

Problem A: Морской бой Маши и Медведя

Prob id: 1
Full score: 100
Score bonus: 10
Input file name: input.txt or standard input
Output file name: output.txt or standard output
Time limit: 1 s
Real time limit: 2 s
Memory limit: 64M

Морской бой Маши и Медведя

Маша и Медведь играют в морской бой. Это игра на клетчатом листке, в которой нужно первым поразить корабли противника. Сейчас ход Маши. Она играла так, что на карте Медведя остались непроверенными 9 клеток, образующих квадрат 3×3 и находящиеся в левом верхнем углу. В этих клетках остался последний однопалубный корабль Медведя. (см. рисунок для понимания ситуации). Осталось нанести последний удар. Все соседние с этими девятью клетки уже были Машей проверены и оказались пустыми.

Вам известно расположение последнего корабля Медведя. Помогите Маше -- напишите программу, которая подскажет ей правильный ход. Строки в игре обозначены большими латинскими буквами A, B и C, столбцы числами 1, 2 и 3.

Input format

В трех строках расположено по три числа через пробел. Среди этих чисел восемь нулей и одна единица. Единица соответствует положению однопалубного корабля Медведя, нули -- пустым клеткам.

Output format

Вывести победный ход Маши. Он должен состоять из двух символов без пробелов: сначала идет строка, обозначенная одним из латинских символов A, B или C, потом идет столбец, обозначенный цифрой 1, 2 или 3.

Examples

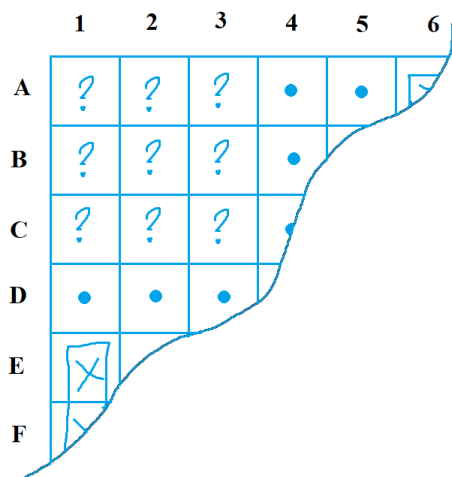
Input

```
0 0 0
0 0 1
0 0 0
```

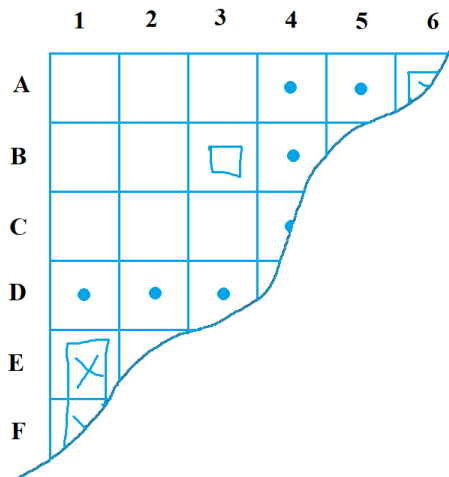
Output

```
B3
```

Notes



Так Маша видит расположение кораблей Медведя



Так Медведь видит расположение своих кораблей

На рисунке показано, как Медведь видит расположение своих кораблей для примера и как его видит Маша. Вопросами обозначены те клетки, о которых Маша еще не знает что в них расположено - корабль или пустая клетка. В других примерах возможна другое расположение однопалубного корабля, но вопросы стоят на тех же клетках.

Проверяемая программа тестируется на 9 тестах. За прохождение каждого она получает 10 баллов. При полном прохождении всех тестов, программа получает дополнительно еще 10 баллов.

Problem B: Тролль-переливашки

Prob Id: 2

Full score: 100

Score bonus:

Input file name: input.txt or standard input

Output file name: output.txt or standard output

Time limit: 1 s

Real time limit: 2 s

Memory limit: 64M

Тролль-переливашки

Тролль Зумбагеш начался научно-популярной литературы и создал вечный двигатель. В его исполнении он состоит из четырех емкостей. Емкости имеют объемы в **A**, **B**, **C** и **D** литров и расположены по кругу в этом порядке.

