Задача A. Кривая дракона High (только для высшей лиги)

Имя входного файла:	`a.in`
Имя выходного файла:	`a.out`
Ограничение по времени:	1 с
Ограничение по памяти:	64 Мб

Кривая дракона – это бесконечная ломаная из звеньев единичной длины, которая строится следующим образом. Выбирается некоторая точка на плоскости (например, точка $(0, 0)$) и одно из четырех направлений, параллельных координатным осям. Левый(правый) дракон порядка n строится так:

- если n равно 0, то отложить отрезок длины 1 от текущей точки в текущем направлении, переместившись в конечную точку;
- в противном случае построить левого дракона порядка $n - 1$ от текущей точки в текущем направлении, повернуться на 90 градусов налево(направо) в его конечной точке и построить правого дракона порядка $n - 1$.

Поскольку левый дракон порядка n содержит в качестве своего начала левого дракона порядка $n - 1$, то вполне корректно определяется левый дракон бесконечного порядка. Именно эта ломаная и называется кривой дракона.

Построим из точки $(0, 0)$ кривые дракона во всех четырех направлениях. Есть теорема, доказанная Д.Кнутом, о том, что эти кривые не пересекаются (за исключением касания в вершинах) и полностью покрывают целочисленную сетку на плоскости. В данной задаче от вас потребуется проверить эту теорему, а именно по заданному единичному отрезку сетки определить какому из четырех драконов он принадлежит.

Формат входного файла

В единственной строке входного файла задаются 4 целых числа: две координаты одного конца некоторого отрезка, параллельного одной из осей координат, и две координаты другого конца.

Ограничения

Все числа не превышают по модулю 10^9. Расстояние между точкам равно 1.

Формат выходного файла

В первой строке выходного файла выведите номер драконовой кривой, которой принадлежит соответствующее звено (1 – дракон, отложенный в положительном направлении оси Ox, 2 – в положительном направлении оси Oy, 3 – в отрицательном направлении оси Ox, 4 – в отрицательном направлении оси Oy). Во второй строке выведите номер этого звена в соответствующей ломаной. Гарантируется, что это число не превосходит 10^{12}.

Примеры

a.in	a.out
`1 0 1 1`	`1` `2`
`2 -2 2 -1`	`3` `8`

Задача B. Кривая дракона Junior (только для юниорской лиги)

Имя входного файла:	`b.in`
Имя выходного файла:	`b.out`
Ограничение по времени:	1 с
Ограничение по памяти:	64 Мб

Кривая дракона – это бесконечная ломаная из звеньев единичной длины, которая строится следующим образом. Выбирается некоторая точка на плоскости (например, точка $(0, 0)$) и одно из четырех направлений, параллельных координатным осям. Левый(правый) дракон порядка n строится так:

- если n равно 0, то отложить отрезок длины 1 от текущей точки в текущем направлении, переместившись в конечную точку;
- в противном случае построить левого дракона порядка $n-1$ от текущей точки в текущем направлении, повернуться на 90 градусов налево(направо) в его конечной точке и построить правого дракона порядка $n-1$.

Поскольку левый дракон порядка n содержит в качестве своего начала левого дракона порядка $n-1$, то вполне корректно определяется левый дракон бесконечного порядка. Именно эта ломаная и называется кривой дракона.

Построим из точки $(0, 0)$ кривые дракона во всех четырех направлениях. Есть теорема, доказанная Д.Кнутом, о том, что эти кривые не пересекаются (за исключением касания в вершинах) и полностью покрывают целочисленную сетку на плоскости. В данной задаче от вас потребуется проверить эту теорему, а именно по заданному единичному отрезку сетки определить какому из четырех драконов он принадлежит.

Формат входного файла

В единственной строке входного файла задаются 4 целых числа: две координаты одного конца некоторого отрезка, параллельного одной из осей координат, и две координаты другого конца.

Ограничения

Все числа не превышают по модулю 10^9. Расстояние между точкам равно 1.

Формат выходного файла

В первой строке выходного файла выведите номер драконовой кривой, которой принадлежит соответствующее звено (1 – дракон, отложенный в положительном направлении оси Ox, 2 – в положительном направлении оси Oy, 3 – в отрицательном направлении оси Ox, 4 – в отрицательном направлении оси Oy). Во второй строке выведите номер этого звена в соответствующей ломаной. Гарантируется, что это число не превосходит 10^7.

