

# Problem A: Тролль-переливашки

Prob Id: 1

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 начитался 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 B: На вкус и цвет все фломастеры разные

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

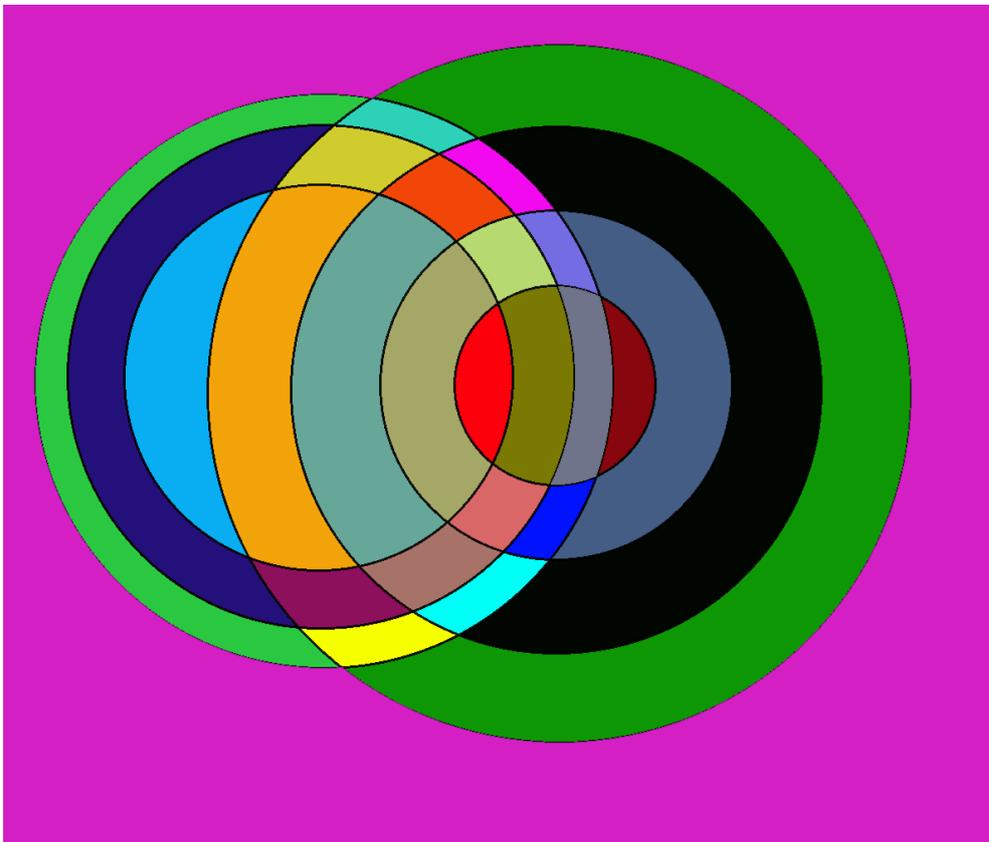
Real time limit: 2 s

Memory limit: 64M

## На вкус и цвет все фломастеры разные

Тролль 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 C: Договор об услуге сортировки

Prob id: 3

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.

## Problem D: Лесозаготовка

Prob Id: 4

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

Некоторый игровой мир расположен на бесконечной прямоугольной клетчатой сетке. В одном из её квадратов 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". Этот робот за один ход может переместиться на одну из соседних по стороне клеток с той, на которой он находится. Запаса его аккумулятора хватает, чтобы переместиться не более чем на  $A$ , ( $A > 0$ ) ходов, 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.

