

Задача А. Календарь

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

В недалёком прошлом (а может быть, в недалёком будущем) Вася решил научиться играть в дартс. К сожалению, у него дома не было мишени, но ему на глаза попался старый откидной календарь. Вася открыл календарь на месяце своего рождения, повесил его на стену и принялся метать в него дротики. Естественно, каждым своим броском Вася протыкал все двенадцать листов календаря.

Каждый раз Вася пытался попасть в свой день рождения, и наконец ему это удалось. Довольный собой, он снял календарь со стены и, к своему удивлению, заметил, что удачный бросок проткнул заодно и день рождения его мамы.

Зная дату васиногo рождения и месяц, в котором родилась его мама, определите день её рождения.

Откидной календарь состоит из 12 страниц, описывающих месяца года. Каждая страница представляет собой табличку из пяти строк и семи столбцов. В клетках этой таблицы расположены числа месяца: в первом столбце записаны числа, которые приходятся на понедельник, во втором — на вторник, и так далее. Первое число месяца всегда расположено в первой строке, а затем числа идут подряд. Если же для нескольких последних чисел в таблице не хватает клеток (они должны располагаться на шестой строке), их дописывают к числам в пятой строке.

Формат входного файла

Входной файл состоит из нескольких наборов входных данных. Каждый такой набор состоит из одной строки, содержащей четыре числа Y , dV , mV и mM . Первое число Y ($2000 \leq Y \leq 2099$) обозначает год, на который был составлен васин календарь. Числа dV и mV обозначают день и месяц рождения Васи. Они представляют собой корректную дату года Y . Последнее число mM ($1 \leq mM \leq 12$) обозначает месяц, в котором родилась васина мама.

Число наборов данных во входном файле не превышает 10 000. Файл заканчивается строкой, содержащей четыре нуля.

Формат выходного файла

Для каждого набора входных данных выведите числа, в которых могла родиться васина мама, в порядке возрастания, или 0, если Вася ошибся, и дротик проткнул пустое поле. Следуйте формату примера максимально точно.

Пример

<code>calendar.in</code>	<code>calendar.out</code>
2009 22 11 1	Case #1:
2009 26 10 11	25
2009 30 08 01	Case #2:
0 0 0 0	23 30
	Case #3:
	0

Ноябрь 2009, второй тест из примера входных данных:

						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23/30	24	25	26	27	28	29

Задача В. Все ниже, и ниже, и ниже

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

Вася может удалить символ из строки w , если полученная строка лексикографически меньше, чем строка w . Например, из строки «abacaba» Вася может получить строки: «aacaba», «abaaba», «abacaa» и «abacab».

Пусть Вася может производить описанную выше операцию несколько раз. Сможет ли он получить строку t из строки s ?

Формат входного файла

Каждая строка входного файла содержит две непустые строки s и t , разделенные одним пробелом. Каждая строка состоит из латинских букв и десятичных цифр. Длина каждой строки (s или t) не превышает 100 000 символов. Файл содержит не более 1 000 000 пробельных символов.

Сравнение символов производится в соответствии с их кодами в таблице ASCII.

Формат выходного файла

Для каждого теста запишите ответ на задачу. Следуйте формату вывода в примере как можно точнее.

Пример

downhill.in	downhill.out
aaa a	Case #1: Yes, it is possible.
AB 0	Case #2: No, it is impossible.
ddbca b	Case #3: Yes, it is possible.
abacaba aaaa	Case #4: Yes, it is possible.
ababc aac	Case #5: No, it is impossible.

Задача С. Изабелла

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

Правила пасьянса Изабелла таковы: в колоде 52 карты, каждая из которых характеризуется мастью и значением. Мастей всего четыре: трефы, бубны, червы и пики. Они обозначаются буквами C, D, H и S, соответственно. Значений карт всего 13: 2, 3, 4, 5, 6, 7, 8, 9, 10 (T), валет (J), дама (Q), король (K) и туз (A). На игровом столе расположено тринадцать стопок карт. В начале игры в каждой стопке лежат три карты лицом вниз и одна карта лицом вверх. Разрешается убирать пары карт одинакового значения, лежащие лицом вверх. Каждый раз, когда убирается карта, лежащая под ней карта (если такая существует) переворачивается и становится доступной для игры.