Примеры

b.in	**b.out**
`1 0 1 1`	`1` `2`
`2 -2 2 -1`	`3` `8`

Задача C. K-цифровое число High (только для высшей лиги)

Имя входного файла:	`c.in`
Имя выходного файла:	`c.out`
Ограничение по времени:	1 с
Ограничение по памяти:	8 Мб

Назовем число K-цифровым, если количество различных цифр в его десятичной записи (без учета незначащих ведущих нулей) не превышает K.

По заданному числу x найдите ближайшее к нему K-цифровое число.

Формат входного файла

В единственной строке задается два целых числа K и x без незначащих ведущих нулей.

Ограничения

$1 \le K \le 10$, $0 \le x \le 10^{10^6}$.

Формат выходного файла

Выведите такое K-цифровое число y, чтобы величина $|y - x|$ имела минимально возможное значение. Если таких чисел несколько, можно выводить любое из них.

Примеры

c.in	c.out
2 23456	23333
1 8691	8888

Задача D. K-цифровое число Junior (только для юниорской лиги)

Имя входного файла:	`d.in`
Имя выходного файла:	`d.out`
Ограничение по времени:	1 с
Ограничение по памяти:	8 Мб

Назовем число K-цифровым, если количество различных цифр в его десятичной записи (без учета незначащих ведущих нулей) не превышает K.

По заданному числу x найдите ближайшее к нему K-цифровое число.

Формат входного файла

В единственной строке задается два целых числа K и x без незначащих ведущих нулей.

Ограничения

$1 \le K \le 10$, $0 \le x \le 10^8$.

Формат выходного файла

Выведите такое K-цифровое число y, что величина $|y - x|$ имеет минимально возможное значение. Если таких чисел несколько, можно выводить любое из них.

Примеры

d.in	d.out
2 23456	23333
1 8691	8888

Задача E. Хромой король High (только для высшей лиги)

Имя входного файла:	`e.in`
Имя выходного файла:	`e.out`
Ограничение по времени:	1 с
Ограничение по памяти:	8 Мб

Рассмотрим бесконечную во все четыре стороны шахматную доску с квадратными клетками. Некоторую клетку этой доски назовем центром. Вертикали, которая проходит через центр, присвоим номер 0. Вертикалям, находящимся правее центральной, присвоим последовательно номера 1, 2, 3 и т.д., левее – -1, -2, -3 и т.д. Аналогично пронумеруем горизонтали (выше центральной – положительными числами, ниже – отрицательными). Координаты любой клетки тогда можно определить парой чисел – номером вертикали и номером горизонтали, в которой она находится. Пусть теперь в центре доски, то есть в клетке с координатами $(0, 0)$, стоит король. Он может перемещаться по стандартным шахматным правилам – в соседнюю клетку по горизонтали, вертикали или диагонали. Однако некоторые из направлений являются запрещенными.

Требуется определить за сколько ходов король сможет попасть в клетку с заданными координатами (x, y).

Формат входного файла

В первой строке входного файла заданы 8 чисел, определяющих возможность перемещения в соответствующем направлении. 1 обозначает разрешенное направление, 0 – запрещенное. Направления перечисляются в порядке обхода против часовой стрелки, начиная с положительного горизонтального (то есть вправо, вправо-вверх, вверх, влево-вверх, влево, влево-вниз, вниз, вправо-вниз). Во второй строке задаются координаты x и y клетки, в которую необходимо попасть.

Ограничения

$-10^{18} \le x, y \le 10^{18}$.

Формат выходного файла

Выведите единственное целое число – минимальное количество разрешенных ходов, которые потребуются королю для того, чтобы добраться из клетки $(0, 0)$ в клетку (x, y). В случае, если такого пути не существует, выведите число -1.

