

## Задача 1. Мастер-класс

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Антон Алексеевич, преподаватель математики в старшей школе, стремится к совершенству в обучении своих учеников. Сегодня он решил продемонстрировать практическое применение математических расчётов и провести мастер-класс по очистке квадратной маркерной доски, длины  $n$  квадратной губкой, длиной  $a$ .

Антон Алексеевич использует свой фирменный метод, затирая каждый сантиметр доски без лишних движений. Он начинает с верхнего левого угла доски, двигаясь слева направо, затем вниз, влево, вверх, повторяя этот процесс, пока вся доска не будет очищена, не проходя губкой по тем местам, которые уже очищены (см. чертёж). Задача заключается в определении длины ломаной линии, которую описывает верхний левый угол квадратной губки при затирании квадратной доски. Губка не вращается во время стирания доски.

**Гарантируется, что длина стороны губки делит длину стороны доски без остатка.**

### Формат входных данных

Первая строка содержит целое число  $n$  ( $1 \leq n \leq 10^9$ ) — длину стороны доски.

Вторая строка содержит целое число  $a$  ( $1 \leq a \leq 10^9$ ,  $a \leq n$ ) — длину стороны губки.  $a$  делит  $n$  без остатка.

### Формат выходных данных

В единственной строке выведите число — длину ломаной, которую пройдёт левый верхний угол губки.

**Обратите внимание, что ответ в этой задаче может превышать возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битные целочисленные типы данных (тип `int64` в языке Pascal, тип `long long` в C++, тип `long` в Java и C#).**

### Система оценки

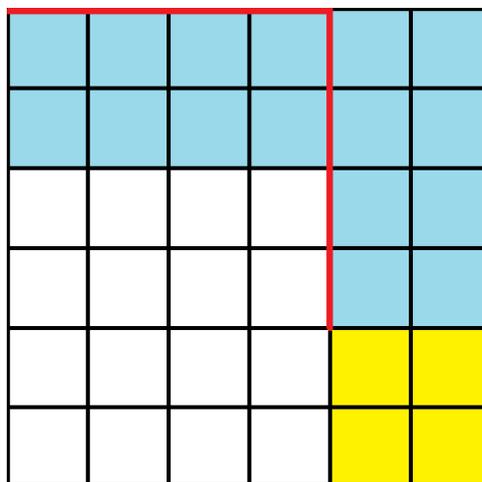
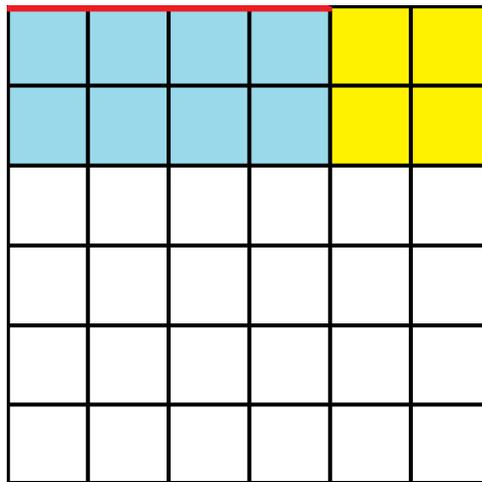
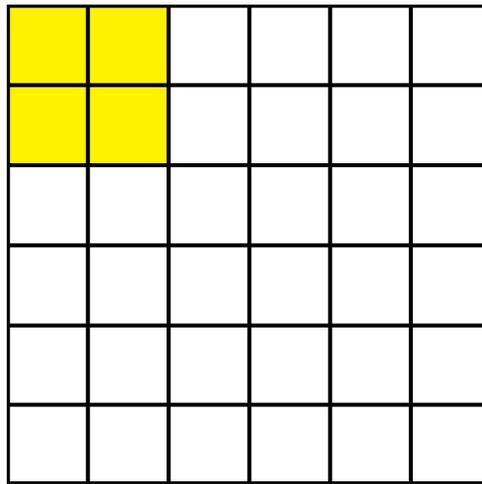
Решения, правильно работающие при  $n, a \leq 10^3$ , будут оцениваться в 40 баллов.