Для начала, Зумбагеш наливает доверху специальную жидкость в емкость **A** и запускает процесс. Далее бесконечно долго выполняется последовательность переливаний. На первом шаге переливание происходит из **A** в **B**, на втором шаге из **B** в **C**, на третьем из **C** в **D**, на четвертом из **D** в **A** и так далее по кругу. Переливание из емкости **X** в емкость **Y** происходит либо до тех пор, пока не закончится жидкость в емкости **X**, либо не заполнится целиком емкость **Y**, после чего начинается следующий такт переливаний из емкости **Y** в следующую по кругу емкость.

В данной задаче требуется определить, какой объем жидкости будет перелит из емкости в емкость на миллион первом шаге.

Input format

В единственной строке через пробел указаны четыре числа **A, B, C, D**. Все эти числа целые в пределах от 1 до 100.

Output format

Вывести одно число -- количество перелитых литров на шаге с номером **1000001**.

Examples

Input

4 2 3 1

Output

1

Notes

Проверка этой задачи проходит на 20 тестах. За прохождение каждого теста попытка получает 5 баллов.

Problem C: Винни-Пух восстанавливает справедливость

Prob id: 3

Full score: 100

Score bonus:

Input file name: input.txt or standard input

Output file name: output.txt or standard output

Time limit: 1 s

Real time limit: 1 s

Memory limit: 64M

Винни-пух восстанавливает справедливость

Кролик пригласил в гости Винни-пуха и Пятачка на чай. К чаю для своих гостей он припас две длинные конфеты. Винни-пух заметил, что длина одной конфеты **a**, а длина другой конфеты **b** сантиметров, причем эти длины не равны.

Он подумал: "Будет несправедливо, если Пятачку достанется конфета меньше чем моя". И он решил восстановить справедливость. Он действовал следующим образом: до тех пор, пока длины конфет различны, он брал большую конфету и откусывал от неё кусок, равный длине меньшей конфеты. В конце концов конфеты все же сравнялись и довольный Винни-пух отдал одну из них Пятачку.

А пораженный справедливостью друга Кролик задумался: "А сколько раз Винни-пух откусил?"

Input format

На вход подаются два целых числа **a** и **b** через пробел -- исходные длины конфет. $1 \leq a, b \leq 10^{18}$, **a** и **b** не равны между собой.

Output format

Вывести одно число -- количество раз, которые придется Винни-пуху откусить от конфет, чтобы восстановить справедливость. Так как входные данные и ответ могут быть очень большими, используйте 64-битные типы данных в языках C++, Java и Pascal.

Examples

Input

17 7

Output

6

Notes

В примере на первом шаге Винни-пух откусит 7 сантиметров и получит конфеты 10 и 7.

На втором шаге он откусит еще 7 сантиметров и получит конфеты 3 и 7.

На третьем шаге он откусит 3 сантиметра и получит конфеты 3 и 4.

На четвертом шаге он откусит еще 3 сантиметра и получит 3 и 1.

На пятом шаге откусит 1 сантиметр и получит 2 и 1.

На шестом шаге откусит 1 сантиметр и получит 1 и 1.

Конфеты выровнялись и процесс закончен.

Подзадачи

Подзадача	Номера тестов	Баллы	Ограничения	Необходимые подзадачи
0 (тесты из условия)	1	0		
1	2-15	30	$1 \leq a, b \leq 1000$	0
2	16-27	70	без дополнительных ограничений	0, 1

Проверка этой задачи проводится по группам тестов. Это значит, что решение должно выдать правильный ответ для каждого теста группы — только в этом случае все баллы за эту подзадачу будут засчитаны проверяемому решению. Если хотя бы на один тест из группы будет дан неверный ответ, баллы, начисляемые проверяемой программе за эту подзадачу, равны 0. Помимо этого, баллы за подзадачу начисляются только в том случае, когда пройдены тесты всех необходимых для нее подзадач.

