

Задача 1. Цапли

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

Петя и Маша пришли в зоопарк. Больше всего Пете понравились цапли. Он был поражен их способностью спать на одной ноге.

В вольере находятся несколько цапель. Некоторые из них стоят на двух ногах, некоторые — на одной. Когда цапля стоит на одной ноге, то другую ее ногу не видно. Петя пересчитал видимые ноги всех цапель, и у него получилось число a .

Через несколько минут к вольеру подошла Маша. За это время некоторые цапли могли поменять позу, поэтому Петя предложил ей заново пересчитать видимые ноги цапель. Когда Маша это сделала, у нее получилось число b .

Выйдя из зоопарка, Петя с Машей заинтересовались, сколько же всего цапель было в вольере. Вскоре ребята поняли, что однозначно определить это число можно не всегда. Теперь они хотят понять, какое минимальное и какое максимальное количество цапель могло быть в вольере.

Требуется написать программу, которая по заданным числам a и b выведет минимальное и максимальное количество цапель, которое могло быть в вольере.

Формат входного файла

Входной файл содержит два целых числа a и b , разделенных ровно одним пробелом ($1 \leq a \leq 10^9$, $1 \leq b \leq 10^9$).

Формат выходного файла

Выведите в выходной файл два целых числа, разделенных пробелом — минимальное и максимальное число цапель, которое могло быть в вольере. Гарантируется, что хотя бы одно количество цапель соответствует условию задачи.

Пример входного и выходного файлов

herons.in	herons.out
3 4	2 3

Пояснения к примеру

В приведенном примере возможны следующие варианты:

1) В вольере две цапли. Когда Петя считал ноги, одна цапля стояла на двух ногах, а другая — на одной. Петя насчитал три ноги. Когда Маша считала ноги, обе цапли стояли на двух ногах, Маша насчитала четыре ноги.

2) В вольере три цапли. Когда Петя считал ноги, все цапли стояли на одной ноге, Петя насчитал три ноги. Когда Маша считала ноги, одна цапля стояла на двух ногах, а еще две — на одной. Маша насчитала четыре ноги.

Система оценивания

Правильные решения для тестов, в которых $1 \leq a \leq 1000$, $1 \leq b \leq 1000$, будут оцениваться из 50 баллов.

Задача 2. Круглый стол

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

Возрождая древние традиции английских рыцарей, в одном городе члены школьного клуба любителей информатики каждую неделю собираются за круглым столом и обсуждают результаты последних соревнований.

Руководитель клуба Иван Петрович недавно заметил, что не все ребята активно участвуют в обсуждении. Понаблюдав за несколькими заседаниями клуба, он заметил, что активность члена клуба зависит от того, кто с кем сидит рядом.

В клуб приходят на занятия m мальчиков и n девочек. Иван Петрович заметил, что мальчик активно участвует в обсуждении только тогда, когда непосредственно рядом с ним с обеих сторон от него сидят девочки, а девочка активно участвует в обсуждении только тогда, когда непосредственно рядом с ней с одной стороны от нее сидит мальчик, а с другой — девочка.

Желая сделать заседание клуба как можно более интересным, Иван Петрович решил разместить участников за круглым столом таким образом, чтобы как можно больше членов клуба приняло активное участие в обсуждении.

Требуется написать программу, которая по заданным числам m и n выведет такой способ размещения m мальчиков и n девочек за круглым столом, при котором максимальное количество членов клуба будет активно участвовать в обсуждении.

Формат входного файла

Входной файл содержит два целых числа m и n , разделенных ровно одним пробелом ($0 \leq m \leq 1000$, $0 \leq n \leq 1000$, $m + n \geq 3$).

Формат выходного файла

Выходной файл должен содержать строку с расположенными в некотором порядке m символами «B» (заглавная латинская буква) и n символами «G» (заглавная латинская буква). Символ «B» означает мальчика, а символ «G» — девочку.