Примеры

e.in	e.out
`1 1 1 1 1 1 1 1` `3 4`	`4`
`1 0 1 0 1 0 1 0` `-3 -4`	`7`

Задача F. Хромой король Junior (только для юниорской лиги)

Имя входного файла:	`f.in`
Имя выходного файла:	`f.out`
Ограничение по времени:	1 с
Ограничение по памяти:	32 Мб

Рассмотрим бесконечную во все четыре стороны шахматную доску с квадратными клетками. Некоторую клетку этой доски назовем центром. Вертикали, которая проходит через центр, присвоим номер 0. Вертикалям, находящимся правее центральной, присвоим последовательно номера 1, 2, 3 и т.д., левее – -1, -2, -3 и т.д. Аналогично пронумеруем горизонтали (выше центральной – положительными числами, ниже – отрицательными). Координаты любой клетки тогда можно определить парой чисел – номером вертикали и номером горизонтали, в которой она находится. Пусть теперь в центре доски, то есть в клетке с координатами $(0, 0)$, стоит король. Он может перемещаться по стандартным шахматным правилам – в соседнюю клетку по горизонтали, вертикали или диагонали. Однако некоторые из направлений являются запрещенными.

Требуется определить за сколько ходов король сможет попасть в клетку с заданными координатами (x, y).

Формат входного файла

В первой строке входного файла заданы 8 чисел, определяющих возможность перемещения в соответствующем направлении. 1 обозначает разрешенное направление, 0 – запрещенное. Направления перечисляются в порядке обхода против часовой стрелки, начиная с положительного горизонтального (то есть вправо, вправо-вверх, вверх, влево-вверх, влево, влево-вниз, вниз, вправо-вниз). Во второй строке задаются координаты x и y клетки, в которую необходимо попасть.

Ограничения

$-500 \le x, y \le 500$.

Формат выходного файла

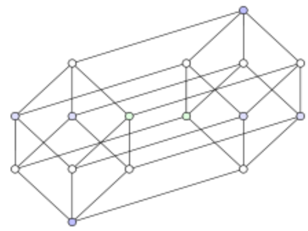
Выведите единственное целое число – минимальное количество разрешенных ходов, которые потребуются королю для того, чтобы добраться из клетки $(0, 0)$ в клетку (x, y). В случае, если такого пути не существует, выведите число -1.

Примеры

f.in	f.out
`1 1 1 1 1 1 1 1` `3 4`	`4`
`1 0 1 0 1 0 1 0` `-3 -4`	`7`

Задача G. Гиперкуб High (только для высшей лиги)

Имя входного файла:	`g.in`
Имя выходного файла:	`g.out`
Ограничение по времени:	4 с
Ограничение по памяти:	8 Мб



N-мерным гиперкубом (или N-кубом) со стороной $a > 0$ называется фигура в N-мерном евклидовом пространстве, которая строится следующим образом. Выберем в N-мерном пространстве какую-нибудь $(N-1)$-мерную гиперплоскость. Построим в ней $(N-1)$-куб. От каждой вершины этого куба проведем отрезок длины a перпендикулярно выбранной плоскости в одном и том же направлении. Концы этих отрезков будут очевидно также лежать в $(N-1)$-мерной плоскости, параллельной исходной, и образовывать $(N-1)$-куб. Выпуклая оболочка всех получившихся вершин (и исходного, и нового $(N-1)$-куба) образует N-мерный гиперкуб. По определению 0-куб – это точка, которая является единственной вершиной этого куба. Нетрудно увидеть, что 1-куб – отрезок на прямой, 2-куб – квадрат на плоскости, 3-куб – обычный трехмерный куб в трехмерном пространстве. Гиперкубы большей размерности увидеть несколько сложнее, но тем не менее формально можно построить по определению.

k-мерной гранью (или k-гранью) N-мерного гиперкуба называется такое пересечение его с k-мерной гиперплоскостью, которое не содержит внутренних точек граней большей размерности. Каждый N-куб имеет одну N-грань, совпадающую с ним самим. Можно доказать, что k-грани представляют собой k-кубы. 0-грани – это вершины N-куба, 1-грани – его ребра и т.д.

Требуется вычислить количество k-мерных граней N-мерного гиперкуба.

Формат входного файла

В единственной строке задаются три целых числа N, K и p.

Ограничения

$0 \le K \le N \le 10^{18}$, $K \le 2000$, $1 \le p \le 10^{18}$.