Напишите программу, которая уберёт как можно больше карт.

Формат входного файла

Входной файл содержит не более 5 тестов. Каждый тест состоит из 13 строк, каждая из которых описывает одну стопку карт, начиная с нижней.

Формат выходного файла

Для каждого теста выведите как можно больше ходов. Если существует несколько решений с максимальным числом ходов, выведите любое из них.

Следуйте формату вывода, приведённому в примере, как можно точнее.

Примеры

<code>isabel.in</code>	
<code>AC AH 6S AT</code>	
<code>2C 2H 2S KT</code>	
<code>TC 3C 3H 3S</code>	
<code>4C 4H 4S 4T</code>	
<code>5C 5H 5S 5T</code>	
<code>6C 7C 6T 7S</code>	
<code>6H 8H QS 7T</code>	
<code>8C AS 8S 8T</code>	
<code>9C 9H 9S 9T</code>	
<code>TH TS QT TT</code>	
<code>JC JH JS JT</code>	
<code>QC QH 7H 3T</code>	
<code>KC KH KS 2T</code>	
<code>isabel.out</code>	
<code>Case #1: Vasya could remove 8 card(s).</code>	
<code>At step 1, Vasya should remove 3S and 3T.</code>	
<code>At step 2, Vasya should remove 7T and 7H.</code>	
<code>At step 3, Vasya should remove QS and QH.</code>	
<code>At step 4, Vasya should remove 8H and 8T.</code>	

Задача D. Рыцари и лжецы

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

Вася проводит отпуск на острове Рыцарей и Лжецов. Каждый из его жителей может быть либо рыцарем, либо лжецом. Рыцари всегда говорят только истину, а лжецы всегда лгут.

Васе рассказали странную историю про этот остров. Однажды n жителей этого острова собрались на праздник и расселись за круглым столом, после чего каждый указал на двух человек и сказал им: «Если я спрошу вас, являюсь я рыцарем или лжецом, ваши ответы совпадут!»

Вася понял, что такая ситуация могла произойти, если все её участники были бы рыцарями. Но ему тут же стало интересно, мог ли среди них быть хоть один лжец.

Формат входного файла

Входной файл содержит не более 1000 тестов. Первая строка каждого теста содержит одно целое число n ($1 \leq n \leq 1000$) — количество жителей острова, пришедших на праздник. Все они занумерованы числами от 1 до n . Далее следуют n строк. На i -й строке записаны два числа a_i и b_i — номера людей, на которых показал i -й человек ($1 \leq a_i, b_i \leq n$). Заметьте, что любые из чисел a_i и b_i могут совпадать.

Сумма n по всем тестам в одном входном файле не превосходит 1000.

Формат выходного файла

Для каждого теста выведите «Yes», если среди жителей мог оказаться лжец, и «No» в противном случае. Следуйте формату вывода, приведённому в примере, как можно точнее.

Примеры

<code>knights.in</code>	<code>knights.out</code>
3	Case #1: Yes
2 3	Case #2: No
1 3	
1 2	
2	
1 2	
1 2	

Задача E. Корни k -й степени

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

Вася занимается разработкой эффективного алгоритма нахождения корня k -й степени. Неудивительно, что в скором будущем ему потребуются наборы тестовых данных.

Васин коллега Петя уже подготовил массив целых чисел a . Однако, оказалось, что эти числа выбирались довольно случайно, а васин алгоритм устроен таким образом, что успешно находит корень, лишь если он является целым числом.

Вася решил создать новый массив из таких чисел, которые будут подходить его алгоритму. Для этого он планирует использовать все такие пары (a_i, a_j) , что $i < j$ и $a_i \cdot a_j = x^k$ для какого-то целого k .

Помогите Васе вычислить количество таких пар.