Problem D: На вкус и цвет все фломастеры разные

Prob Id: 4

Full score: 100

Input file name: input.txt or standard input

Output file name: output.txt or standard output

Time limit: 1 s

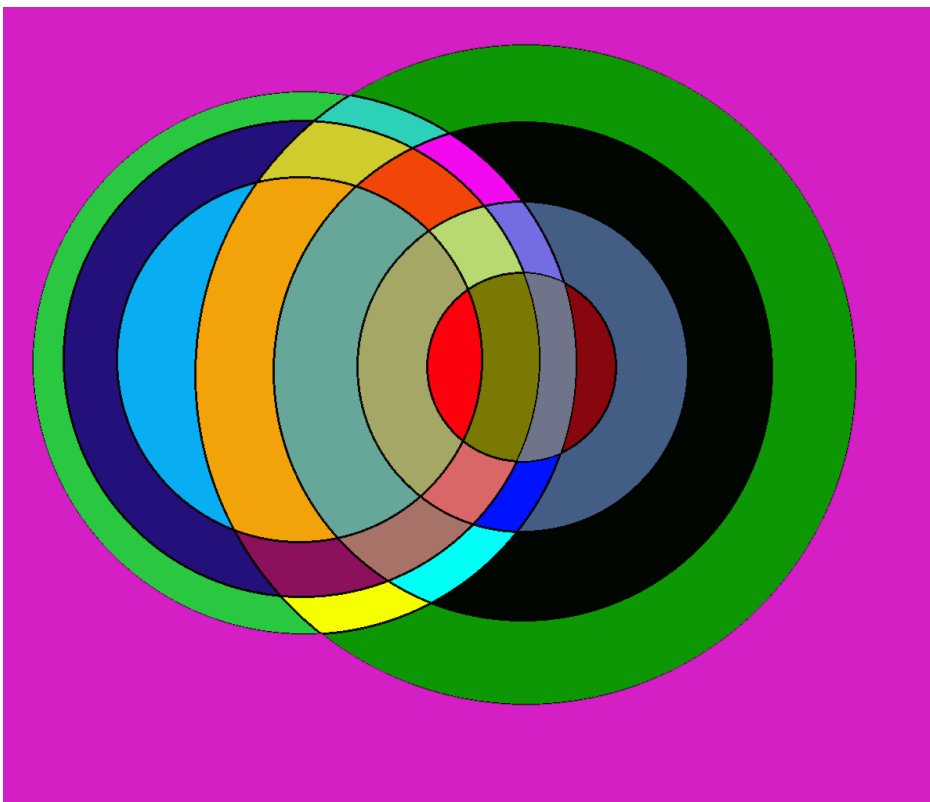
Real time limit: 2 s

Memory limit: 64M

На вкус и цвет все фломастеры разные

Троль Зумбагеш где-то раздобыл циркуль и большую-пребольшую упаковку с разноцветными фломастерами. Недолго думая, он нарисовал n окружностей с одним и тем же центром и разными радиусами (он не знает, что такие окружности называются концентрическими, но мы-то с вами знаем об этом). Этого ему показалось мало и он, выбрав другую точку и используя её как центр, нарисовал еще m других концентрических окружностей с разными радиусами. Получилось так, что каждая окружность из первой группы пересекалась с каждой окружностью из второй в двух точках.

Получившийся результат ему страшно понравился, но для завершения шедевра чего-то не хватало. И он понял - нужно добавить произведению цвет. Тогда он взял фломастеры и раскрасил каждую полученную пересечением окружностей область в свой особый цвет (в том числе и фон).



Поняв, что получившееся произведение ничем не хуже известных работ авангардистов, Зумбагеш повесил картину у себя в комнате и долгими зимними вечерами умиротворенно разглядывал её. Но одна мысль не давала ему покоя: он забыл, сколько цветов использовал для создания шедевра. Ваша задача -- подсказать ему ответ.

Input format

На вход подаются два числа n и m через пробел.

$0 \leq n, m \leq 10^9$.

Output format

Вывести одно число -- количество областей, на которые система из двух концентрических окружностей делит плоскость. Так как ответ может быть очень большим, используйте 64-битные типы данных в языках C++, Java, Pascal.

Examples

Input

3 4

Output

26

Notes

Подзадачи

Подзадача	Номера тестов	Баллы	Ограничения	Необходимые подзадачи
0 (тесты из условия)	1	0		
1	2-9	20	$\min(n, m) = 0$	
2	10-20	40	$1 \leq n, m \leq 999$	
3	21-35	40	$10^3 \leq n, m \leq 10^9$	

Проверка этой задачи проводится по группам тестов. Это значит, что решение должно выдать правильный ответ для каждого теста группы — только в этом случае все баллы за эту подзадачу будут засчитаны проверяемому решению. Если хотя бы на один тест из группы будет дан неверный ответ, баллы, начисляемые проверяемой программе за эту подзадачу, равны 0. Помимо этого, баллы за подзадачу начисляются только в том случае, когда пройдены тесты всех необходимых для нее подзадач.