Формат выходного файла

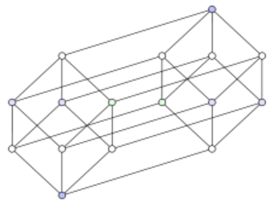
В единственную строку выведите $K+1$ число: остаток от деления на p количества 0-граней, 1-граней, $\ldots$ K-граней N-куба.

Примеры

g.in	g.out
`0 0 10`	`1`
`1 1 10`	`2 1`
`2 2 10`	`4 4 1`
`3 3 10`	`8 2 6 1`

Задача H. Гиперкуб Junior (только для юниорской лиги)

Имя входного файла:	`h.in`
Имя выходного файла:	`h.out`
Ограничение по времени:	1 с
Ограничение по памяти:	8 Мб



N-мерным гиперкубом (или N-кубом) со стороной $a > 0$ называется фигура в N-мерном евклидовом пространстве, которая строится следующим образом. Выберем в N-мерном пространстве какую-нибудь $(N-1)$-мерную гиперплоскость. Построим в ней $(N-1)$-куб. От каждой вершины этого куба проведем отрезок длины a перпендикулярно выбранной плоскости в одном и том же направлении. Концы этих отрезков будут очевидно также лежать в $(N-1)$-мерной плоскости, параллельной исходной, и образовывать $(N-1)$-куб. Выпуклая оболочка всех получившихся вершин (и исходного, и нового $(N-1)$-куба) образует N-мерный гиперкуб. По определению 0-куб – это точка, которая является единственной вершиной этого куба. Нетрудно увидеть, что 1-куб – отрезок на прямой, 2-куб – квадрат на плоскости, 3-куб – обычный трехмерный куб в трехмерном пространстве. Гиперкубы большей размерности увидеть несколько сложнее, но тем не менее формально можно построить по определению.

k-мерной гранью (или k-гранью) N-мерного гиперкуба называется такое пересечение его с k-мерной гиперплоскостью, которое не содержит внутренних точек граней большей размерности. Каждый N-куб имеет одну N-грань, совпадающую с ним самим. Можно доказать, что k-грани представляют собой k-кубы. 0-грани – это вершины N-куба, 1-грани – его ребра и т.д.

Требуется вычислить количество k-мерных граней N-мерного гиперкуба.

Формат входного файла

В единственной строке задаются три целых числа N, K и p.

Ограничения

$0 \le K \le N \le 10000$, $K \le 2000$, $1 \le p \le 10^{18}$.

Формат выходного файла

В единственную строку выведите $K+1$ число: остаток от деления на p количества 0-граней, 1-граней, $\ldots K$-граней N-куба.

Примеры

h.in	**h.out**
`0 0 10`	`1`
`1 1 10`	`2 1`
`2 2 10`	`4 4 1`
`3 3 10`	`8 2 6 1`

Задача I. Ферзи High (только для высшей лиги)

Имя входного файла:	`i.in`
Имя выходного файла:	`i.out`
Ограничение по времени:	1 с
Ограничение по памяти:	128 Мб

На d-мерной шахматной доске, каждое измерение которой составляет N, в различных ячейках стоят K ферзей. Ферзь может перемещаться на любое натуральное количество клеток в любом прямом или диагональном направлении. Более точно, ферзь может попасть из ячейки с координатами $(x_1, x_2, \ldots, x_d)$ в ячейку с координатами $(x'_1, x'_2, \ldots, x'_d)$ тогда и только тогда, когда среди чисел $|x_i - x'_i|$ есть по крайней мере одно ненулевое, и все ненулевые значения равны между собой. Будем говорить, что один ферзь угрожает другому, если существует ход из ячейки первого ферзя в ячейку второго. Очевидно, что это отношение симметрично, то есть если один ферзь угрожает другому, то и другой первому. Будем говорить, что один ферзь находится под боем у другого, если второй угрожает первому и все ячейки между ними свободны. Это отношение также симметрично.

Требуется определить количество пар ферзей, угрожающих друг другу и находящихся под боем друг у друга.

Формат входного файла

В первой строке задаются три целых числа d, N, K. В каждой из последующих K строк записано по d целых чисел x_1, $\ldots$, x_d, определяющих координаты ячейки соответствующего ферзя.