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", радиус действия которого был равен  $B$ , причем  $B > A$ . 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, имеет по крайней мере один нечетный делитель и находится в пределах от 20 до  $10^{12}$ . Гарантируется, что имеется как минимум один способ подобрать два числа  $A$  и  $B$ , таких, что  $B > A > 0$ , 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  $B > A > 0$  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

36

### Output

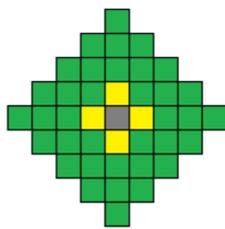
3

3 5

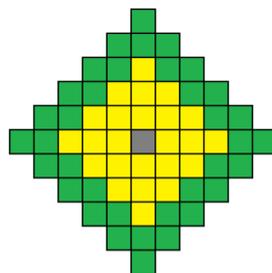
1 4

8 9

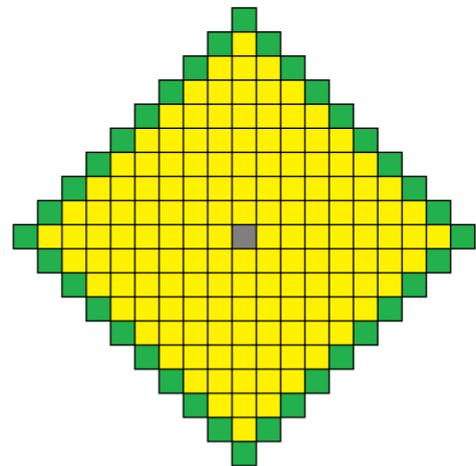
### Notes



**A = 1, B = 4, S = 36**



**A = 3, B = 5, S = 36**



**A = 8, B = 9, S = 36**

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 36 клеток. Можно показать, что других вариантов решения для **S = 36** 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 E: Острова и протоки

Prob id: 5

Full score: 100

Score bonus:

Input file name: input.txt or standard input

Output file name: output.txt or standard output

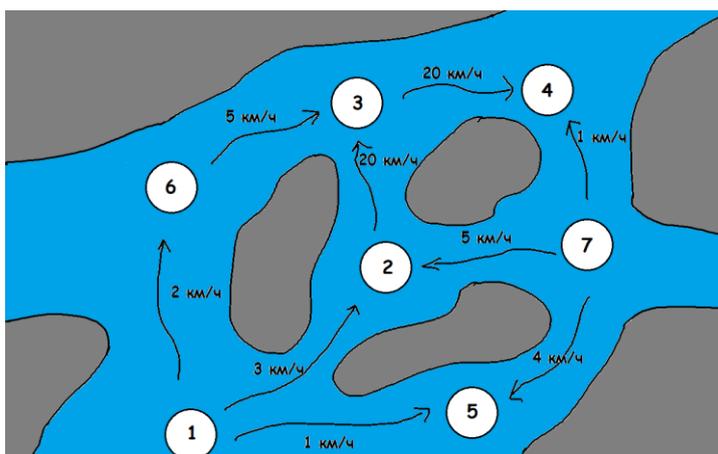
Time limit: 2 s

Real time limit: 2 s

Memory limit: 64M

## Острова и протоки

Группа туристов сплавляется на катамаране по реке. По ходу сплава они попали в сложную систему островов и протоков. Места слияния протоков на карте обозначены числами от 1 до  $n$ . Между некоторыми из них имеются потоки с течением. Для каждого такого потока известна его скорость в км/ч и направление течения. У всех протоков длина одинаковая и равна 1 км.



$v = 10$  км/ч

$$1 \rightarrow 2 \rightarrow 7: 1/13 + 1/5 \sim 0,277$$

$$1 \rightarrow 5 \rightarrow 7: 1/11 + 1/6 \sim 0,258$$

$$1 \rightarrow 6 \rightarrow 3 \rightarrow 4 \rightarrow 7: 1/12 + 1/15 + 1/30 + 1/9 \sim 0,294$$

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 7: 1/13 + 1/30 + 1/30 + 1/9 \sim 0,255$$

Постоянная скорость группы на катамаране в стоячей воде равна  $v$  км/ч. Туристы могут плыть как по течению, тогда их скорость и скорость течения суммируются, так и против течения, тогда из их скорости нужно вычесть скорость встречного потока. Само собой, если скорость встречного потока больше либо равна их собственной, они не смогут проплыть вдоль него.