Problem E: Лесозаготовки (простая версия)

Prob Id: 5

Full score: 100

Score bonus:

Input file name: `input.txt` or standard input

Output file name: `output.txt` or standard output

Time limit: 1 s

Real time limit: 2 s

Memory limit: 64M

Лесозаготовка (простая версия)

Некоторый игровой мир расположен на бесконечной прямоугольной клетчатой сетке. В одном из её квадратиков находится город, все остальные изначально заполнены лесом.

Обитатели города нуждаются в дереве для постройки зданий. Для того, чтобы заготавливать лес, они для начала построили робота "Лесоруб-1". Запаса его аккумулятора хватает, чтобы переместиться в любую клетку квадрата $A \times A$ с центром в клетке города и стороной A , заготовить на достигнутой клетке весь лес и транспортировать его обратно в город. После этого, на данной клетке лес отсутствует до конца игры. В городе робот заряжает аккумулятор и снова идет добывать лес на какой-то другой клетке. Обратите внимание, что для числа A выполняются следующие свойства: $A > 1$, A -- целое нечетное

После некоторого времени, робот "Лесоруб-1" вырубил лес во всех клетках, до которых он мог дойти и вернуться с лесом, и был демонтирован. Для продолжения лесозаготовок, жители города построили более продвинутого робота "Лесоруб-2", который мог добывать лес в любой клетке квадрата $B \times B$ с центром в клетке города и стороной B , причем $B > A$, B -- целое нечетное. Этот робот снова через некоторое время вырубил лес во всех клетках, до которых мог добраться.

Известно, что "Лесоруб-2" по ходу своей деятельности заготовил лес в S клетках. По числу S найти как можно больше вариантов чисел A и B , которые могли бы привести к такому показателю "Лесоруба-2".

Input format

На вход подается единственное число S -- количество клеток, в которых робот "Лесоруб-2" произвел заготовку леса. Число S -- кратно четырем и находится в пределах от 20 до 10^{12} . Гарантируется, что имеется как минимум один способ подобрать два числа A и B , таких, что $B > A > 1$, оба числа **нечетные**, и робот "Лесоруб-2" с мощностью B , работая после робота "Лесоруб-1" с мощностью A заготовит лес ровно в S клетках.

Output format

В первой строке вывести количество n вариантов ответа, найденных вашим решением. В следующих n строках вывести по два числа A и B через пробел, таких, что $B > A > 1$, оба числа **нечетные** и они дадут при подстановке в условия задачи в ответе число S . Первое число в строке должно быть числом A , второе числом B . Варианты пар чисел A и B можно выводить в любом порядке.

Examples

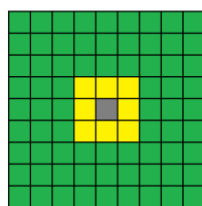
Input

72

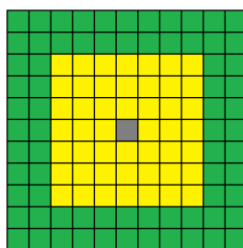
Output

3
17 19
3 9
7 11

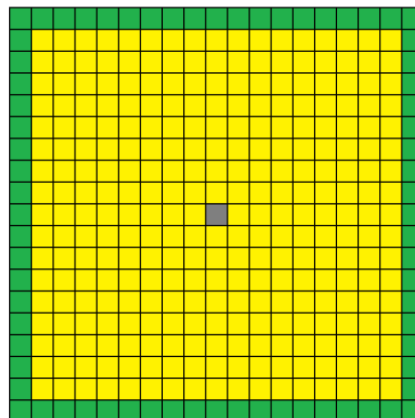
Notes



$A = 3, B = 9, S = 72$



$A = 7, B = 11, S = 72$



$A = 17, B = 19, S = 72$

На рисунке можно видеть иллюстрацию к примеру из условия. Серым обозначена клетка с городом, желтым -- клетки, на которых заготавливал лес "Лесоруб-1", на данный момент пустые, зеленым -- клетки, на которых будет заготавливать лес "Лесоруб-2". Видно, что во всех трех случаях "Лесоруб-2" заготовит лес из 72 клеток. Можно показать, что других вариантов решения для $S = 72$ нет.