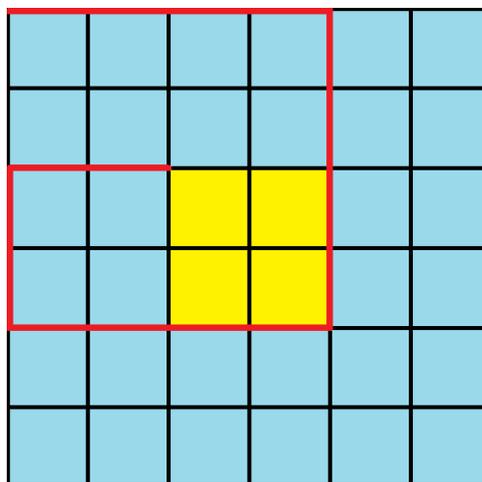
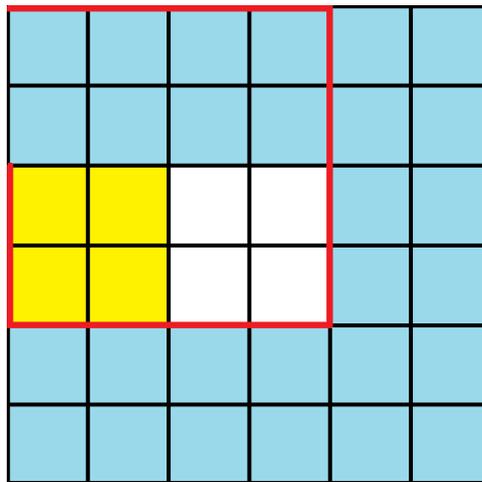
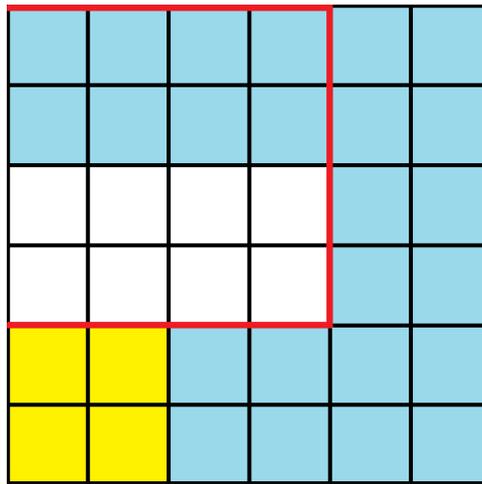
### Пример

стандартный ввод	стандартный вывод
6	16
2	

### Замечание

Проиллюстрируем пример к условию. Жёлтым на схеме обозначена губка, голубым — очищенная область, красным — маршрут левого верхнего угла губки.





## Задача 2. Наши слоны

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

На столе лежит шахматная доска, в которой  $n$  строк и  $m$  столбцов. Фигура «слон» может ходить по диагонали на любое количество клеток. Пустая клетка находится «под боем», если какой-либо из слонов на доске может одним ходом перейти на эту клетку. На доске в четырёх углах стоят четыре слона. Определите, сколько клеток находится «под боем».

### Формат входных данных

В первой строке содержится количество строк шахматной доски  $n$ , а во второй — столбцов  $m$  ( $2 \leq n, m \leq 10^8$ ).

### Формат выходных данных

В единственной строке выведите целое число – количество клеток, находящихся «под боем».

### Система оценки

Решения, правильно работающие при  $n, m \leq 500$ , будут оцениваться в 52 балла.

### Примеры

стандартный ввод	стандартный вывод
2 4	4
2 2	0
4 7	10

### Замечание



Иллюстрация к примеру 1

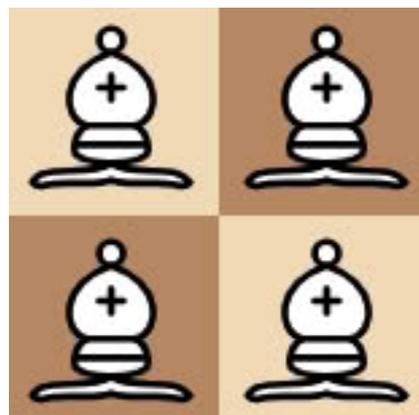


Иллюстрация к примеру 2

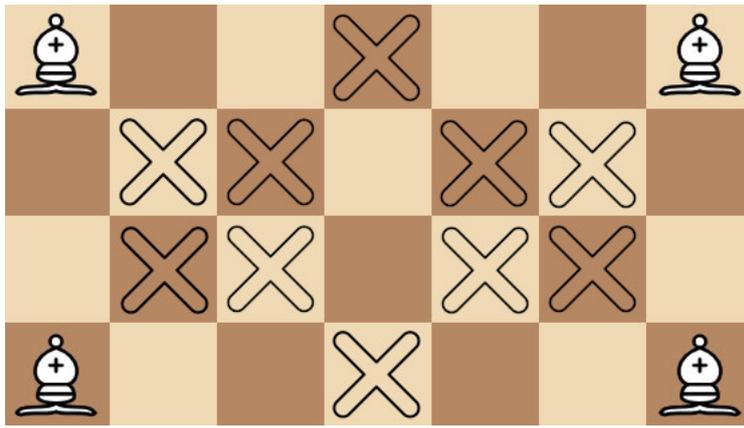


Иллюстрация к примеру 3

## Задача 3. Пирамидки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Ваня собирает пирамидки из кубиков. Всего он хочет построить  $n$  пирамидок. Ваня собирает пирамидки следующим образом:

1. Изначально в каждой пирамидке находится по  $a_1$  кубиков в первом ряду.
2. Каждой второй пирамидке, т.е. чей номер делится на 2, добавляется по  $a_2$  кубиков во второй ряд.
3. Каждой четвертой пирамидке, т.е. чей номер делится на 4, добавляется по  $a_3$  кубиков в третий ряд.
4. Каждой восьмой пирамидке, т.е. чей номер делится на 8, добавляется по  $a_4$  кубиков в четвертый ряд, и так далее.

Более формально, каждой пирамидке, чей номер делится на  $i - 1$  степень двойки, добавляется по  $a_i$  кубиков в  $i$  ряд. Помогите Ване определить количество кубиков, которое ему нужно для построения всех пирамидок.

### Формат входных данных

Первая строка входных данных содержит число  $n$  ( $1 \leq n \leq 10^9$ ) — количество пирамидок, которые хочет построить Ваня.

Вторая строка входных данных содержит число  $k$  ( $1 \leq k \leq 30$ ) — количество рядов, для которых известно, сколько в них будет кубиков. Гарантируется, что в каждой пирамидке не более чем  $k$  рядов. Каждая из следующих  $k$  строк содержит  $a_i$  ( $1 \leq a_i \leq 10^9$ ) — количество кубиков в  $i$ -м ряде пирамидки, количество кубиков, требующихся для каждого следующего слоя не убывает.

### Формат выходных данных

В единственной строке выведите число — количество кубиков, которые нужны Ване для постройки всех пирамидок.

Обратите внимание, что ответ в этой задаче может превышать возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битные целочисленные типы данных (тип `int64` в языке Pascal, тип `long long` в C++, тип `long` в Java и C#).

### Система оценки

Решения, правильно работающие при  $n \leq 15$ ,  $a_i \leq 100$ , будут оцениваться в 20 баллов.

Решения, правильно работающие при  $n \leq 10^5$ , будут оцениваться в 50 баллов.

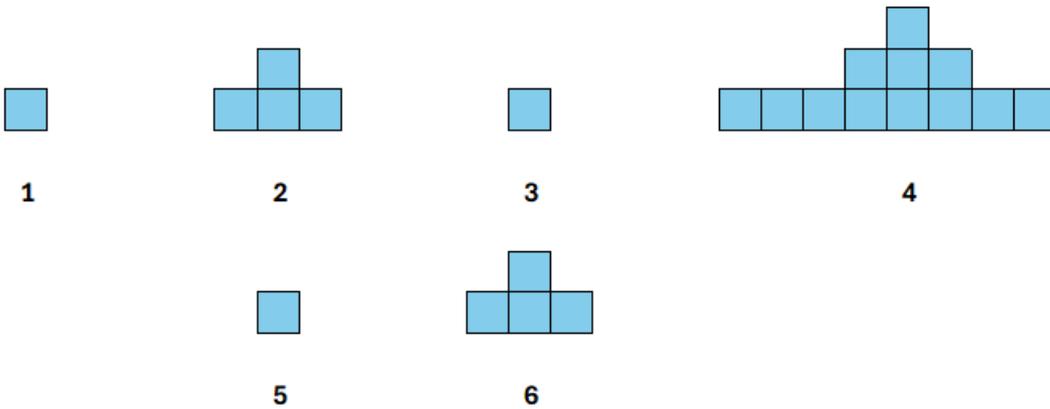
### Примеры

стандартный ввод	стандартный вывод
6 3 1 3 8	23
8 4 2 5 8 11	63

## Замечание

Рассмотрим первый пример. Пирамидки с нечетными номерами 1, 3, 5 имеют по одному ряду и состоят из 1 кубика. Пирамидки с номерами 2, 6 имеют по два ряда и состоят из  $1+3$  кубиков. Пирамидка с номером 4 имеет 3 ряда и состоит из  $1+3+8$  кубиков. В сумме Ване потребуется  $1 + (1 + 3) + 1 + (1 + 3 + 8) + 1 + (1 + 3) = 23$  кубиков.

Сами пирамидки изображены на рисунке ниже.