Формат входного файла

Входной файл содержит не более 10 000 тестов. Первая строка каждого теста содержит целые числа n и k ($1 \leq n, k \leq 100\,000$). Вторая строка содержит n чисел a_i ($1 \leq a_i \leq 1\,000\,000$).

Сумма n по всем тестам не превосходит 100 000.

Формат выходного файла

Для каждого теста запишите ответ на задачу. Следуйте формату вывода в примере как можно точнее.

Пример

<code>kroots.in</code>	<code>kroots.out</code>
4 2	Case #1: There are 3 pairs.
2 8 3 2	Case #2: There are 0 pairs.
3 5	Case #3: There are 1 pairs.
17 239 1	
2 2	
17 17	

Задача F. Марсианин Али-Баба

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

Марсианин Али-Баба очень любит искать клады. Однажды он нашел старинную карту сокровищ. После месяцев скитаний по Марсу он пришел к секретной двери, рядом с которой он обнаружил N отверстий, расположенных по кругу. На обратной стороне карты было сказано, что в каждом отверстии лежит марсианская селёдка, которая может быть повернута либо хвостом, либо головой наружу. У Али-Бабы M рук. Он может засовывать их в отверстия и переворачивать селёдки внутри них, как ему вздумается (он понимает на ощупь, как повернута та или иная селёдка). Проблема в том, что после такого действия круг с отверстиями начинает быстро вращаться, и после того, как он остановится, Али-Баба не может определить, куда он только что засовывал руки.

Али-Баба знает, что одно из отверстий является ключом к двери. А именно, что дверь откроется, если селёдка в этом отверстии повернута определённым образом. Но он не знает, что это за отверстие и как должна быть повернута селёдка в нем. Также Али-Бабе известно, что если селёдки во всех отверстиях, кроме ключа, будут повернуты в другую сторону, нежели селёдка в ключе, то дверь откроется всё равно.

Определите, сможет ли Али-Баба действовать так, чтобы открыть эту секретную дверь вне зависимости от начального положения селёдок.

Формат входного файла

Входной файл состоит из нескольких наборов входных данных. Каждый такой набор состоит из одной строки, содержащей два числа N и M ($1 \leq N, M \leq 15$) — количество отверстий с селёдками и рук у Али-Бабы соответственно.

Число наборов данных во входном файле не превышает 100.

Файл заканчивается строкой, содержащей два нуля.

Формат выходного файла

Для каждого набора входных данных выведите «Yes», если Али-Баба может открыть дверь, и «No», если не может. Следуйте формату примера максимально точно.

Примеры

<code>martian.in</code>	<code>martian.out</code>
3 1	Case #1:
3 2	No
0 0	Case #2:
	Yes

Задача G. Самолёт

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

В городе Энске идёт мокрый снег. Несмотря на это, самолёту необходимо отправиться по расписанию. Причина волнения экипажа в том, что снег налипает на крылья, меняя балансировку самолёта, поэтому необходимо что-то предпринять для подготовки самолёта к подъёму в воздух.

В аэропорте Энска довольно запутанная система взлётно-посадочных полос. Пока самолёт проходит те или иные из них, снег может налипнуть на одно из крыльев.

Задача экипажа сейчас состоит в том, чтобы выбрать путь для разгона, минимизирующий разность количеств снега, налипшего на крылья. Напишите программу, вычисляющую наилучший результат, которого возможно достичь.

Формат входного файла

Входной файл содержит не более 10 тестов. Каждый тест начинается строкой, содержащей два целых числа n и k — количество пересечений полос и самих отрезков взлётно-посадочных полос, соответственно ($2 \leq n \leq 10\,000$, $1 \leq k \leq 10\,000$). Далее следуют k строк, каждая из которых описывает часть взлётно-посадочной полосы. Описание состоит из трёх целых чисел a , b и w ($1 \leq a \neq b \leq n$, $-1 \leq w \leq 1$). Такое описание соответствует части полосы, соединяющей пересечения a и b , на которой снег липнет к левому крылу ($w = -1$), к правому крылу ($w = 1$), или где снег не налипает на крылья ($w = 0$).