Символы следует расположить в том порядке, в котором нужно разместить членов клуба вокруг стола. Соседние символы соответствуют членам клуба, которые сидят рядом. Рядом сидят также члены клуба, соответствующие первому и последнему символу выведенной строки.

Примеры входных и выходных файлов

table.in	table.out
1 2	BGG
2 2	BGBG

Пояснения к примерам

В первом примере все члены клуба примут активное участие в обсуждении.

Во втором примере мальчики примут активное участие в обсуждении, а девочки нет. В этом примере можно также разместить членов клуба следующим образом: «BVBGG». В этом случае активное участие в обсуждении примут обе девочки, а мальчики — нет. Разместить всех так, чтобы три или четыре члена клуба приняли активное участие в обсуждении, нельзя.

Задача 3. Поврежденный XML

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

Формат XML является распространенным способом обмена данными между различными программами. Недавно программист Иванов написал небольшую программу, которая сохраняет некоторую важную информацию в виде XML-строки.

XML-строка состоит из открывающих и закрывающих тегов.

Открывающий тег начинается с открывающей угловой скобки (<), за ней следует имя тега — непустая строка из строчных букв латинского алфавита, а затем закрывающая угловая скобка (>). Примеры открывающих тегов: <a>, <dog>.

Закрывающий тег начинается с открывающей угловой скобки, за ней следует прямой слеш (/), затем имя тега — непустая строка из строчных букв латинского алфавита, а затем закрывающая угловая скобка. Примеры закрывающихся тегов: , </dog>.

XML-строка называется *корректной*, если она может быть получена по следующим правилам:

- Пустая строка является корректной XML-строкой.
- Если A и B — корректные XML-строки, то строка AB, получающаяся приписыванием строки B в конец строки A, также является корректной XML-строкой.
- Если A — корректная XML-строка, то строка <X>A</X>, получающаяся приписыванием в начало A открывающегося тега, а в конец — закрывающегося с таким же именем, также является корректной XML-строкой. Здесь X — любая непустая строка из строчных букв латинского алфавита.

Например, представленные ниже строки:

```
<a></a>  
<a><ab></ab><c></c></a>  
<a></a><a></a><a></a>
```

являются корректными XML-строками, а такие строки как:

```
<a></b>  
<a><b>  
<a><b></a></b>
```

не являются корректными XML-строками.

Иванов отправил файл с сохраненной XML-строкой по электронной почте своему коллеге Петрову. Однако, к сожалению, файл повредился в процессе пересылки: ровно один символ в строке заменился на некоторый другой символ.

Требуется написать программу, которая по строке, которую получил Петров, восстановит исходную XML-строку, которую отправлял Иванов.

Формат входного файла

Входной файл содержит одну строку, которая заменой ровно одного символа может быть превращена в корректную XML-строку. Длина строки лежит в пределах от 7 до 1000, включительно. Строка содержит только строчные буквы латинского алфавита и символы «<» (ASCII код 60), «>» (ASCII код 62) и «/» (ASCII код 47).

Строка во входном файле заканчивается переводом строки.

Формат выходного файла

Выходной файл должен содержать корректную XML-строку, которая может быть получена из строки во входном файле заменой ровно одного символа на другой. Если вариантов ответа несколько, можно вывести любой.

Примеры входных и выходных файлов

xml.in	xml.out
<a>	<a>
<a><aa>	<a>
<a><>a>	<a>
<a/	<a>

Система оценивания

Правильные решения для тестов, в которых одна буква заменена на другую букву, оцениваются из 50 баллов.

Задача 4. Игра с числами

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

Сегодня на уроке математики Петя и Вася изучали понятие арифметической прогрессии. Арифметической прогрессией с разностью d называется последовательность чисел a_1, a_2, \dots, a_k , в которой разность между любыми двумя последовательными числами равна d . Например, последовательность 2, 5, 8, 11 является арифметической прогрессией с разностью 3.

