + 1$ и $i + 2$ — если i четно.



Когда же Лосяш стал исследовать закономерности свечения, то обнаружил, что если он дотрагивался до клубней с номерами u и v , то светиться начал клубень с минимальным номером, находящийся на кратчайшем пути между u -м и v -м клубнями.

Так как пока что растение Лосяша не очень большое, то он попросил вас вычислить номер клубня, который начнет светиться, если он дотронется до клубней u и v .

Формат входного файла

В первой строке входного файла содержится одно целое число n ($1 \leq n \leq 100$) — количество пар клубней, интересных Лосяшу. В следующих n строках записано по два числа v_i и u_i ($1 \leq u_i, v_i \leq 10^9, u_i \neq v_i$) — номера i -й пары клубней.

Формат выходного файла

В i -й строке выходного файла выведите номер клубня, который начнет светиться, если дотронуться до клубней u_i и v_i .

Примеры

<code>lplant.in</code>	<code>lplant.out</code>
4	1
1 2	2
3 4	4
5 6	8
8 10	

Задача F. Священное число

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

Все Смешарики очень любят слушать рассказы старого доброго ворона-артиста Кар Карыча. Особенно им нравятся его рассказы про путешествия в далёкие страны. Вот и сегодня они пришли к нему в гости выпить вкусного чаю с печенюшками и послушать его рассказы про таинственные далёкие земли. Кар Карыч не разочаровал их и на этот раз и рассказал им про своё путешествие на острова.

На далёком-далёком острове в Тихом океане каждый год жители племени Улюлю-ололо отмечают праздник 1000 богов. Число 1000 — священное число племени, и жители верят, что все остальные числа подчиняются ему. Чтобы убедиться в этом, каждый год, во время праздника, вождь загадывает главному жрецу натуральное число n , из которого он должен получить священное число. К сожалению, главный жрец — плохой математик, и он умеет делать всего два действия над числом — вычесть 1 или умножить его на 2. Солнце на острове садится рано, поэтому главный жрец не успеет сделать больше 50 операций над числом. Если жрец не сможет получить священное число, боги разгневаются и накажут всё племя, но ещё раньше племя разозлится и накажет жреца, и их кара будет страшна!

Нюше стало очень жалко главного жреца, она думает, что не из каждого числа можно получить 1000, тем более при помощи всего 2 операций. Кар Карыч хочет успокоить Нюшу и доказать, что главный жрец всегда может отвести гнев богов от племени, получив священное число. К сожалению, Кар Карыч, как и главный жрец, тоже не очень хороший математик, поэтому он просит помочь вас!

Формат входного файла

В единственной строке входного файла дано натуральное число n ($1 \leq n \leq 100$).

Формат выходного файла

В первой строке выходного файла выведите одно число k — количество действий над числом n . В последующих k строках выведите сами действия в формате `'-'`, если из числа вычли 1 и `'*'`, если число умножалось на 2.

Примеры

<code>number.in</code>	<code>number.out</code>
8	9 * * * - * - * * *

Задача G. Запасы на зиму

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

Копатыч — добрый и хозяйственный медведь-огородник, который выращивает продукты для всех Смешариков. На зиму он обычно впадает в спячку, но остальным Смешарикам он всегда оставляет много разных запасов, чтобы они зимой могли ни о чем не заботиться.

Вот и сейчас, заметив приближение зимы, он решил заняться заготовкой продуктов. Сначала он подсчитал, сколько всего у него есть продуктов, и на сколько банок хватит каждого. Копатыч выяснил интересный факт: каждого продукта хватало на чётное количество банок. Соответственно, после упаковки у него получилось $2k$ банок с продуктами.

Затем хозяйственный Копатыч решил все банки пронумеровать. Холодная зима, во время которой Копатыч спит, длится два месяца. Чтобы Смешарикам не пришлось самим делить продукты, Копатыч решил банкам на первый месяц дать отрицательные номера, а на второй — положительные. При этом он придерживался ещё двух правил. Первое правило: номера банок с одним и тем же продуктом должны иметь одинаковую абсолютную величину. И второе — количество банок с каждым продуктом на первый месяц и на второй должно быть одинаковым.