Система оценки: за каждый правильный вариант ответа внутри одного теста, ваша попытка получит 1 балл. Таким образом, например, за тест, в котором имеется три правильных варианта ответа, ваша попытка получит три балла. Тесты сгенерированы таким образом, что за полное решение всех тестов суммарно можно набрать ровно 100 баллов. Каждый тест оценивается независимо от других. **Пример из условия не встречается среди основных тестов для этой задачи.**

Если ваша попытка выводит количество ответов, большее чем должно быть в полном наборе ответов на этот тест, она получит 0 баллов. Повторяющиеся варианты ответов (при условии, что их суммарное количество не превосходит полного ответа) учитываются по одному разу.

Участник может видеть по данной задаче только полученные баллы. Протокол проверки не предоставляется.

Problem F: Договор об услуге сортировки

Prob id: 6
Full score: 100
Input file name: `input.txt` or standard input
Output file name: `output.txt` or standard output
Time limit: 1 s
Real time limit: 2 s
Memory limit: 64M

Договор об услуге сортировки

Заказчик хочет чтобы для его набора чисел a_1, a_2, \dots, a_n произвели услугу сортировки по неубыванию.

В силу ряда особенностей, способ сортировки устанавливается следующий: исполнитель должен разбить сортируемый набор по своему усмотрению на любое число непустых **непересекающихся** отрезков и отсортировать каждый из них по неубыванию. В итоге этих действий весь исходный набор так же должен быть отсортирован по неубыванию. Менять порядок выбранных отрезков запрещено. В частности, исполнитель может разбить весь набор на один отрезок из всех чисел, либо на n отрезков по одному числу.

За каждую отдельную операцию сортировки непустого отрезка заказчик выплачивает исполнителю гонорар в размере 1 денежной единицы при условии, что данная сортировка **изменяет** исходный вид сортируемого отрезка. Если же отрезок при сортировке не изменяется, оплата за него не производится.

По предоставленному заказчиком набору определить максимальный возможный гонорар исполнителя.

Input format

В первой строке находится одно число n -- размер набора заказчика. $1 \leq n \leq 2 \cdot 10^5$

Во второй строке через пробел указываются числа a_1, a_2, \dots, a_n -- предоставленный заказчиком набор чисел. $1 \leq a_i \leq 10^{18}$

Output format

Вывести наибольший возможный гонорар, который сможет получить исполнитель за сортировку этой перестановки, при условии, что исполнитель сам определяет границы разбиения предоставленного набора на отрезки.

Examples

Input

```
4
2 1 4 3
```

Output

```
2
```


Input

4
1 2 2 4

Output

0

Input

5
4 1 3 3 2

Output

1

Input

11
3 4 3 2 4 7 8 7 12 12 9

Output

3

Notes

В первом примере нужно разбить на два отрезка: с первого элемента по второй и с третьего элемента по четвертый.

Во втором примере весь набор уже отсортирован и исполнитель не сможет уже ничего дополнительно отсортировать.

В третьем примере единственный способ разбиения -- это взять весь набор и отсортировать его.

В четвертом примере можно разбить например так: с первого элемента по четвертый, далее с пятого элемента по шестой, потом с седьмого элемента по восьмой и наконец с девятого элемента по одиннадцатый. За сортировку первого, третьего и четвертого отрезков исполнитель получит по одной единице, а второй уже отсортирован, поэтому за его сортировку платить не нужно. Другой вариант - вместо двух первых отрезков сразу взять один отрезок с первого элемента по шестой и отсортировать его, а далее так же как и в первом варианте. Есть и другие способы разбиения с гонораром в три единицы. Больше получить на сортировке набора из четвертого примера не получится.

Подзадачи

Подзадача	Номера тестов	Баллы	Ограничения	Необходимые подзадачи
0 (тесты из условия)	1-4	0		
1	5-29	50	$1 \leq n \leq 1000$	0
2	30-38	50	без дополнительных ограничений	0, 1

Проверка этой задачи проводится по группам тестов. Это значит, что решение должно выдать правильный ответ для каждого теста группы — только в этом случае все баллы за эту подзадачу будут засчитаны проверяемому решению. Если хотя бы на один тест из группы будет дан неверный ответ, баллы, начисляемые проверяемой программе за эту подзадачу, равны 0. Помимо этого, баллы за подзадачу начисляются только в том случае, когда пройдены тесты всех необходимых для нее подзадач.