Согласно заявленному маршруту, туристам нужно как можно быстрее попасть из пункта 1 в пункт  $n$ . Требуется определить за какое минимальное время они смогут это сделать.

## Input format

В первой строке содержится три целых числа  $n$ ,  $m$  и  $v$  через пробел -- число точек слияния протоков, число протоков и скорость группы в стоячей воде соответственно.  $3 \leq n \leq 10^5$ ,  $3 \leq m \leq 2 * 10^5$ ,  $1 \leq v \leq 100$ .

В следующих строках описаны протоки в виде трех целых чисел  $a$ ,  $b$  и  $vp$  через пробел. Здесь  $a$  и  $b$  -- точки слияния между которыми протекает данный проток и  $vp$  -- его скорость в км/ч. Течение в протоке направлено от пункта  $a$  к пункту  $b$ . Длина любого протока считается равной 1 км.

$1 \leq a, b \leq n$ ,  $0 \leq vp \leq 100$ .

## Output format

Вывести одно вещественное число -- наименьшее время в часах, за которое они смогут попасть из пункта 1 в пункт  $n$  с точностью не менее шести знаков после десятичной точки. Если возможности попасть из пункта 1 в пункт  $n$  нет, вывести -1.

## Examples

### Input

```
7 9 10
1 2 3
1 5 1
1 6 2
```

6 3 5  
2 3 20  
3 4 20  
7 5 4  
7 2 5  
7 4 1

### Output

0.254701

### Notes

На рисунке вы можете видеть карту для примера и описание всех вариантов движения из пункта 1 в пункт 7.

Можно видеть, что быстрее всего результат достигается по маршруту 1 -> 2 -> 3 -> 4 -> 7.

### Подзадачи

Подзадача	Номера тестов	Баллы	Ограничения	Необходимые подзадачи
0 (тесты из условия)	1	0		
1	2-10	10	$1 \leq n, m \leq 100, vp = 0$ для всех протоков	
2	11-23	30	$1 \leq n \leq 1000, 1 \leq m \leq 2000$	0, 1
3	24-35	60	без дополнительных ограничений	0, 1, 2

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 F: Партия в игру "Щелк"

Prob Id: 6

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

### Партия в игру "Щелк"

Два друга играют в интересную и вкусную игру "Щелк". У них есть шоколадка, разбитая на квадратные дольки, эти дольки расположены в  $n$  строк и  $m$  столбцов. Друзья по-очереди делают ходы. За один ход игрок, чья очередь сейчас ходить, выбирает одну из оставшихся долек и съедает все дольки, которые находятся выше и левее этой дольки, включая саму эту дольку. При этом фиксируются координаты выбранной дольки и количество съеденных на этом шаге долек.

В данной задаче нас не интересует ни кто выиграл, ни в чем смысл игры. Друзья даже могут не съесть шоколадку полностью. Ваша задача просто по описанию последовательности ходов друзей для каждого их хода вывести сколько долек было съедено на этом ходе.

## Input format

В первой строке заданы два натуральных числа  $n$  и  $m$  -- исходные размеры шоколадки.  $1 \leq n, m \leq 10^9$ .

Во второй строке находится число  $k$  -- количество сделанных в партии ходов.  $1 \leq k \leq \min(n * m, 3 * 10^5)$ .

В следующих  $k$  строках находятся по два целых числа  $x$  и  $y$  через пробел -- координаты текущего выбора игрока. Гарантируется, что все ходы правильные, то есть выбираемая на текущем ходе долька еще не съедена к этому моменту.  $1 \leq x \leq n, 1 \leq y \leq m$ .

## Output format

Вывести  $k$  строк. В каждую строку нужно вывести одно число -- количество съеденных на соответствующем ходу долек.