После урока Петя и Вася придумали новую игру с числами. Игра проходит следующим образом.

В корзине находятся n фишек, на которых написаны различные целые числа a_1, a_2, \dots, a_n . По ходу игры игроки выкладывают фишки из корзины на стол. Петя и Вася делают ходы по очереди, первым ходит Петя. Ход состоит в том, что игрок берет одну фишку из корзины и выкладывает ее на стол. Игрок может сам решить, какую фишку взять. После этого он должен назвать целое число $d \geq 2$ такое, что все числа на выбранных к данному моменту фишках являются членами некоторой арифметической прогрессии с разностью d , не обязательно последовательными. Например, если на столе выложены фишки с числами 2, 8 и 11, то можно назвать число 3, поскольку эти числа являются членами приведенной в начале условия арифметической прогрессии с разностью 3.

Игрок проигрывает, если он не может сделать ход из-за отсутствия фишек в корзине или из-за того, что добавление любой фишки из корзины на стол приводит к тому, что он не сможет подобрать соответствующее число d .

Например, если в корзине имеются числа 2, 3, 5 и 7, то Петя может выиграть. Для этого ему необходимо первым ходом выложить на стол число 3. После первого хода у него много вариантов назвать число d , например он может назвать $d = 3$. Теперь у Васи два варианта хода.

- 1) Вася может вторым ходом выложить фишку с числом 5 и назвать $d = 2$. Тогда Петя выкладывает фишку с числом 7, называя $d = 2$. На столе оказываются фишки с числами 3, 5 и 7, а в корзине осталась только фишка с числом 2. Вася не может ее выложить, поскольку после этого он не сможет назвать корректное число d . В этом случае Вася проигрывает.
- 2) Вася может вторым ходом выложить фишку с числом 7 и также назвать, например, $d = 2$. Тогда Петя выкладывает фишку с числом 5, называя также $d = 2$. Вася снова попадает в ситуацию, когда на столе оказываются фишки с числами 3, 5 и 7, а в корзине осталась только фишка с числом 2, и он также проигрывает.

Заметим, что любой другой первый ход Пети приводит к его проигрышу. Если он выкладывает число 2, то Вася отвечает числом 7, и Петя не может выложить ни одной фишки. Если Петя выкладывает фишку с числом 5 или 7, то Вася выкладывает фишку с числом 2, и у Пети также нет допустимого хода.

Требуется написать программу, которая по заданному количеству фишек n и числам на фишках a_1, a_2, \dots, a_n определяет, сможет ли Петя выиграть вне зависимости от действий Васи, и находит все возможные первые ходы Пети, ведущие к выигрышу.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \leq n \leq 200$).

Вторая строка содержит n различных целых чисел a_1, a_2, \dots, a_n (для всех i от 1 до n выполняется неравенство $1 \leq a_i \leq 10^5$). Соседние числа разделены ровно одним пробелом.

Формат выходного файла

Первая строка выходного файла должна содержать число k — количество различных первых ходов, которые может сделать Петя, чтобы выиграть. Если Вася может выиграть вне зависимости от действий Пети, строка должна содержать цифру 0.

Во второй строке должно содержаться k различных целых чисел — все выигрышные числа. Будем называть число выигрышным, если, выложив в качестве первого хода фишку, содержащую это число, Петя может выиграть вне зависимости от действий Васи. Соседние числа в строке должны быть разделены ровно одним пробелом.

Примеры входных и выходных файлов

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

Пояснения к примерам

Первый пример рассматривается в тексте условия этой задачи.

Во втором примере, какую бы фишку не выложил Петя первым ходом, Вася в ответ выкладывает другую фишку, и Петя не может сделать ход из-за отсутствия фишек в корзине.

Система оценивания

Правильные решения для тестов, в которых $1 \leq n \leq 15$, будут оцениваться из 40 баллов.