Заметьте, что из-за экологических проблем снег имеет довольно сложную структуру (более того, учёные полагают, что он обладает мыслительными способностями...) Именно поэтому крыло, к которому липнет снег, не зависит от направления движения по полосе.

Самолёт начинает движение на пересечении номер 1 и должен взлететь в пересечении номер n . Путь самолёта должен проходить по полосам, переход с одной полосы на другую (возможно, ту же самую) осуществляется на пересечении, которое они обе соединяют с какими-то ещё.

Других ограничений на перемещение самолёта нет, способ взлететь существует во всех тестах.

Формат выходного файла

Для каждого теста выведите минимальную возможную разность. Следуйте формату, приведённому в примере, как можно точнее.

Пример

<code>plane.in</code>	
<code>2 1</code>	
<code>2 1 -1</code>	
<code>5 5</code>	
<code>1 2 1</code>	
<code>2 3 0</code>	
<code>2 4 0</code>	
<code>5 3 -1</code>	
<code>5 3 1</code>	
<code>plane.out</code>	
<code>Case #1: The plane needs to bear the difference of 1.</code>	
<code>Case #2: The plane needs to bear the difference of 0.</code>	

Задача N. Префикс-функция

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

На досуге Вася занимается изучением префикс-функции. Для начала опишем, что это такое: пусть дана строка s . Обозначим её префикс длины i как p_i , а суффикс длины i как s_i . Тогда префикс-функцией строки s называется такое максимальное число $\pi(s)$, меньшее её длины, что $p_{\pi(s)} = s_{\pi(s)}$.

Вася не слишком аккуратен, поэтому его компьютер часто заражается вирусами. Последний раз вирус испортил файл со строкой, префикс-функцию префиксов которой Вася изучал последнюю неделю. К счастью, некоторые значения префикс-функции сохранились.

Вася хочет восстановить строку s . Он сомневается, что удастся сделать это однозначно, поэтому для определённости его интересует лексикографически минимальная строка с таким набором известных префикс-функций префиксов.

Напишите программу, которая поможет Васе. В исходной строке s было использовано много разных символов, например, китайских иероглифов. Поэтому при восстановлении можете считать, что у вас в наличии есть столько символов, сколько понадобится, и все они пронумерованы натуральными числами, начиная с единицы.

Формат входного файла

Входной файл содержит несколько тестов. Каждый из них начинается строкой, содержащей единственное целое число n — длину строки. Вторая строка теста содержит n целых чисел — значения $\pi(p_1), \pi(p_2), \dots, \pi(p_n)$ ($-1 \leq \pi(p_i) \leq n$). Те значения, которые не сохранились, обозначены -1 .

Файл завершается строкой с одним нулём, который не является тестом и который не нужно обрабатывать. Сумма n по всем тестам в одном входном файле не превысит 2000.

Формат выходного файла

Для каждого теста выведите лексикографически минимальную строку, подходящую под соответствующий набор значений префикс-функции, либо сообщите, что такой строки не существует.

Следуйте формату вывода, приведённому в примере, как можно точнее.

Примеры

<code>prefix.in</code>
7
0 0 1 0 1 2 3
4
-1 -1 -1 3
2
1 0
0
<code>prefix.out</code>
Case #1: The minimal string is 1 2 1 3 1 2 1
Case #2: The minimal string is 1 1 1 1
Case #3: There is no such string

Задача I. Ещё одна игра с камнями

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

Вася и Петя играют в неинтересную игру с самыми обычными правилами: на столе лежит сороконожка, у которой есть N ножек, а также камень. Первый игрок берёт со стола камень и отдавливает сороконожке одну ножку. Каждый следующий ход производится так: если на последнем ходу игрок отдал k ножек, то его соперник может отдалить k или $k + 1$ ножку. Отдавленную один раз ножку нельзя отдалить во второй раз. Проигрывает тот, кто не может ходить, то есть сороконожка. Однако нас интересует, выиграет первый игрок, или второй.