Однако, проснувшись сегодня утром, он заметил ужасный факт: за ночь одна из банок пропала. И теперь он хочет по номерам, написанным на оставшихся банках, определить номер пропавшей.

Формат входного файла

В первой строке входного файла дано нечётное целое число n ($1 \leq n < 100\,000$) — количество оставшихся у Копатыча банок. В следующей строке дано n целых чисел a_i ($1 \leq |a_i| \leq 1000$) — номера всех оставшихся банок.

Формат выходного файла

Выведите в выходной файл одно целое число — номер, который был написан на пропавшей банке.

Примеры

<code>reserve.in</code>	<code>reserve.out</code>
5 42 -1 -42 -5 5	1

Задача Н. Суммы

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

Крош любит праздники, потому что ему часто в честь этих праздников дарят подарки. И в этот день знаний ему посчастливилось получить несколько.

В гости к Крошу пришли Бараш и Лосяш, чтобы поздравить Кроша с этим событием. Они знают, что он любит считать. Особенно он любит складывать два числа.

Бараш и Лосяш сразу же вручили свои подарки Крошу. Когда Крош открыл подарки оказалось, что подарки почти одинаковы. Каждый подарок содержал кубики, на которых написаны все числа от a до b включительно, на каждом кубике было написано ровно одно число, и все кубики в каждом из подарков были различны. Отличались подарки только цветом кубиков: Бараш подарил синие кубики, а Лосяш — красные.

Когда гости разошлись, Крош решил поиграть в кубики. Он загадал два числа: c и d , и записал их на листочке, чтобы не забыть. Дальше он должен был взять один синий кубик и один красный так, чтобы сумма чисел, написанных на этих кубиках, была от c до d включительно. Тут его заинтересовал вопрос, сколькими способами можно это сделать.

Формат входного файла

В первой строке входного файла заданы четыре целых числа a , b , c и d . $0 \leq a \leq b \leq 10^9$, $0 \leq c \leq d \leq 2 \cdot 10^9$

Формат выходного файла

Выведите одно целое число: количество способов выбрать два кубика разных цветов так, чтобы сумма оказалась в заданном диапазоне.

Примеры

<code>sums.in</code>	<code>sums.out</code>
3 6 7 9	9
1 1 10 10	0

Задача I. Поезд

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

Пин опять что-то смастерил! На этот раз изобретением оказался поезд. Материалов у Пина было не так и много, и поэтому, в нём всего один вагон. Зато это настоящий плацкартный вагон на целых пятьдесят четыре места!

2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36
1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35
53 54		51 52		49 50		47 48		45 46		43 44		41 42		39 40		37 38	

Нижние полки имеют нечётные номера, а верхние — чётные.

Пин уже провёл некоторые его испытания. Дорога была неровная, поезд трясло, и некоторые места в вагоне развалились и пришли в непригодность. Но зато теперь Пин уверен в том, что остальные места крепкие и не развалятся!

Прознав про изобретение Пина, Крош и Ёжик сразу же захотели его испытать. Им немного страшно, и они хотят ехать вместе. Но на одно место помещается только один смешарик. Поэтому, они хотят выбрать два места, на которых ещё можно ехать. Они хотят выбрать два места, которые находятся в как можно более близких купе. Если таких вариантов есть несколько, они предпочитают не ехать на боковых местах, потому что они как-то подозрительно шатаются. Из всех таких вариантов они хотят выбрать тот, в котором как можно больше нижних мест, потому что поезд трясёт, и с верхних полок можно упасть. Помогите им выбрать два таких места.

Формат входного файла

В первой строке задано натуральное число n ($2 \leq n \leq 54$) — количество мест в вагоне, которые ещё не развалились. Далее, в следующей строке даны номера a_i ($1 \leq a_i \leq 54$) этих мест. Все a_i различны.

Формат выходного файла

Выведите два числа — номера свободных мест, на которых могут поехать Ёжик и Крош. Если ответов несколько, выведите любой.

Примеры

<code>train.in</code>	<code>train.out</code>
6 54 5 6 7 8 19	5 7