## Задача 4. Тайна Магических Доминошек

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В далёком королевстве, где магия и реальность переплетаются, существует легенда о бесконечной цепи магических доминошек. Эти доминошки выстроены вдоль старинной дороги и обладают особыми свойствами. Каждая доминошка расположена на координате  $x_i$  и имеет магическую силу, равную  $h_i$ , определяющую её способность влиять на другие доминошки.

Однажды юный волшебник Гарри решил испытать силу этой цепи и толкнул первую доминошку. При падении доминошка выпускает магический импульс, который распространяется вперёд и заставляет падать все доминошки, находящиеся на координатах не больше, чем  $x_i + h_i$ . Каждая следующая доминошка, попавшая под действие импульса, действует аналогично, создавая эффект магической цепной реакции.

Строго говоря, при воздействии на доминошку на позиции  $x_i$  будут задеты, и как следствие, выпустят уже свой магический импульс, все доминошки с такими координатами  $x_j$ , что  $x_i + h_i \leq x_j$ .

Кроме Гарри толкнуть доминошки решила и Гермиона. Она решила проверить, как поведут себя доминошки, если их толкнуть с другой стороны. Сложив свои заклинания и магические силы, она толкнула последнюю доминошку, которая находилась на самой правой позиции. И вот что произошло:

Самая правая доминошка, которую толкнула Гермиона, выпустила магический импульс, который распространяется назад, заставляя падать доминошки, находящиеся на координатах не больше, чем  $x_i - h_i$ . Иными словами, если доминошка на позиции  $x_i$  падает, то она сама выпускает магический импульс, под воздействие которого попадают все доминошки, чьи координаты удовлетворяют условию  $x_i - h_i \geq x_j$ .

Таким образом, доминошки, падающие с одной стороны, могут встретиться с доминошками, падающими с другой, при этом магические импульсы с двух сторон продолжают распространяться так же, как и до встречи. Теперь обе цепные реакции Гарри и Гермионы идут навстречу друг другу, и вы очень хотите узнать, сколько же суммарно доминошек упадёт?

### Формат входных данных

Первая строка содержит одно целое число  $N$  ( $2 \leq N \leq 10^5$ ) — количество волшебных доминошек.

Следующие  $2N$  строк содержат два набора целых чисел:

Первые  $N$  строк содержат координаты доминошек  $x_i$  ( $0 \leq x_i \leq 10^9$ ). Следующие  $N$  строк содержат магическую силу доминошек  $h_i$  ( $0 \leq h_i \leq 10^9$ ). Гарантируется, что координаты образуют возрастающую последовательность (никакие 2 доминошки не имеют одинаковых координат).

### Формат выходных данных

Выведите одно целое число — общее число доминошек, которые упадут, если волшебники толкнут доминошки.

### Система оценки

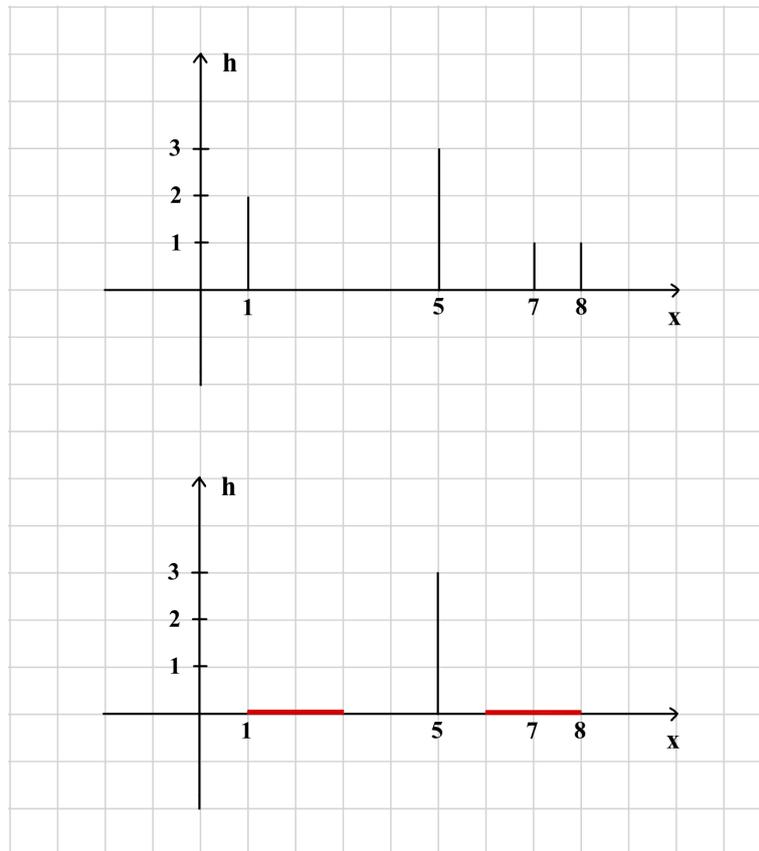
Решения, правильно работающие при  $n \leq 1000$ , будут оценены не менее чем в 50 баллов.