Ограничения

$1 \le d \le 5$, $1 \le N \le 1000$, $0 \le K \le 40000$, $1 \le x_i \le N$.

Формат выходного файла

В первой строке выведите количество пар ферзей, угрожающих друг другу, во второй – количество пар ферзей, находящихся под боем друг у друга.

Примеры

i.in	i.out
2 8 5 1 1 3 3 3 1 1 3 5 5	8 7
4 5 4 1 1 1 1 3 1 3 1 5 3 5 3 4 4 4 4	4 4

Задача J. Ферзи Junior (только для юниорской лиги)

Имя входного файла:	`j.in`
Имя выходного файла:	`j.out`
Ограничение по времени:	1 с
Ограничение по памяти:	128 Мб

На шахматной доске размера $N \times N$ в различных клетках стоят K ферзей. Ферзь может перемещаться на любое натуральное количество клеток в горизонтальном, вертикальном или диагональном направлении. Более точно, ферзь может попасть из клетки с координатами (x, y) в клетку с координатами (x', y') тогда и только тогда, когда числа $|x - x'|$ и $|y - y'|$ равны между собой и отличны от нуля, либо одно из них равно нулю, а другое отлично от нуля. Будем говорить, что один ферзь угрожает другому, если существует ход из ячейки первого ферзя в клетку второго. Очевидно, что это отношение симметрично, то есть если один ферзь угрожает другому, то и другой первому. Будем говорить, что один ферзь находится под боем у другого, если второй угрожает первому и все клетки между ними свободны. Это отношение также симметрично.

Требуется определить количество пар ферзей, угрожающих друг другу и находящихся под боем друг у друга.

Формат входного файла

В первой строке задаются два целых числа N, K. В каждой из последующих K строк записано по два целых числа x, y, определяющих координаты соответствующего ферзя.

Ограничения

$1 \le N \le 10^6$, $0 \le K \le 10^5$, $1 \le x, y \le N$.

Формат выходного файла

В первой строке выведите количество пар ферзей, угрожающих друг другу, во второй – количество пар ферзей, находящихся под боем друг у друга.

Примеры

j.in	j.out
8 5 1 1 3 3 3 1 1 3 5 5	8 7
5 4 1 1 3 1 5 3 4 4	4 4

Задача K. Изменение на отрезке High (только для высшей лиги)

Имя входного файла:	`k.in`
Имя выходного файла:	`k.out`
Ограничение по времени:	1 с
Ограничение по памяти:	8 Мб

Задан набор из N целых чисел a_0, a_1, $\ldots$, a_{N-1}. Изначально все эти числа равны 0. Далее поступают запросы на изменение и вывод. Для запроса на изменение задаются три числа l, r, d. По этому запросу к каждому из элементов a_i $(l \le i \le r)$ необходимо прибавить значение d. Для запроса на вывод задается одно число i. По этому требуется вывести текущее значение элемента a_i.

Формат входного файла

В первой строке входного файла задается два целых числа N и M, обозначающих количество элементов и количество запросов соответственно. В последующих M строках задаются запросы. Запрос на изменение задается строкой вида “`A` l r d”, запрос на вывод – строкой “`Q` i”.

Ограничения

Все числа целые.
$1 \le N \le 10^6$, $0 \le M \le 10^6$, $0 \le l \le r < N$, $0 \le i < N$, $|d| \le 10^3$.

Формат выходного файла

Для каждого запроса на вывод выведите в отдельной строке текущее значение соответствующего элемента.

Пример

k.in	**k.out**

`10 6`
`A 3 7 1`
`Q 4`
`A 1 5 2`
`Q 4`
`Q 1`
`Q 6`
	`1`
	`3`
	`2`
	`1`

Задача L. Изменение на отрезке Junior (только для юниорской лиги)

Имя входного файла:	`l.in`
Имя выходного файла:	`l.out`
Ограничение по времени:	1 с
Ограничение по памяти:	8 Мб

Задан набор из N целых чисел a_0, a_1, $\ldots$, a_{N-1}. Изначально все эти числа равны 0. Далее поступают запросы на изменение и вывод. Для запроса на изменение задаются три числа l, r, d. По этому запросу к каждому из элементов a_i ($l \le i \le r$) необходимо прибавить значение d. Для запроса на вывод задается одно число i. По этому требуется вывести текущее значение элемента a_i.