## Examples

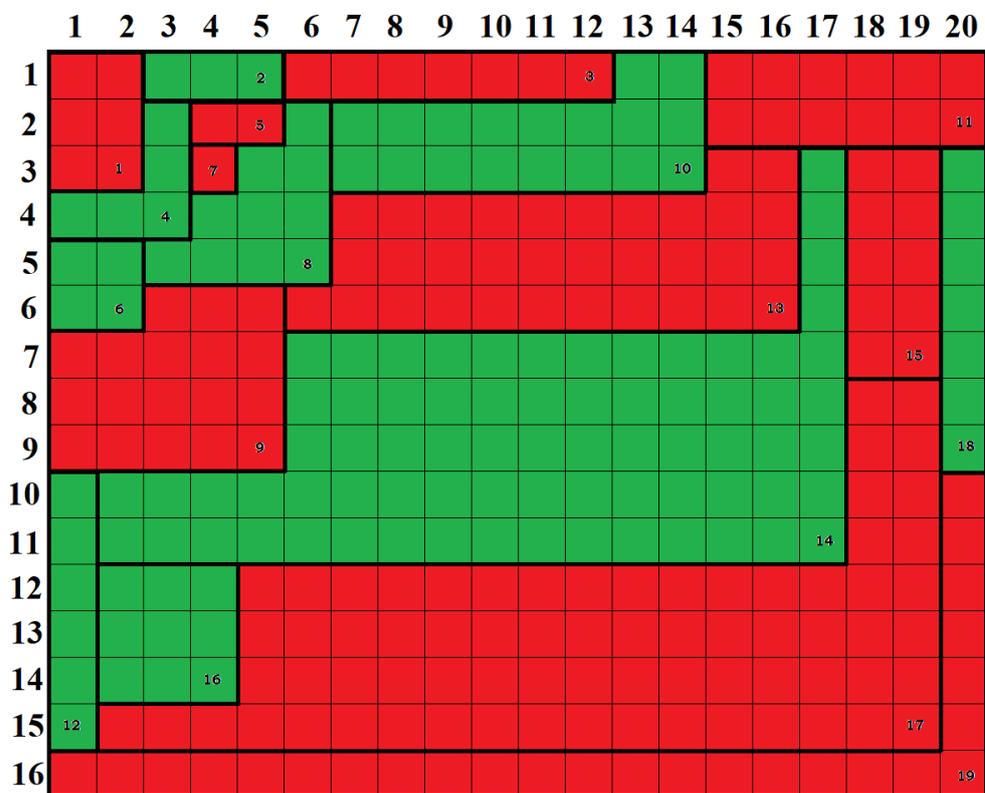
### *Input*

```
16 20
19
3 2
1 5
1 12
4 3
2 5
6 2
3 4
5 6
9 5
3 14
2 20
15 1
6 16
11 17
7 19
14 4
15 19
9 20
16 20
```

### *Output*

```
6
3
7
5
2
4
1
10
18
18
12
6
33
72
10
9
71
7
26
```

## Notes



На рисунке представлен ход игры для примера из условия. Первый игрок делает нечетные ходы, они отмечены красным цветом. Второй игрок делает четные ходы, они отмечены зеленым. Видно, что, например на восьмом ходу второй игрок выбирает дольку в пятой строке и шестом столбце и съедает десять долек. А на семнадцатом ходу первый игрок выбирает дольку в пятнадцатой строке и девятнадцатом столбце и съедает семьдесят одну дольку.

### Подзадачи

Подзадача	Номера тестов	Баллы	Ограничения	Необходимые подзадачи
0 (тесты из условия)	1	0		
1	2-9	5	$1 \leq n, m \leq 30, k \leq 900$	0
2	10-22	10	$1 \leq n, m \leq 300, k \leq 90000$	0, 1
3	23-30	25	$1 \leq n, m \leq 10^9, k \leq 3000$	0, 1, 2
4	31-50	25	$1 \leq n, m \leq 10^5, k \leq 3 * 10^5$	0, 1, 2
5	51-60	35	$1 \leq n, m \leq 10^9, k \leq 3 * 10^5$	0, 1, 2, 3, 4

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.