Формат входного файла

Входной файл содержит несколько тестов. Каждый тест представляет собой одну строку, на которой записано единственное целое число N ($1 \leq N \leq 100\,000$). Набор тестов завершается строкой с числом 0, которое не нужно обрабатывать.

Формат выходного файла

Для каждого теста выведите, кто выигрывает при правильной игре. Следуйте формату вывода, приведённому в примере, как можно точнее.

Пример

<code>stonegame.in</code>	<code>stonegame.out</code>
1	Case #1: First player wins.
2	Case #2: Second player wins.
3	Case #3: Second player wins.
0	

Задача J. Три окружности

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

На плоскости отмечены три различные точки. Вам необходимо построить окружности с центрами в этих точках так, чтобы все эти окружности попарно касались, или определить, что это сделать невозможно.

Формат входного файла

Входной файл состоит из нескольких наборов входных данных. Каждый такой набор состоит из одной строки, содержащей шесть целых чисел $x_1, y_1, x_2, y_2, x_3, y_3$ — координаты заданных точек. В каждом наборе все точки различны между собой. Все числа не превосходят 10^6 по модулю.

Число наборов данных во входном файле не превышает 10 000.

Файл заканчивается строкой, содержащей шесть нулей.

Формат выходного файла

Для каждого набора входных данных выведите три вещественных числа — радиусы окружностей с центрами в соответствующих точках, или три числа -1 , если такие окружности построить нельзя. Требуется найти ответ, в котором все радиусы будут не меньше 0.1. Гарантируется, что во всех тестах, где ответ существует, будет существовать также ответ, удовлетворяющий этому условию.

Если ответов несколько, выведите любой из них. Выводите ответ с как можно большей точностью.

Следуйте формату вывода, приведённому в примере, максимально точно.

Пример

<code>three.in</code>	<code>three.out</code>
0 0 2 0 0 2 0 0 0 0 0 0	Case #1: 0.5857864 1.4142136 1.4142136

Задача К. Восстановление триангуляции

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

Триангуляцией правильного многоугольника называется разбиение его на треугольники, вершины которых совпадают с вершинами многоугольника. Легко понять, что для правильного N -угольника такое разбиение задается $N - 3$ попарно не пересекающимися диагоналями, которые в нём проведены. Пронумеруем все вершины этого многоугольника числами от 1 до N в порядке обхода по часовой стрелке. Тогда каждая диагональ, в свою очередь, будет задаваться двумя числами — номерами вершин, которые она соединяет.

Пай-девочка Аня нарисовала карандашом на листе бумаги правильный N -угольник и триангулировала его. На другом листе она выписала все диагонали, которые составили эту триангуляцию, а потом пошла на кухню пить чай. Затем в комнату пришел хулиган Андрей и решил сделать пакость. Он разорвал лист, на котором был нарисован многоугольник, и стёр с другого листа из записи каждой диагонали максимальный из двух номеров вершин.

Когда Аня увидела, что натворил Андрей, она очень расстроилась. Помогите ей восстановить исходную триангуляцию!

Формат входного файла

Входной файл состоит из нескольких наборов входных данных. Каждый такой набор состоит из двух строк. В первой строке содержится единственное число N ($N \geq 4$) — количество вершин правильного многоугольника, который Аня триангулировала. Во второй строке содержатся $N - 3$ числа, оставшиеся на втором листе. Гарантируется, что все входные данные корректны. Входной файл будет заканчиваться строкой, содержащей 0.

Общее количество чисел во всех наборах не превышает 100 000.

Формат выходного файла

Для каждого набора входных данных выведите диагонали, составляющие анину триангуляцию. Каждая диагональ должна выводиться на отдельной строке, причём первым должен выводиться наименьший из номеров вершин, составляющих эту диагональ. Диагонали должны быть выведены в лексикографическом порядке. Если таких триангуляций несколько, выведите любую из них. Следуйте формату примера максимально точно.

Примеры

<code>triangulation.in</code>	<code>triangulation.out</code>
4	Case #1:
1	1 3
5	Case #2:
1 3	1 3
0	3 5