## Пример

стандартный ввод	стандартный вывод
4 1 5 7 8 2 3 1 1	3

## Замечание

В первом примере первая доминошка, которую толкнул Гарри, не достанет до следующей (т.к. её «высота» всего 2), а поэтому от импульса слева упадёт только она. Доминошка, стоящая на координате 8, которую толкнула Гермиона, заденет доминошку, стоящую на координате 7, которая, в свою очередь, не заденет доминошку на координате 5, т.к. её «высота» равна 1. На рисунке ниже показаны доминоши до того, как их толкнули и их итоговое состояние (красным отмечены места, которые «задела» хотя бы одна доминошка.)



Пояснение к первому примеру

## Задача 5. Покраска забора

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Недавно Егор узнал, что средняя зарплата маляра в России составляет более ста тысяч рублей в месяц. Сейчас он получает немного меньше (примерно ноль рублей в месяц), а потому пошел работать маляром.

В первый же день работы он получил свое первое задание — покрасить забор, состоящий из  $k$  досок. Опишем этот процесс немного подробнее. Забор можно представить как  $k$  палок, стоящих в ряд. Слева от забора стоит телега, в которой есть  $n$  банок с красками — при этом  $i$ -й банки хватит ровно на  $a_i$  досок.

Изначально Егор стоит слева от забора (прямо перед телегой), при этом в его руке нет банки с краской. Егор пока что не окончил ВУЗ, поэтому набор его способностей ограничен:

- Взять банку с краской. Если Егор стоит перед телегой, то он может взять в руку любую банку, в которой еще осталась краска. Если у Егора уже была какая-то банка в руке, то он возьмет новую, а старую оставит у телеги. Это действие **не занимает** времени.
- Сдвинуться вправо. После этого Егор перейдет к следующей доске. Если Егор стоял у телеги, то после этого перемещения он будет стоять перед первой доской. При этом Егор не может сдвинуться вправо, если он уже у последней доски. Это действие занимает 1 секунду.
- Сдвинуться влево. После этого Егор перейдет к предыдущей доске. Если Егор стоит у первой доски, то после этого перемещения он будет стоять перед телегой. Конечно же, Егор не может сдвинуться влево, если он стоит перед телегой. Это действие занимает 1 секунду.
- Покрасить доску. Если Егор сейчас стоит перед непокрашенной доской, а также у него в руках есть банка с краской, то он может покрасить эту доску. Это действие занимает 1 секунду.

Раньше Егор был пасечником, а потому ему не понаслышке известна фраза «Время — деньги!». Соответственно, он хочет как можно быстрее покрасить весь забор. Вы ведь помните, что Егор не окончил ВУЗ? Именно поэтому вам придется посчитать, какое минимальное время потребуется на покраску всего забора. **Обратите внимание** на то что Егор имеет право закончить покраску в любом месте забора, а также ему не надо после этого возвращаться к телеге.

### Формат входных данных

В первой строке дано целое число  $k$  ( $1 \leq k \leq 10^9$ ) — количество досок в заборе.

Во второй строке дано целое число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — количество банок в телеге.

В следующих  $n$  строках даны целые числа  $a_i$  ( $1 \leq a_i \leq 10^9$ ) — на сколько досок хватит краски в банке с номером  $i$ .

### Формат выходных данных

Если Егору не хватит краски для выполнения своего задания, то выведите «-1» (без кавычек). Иначе выведите минимальное количество секунд, которое потребуется Егору для покраски всего забора.

## Примеры

стандартный ввод	стандартный вывод
5 3 1 3 2	14
100 1 1	-1
6 3 5 5 5	14

## Замечание

Рассмотрим первый пример. В заборе есть всего пять досок. Чтобы покрасить его как можно быстрее, Егору нужно сначала взять третью банку, потом покрасить первые две доски — на это у него уйдет 4 секунды. Затем он должен вернуться назад и взять банку, которой хватит на три доски, — на это еще 2 секунды. Затем надо докрасить забор — на это уйдет еще 8 секунд, первые две из которых он будет идти рядом с уже покрашенными досками. Итого потребуется  $4 + 2 + 8 = 14$  секунд.

Во втором примере у Егора есть возможность покрасить только одну доску, но этого не хватает на покраску всего забора.