Формат входного файла

В первой строке входного файла задается три целых числа N, M_A и M_Q, обозначающих количество элементов, количество запросов на изменение и на вывод соответственно. В последующих $M_A + M_Q$ строках задаются запросы. Запрос на изменение задается строкой вида "`A` l r d", запрос на вывод – строкой "`Q` i".

Ограничения

Все числа целые.
$1 \le N \le 10^6$, $0 \le M_A, M_Q \le 10^6$, $0 \le l \le r < N$, $0 \le i < N$, $|d| \le 10^3$.
Гарантируется, что по крайней мере одно из чисел N, M_A или M_Q не будет превосходить 50.

Формат выходного файла

Для каждого запроса на вывод выведите в отдельной строке текущее значение соответствующего элемента.

Пример

l.in	**l.out**
10 2 4 A 3 7 1 Q 4 A 1 5 2 Q 4 Q 1 Q 6	1 3 2 1

Задача M. СуперНим High (только для высшей лиги)

Имя входного файла:	`m.in`
Имя выходного файла:	`m.out`
Ограничение по времени:	2 с
Ограничение по памяти:	8 Мб

Ним – это игра для двух игроков, которые по очереди берут предметы, разложенные на несколько кучек. Каждый игрок в свой ход может выбрать одну из непустых кучек и забирать из нее любое ненулевое количество предметов. Проигрывает тот из игроков, кто в свою очередь не сможет сделать ход (то есть последний предмет заберет на предыдущем ходе его соперник).

Данная игра математически полностью проанализирована. Проигрышными позициями в игре называются такие позиции, что при любых дальнейших ходах игрока, который должен делать ход из данной позиции, его соперник имеет возможность делать такие ходы, которые приведут к проигрышу игрока (то есть соперник имеет выигрышную стратегию). Выигрышными – такие, что, каковы бы ни были в дальнейшем действия соперника, игрок, который выполняет ход из данной позиции, имеет возможность делать такие ходы, которые приведут его к победе (то есть игрок имеет выигрышную стратегию). Известно, что выигрышность позиции определяется ним-суммой: $p_1 \operatorname{xor} p_2 \operatorname{xor} \ldots \operatorname{xor} p_N$, где p_i – количество предметов в i-ой кучке, xor – операция побитового исключающего "или" (сложение двоичных представлений чисел без учета переносов в следующие разряды). Если эта сумма отлична от нуля, то позиция выигрышная, если равна нулю – проигрышная.

Здесь мы будем иметь дело с огромным количеством кучек. Ваша задача – по заданным размерам кучек определить количество выигрышных вариантов первого хода первого игрока из начальной позиции (то есть такие ходы, при которых выигрышная стратегия все еще будет оставаться у первого игрока).

Формат входного файла

В первой строке входного файла содержится целое число M, определяющее количество групп кучек. Каждая из последующих M строк описывает соответствующую группу с помощью трех целых чисел a, b, N. Если расположить все кучки в группе в порядке неубывания размеров, то a – размер наименьшей кучки, b – размер наибольшей, N – количество кучек в группе, а разность между размерами двух соседних кучек постоянна.

Ограничения

$1 \le M \le 20000$, $0 \le a \le b \le 10^{18}$, $1 \le N \le 10^{18}$, $b - a$ кратно $N - 1$.

Формат выходного файла

Выведите единственное целое число – количество вариантов первого хода, которое допускает выигрышная стратегия.

Примеры

m.in	m.out
2 3 5 3 2 6 3	3
2 1 6 6 8 15 2	0

Задача N. СуперНим Junior (только для юниорской лиги)

Имя входного файла:	`n.in`
Имя выходного файла:	`n.out`
Ограничение по времени:	1 с
Ограничение по памяти:	8 Мб

Ним – это игра для двух игроков, которые по очереди берут предметы, разложенные на несколько кучек. Каждый игрок в свой ход может выбрать одну из непустых кучек и забирать из нее любое ненулевое количество предметов. Проигрывает тот из игроков, кто в свою очередь не сможет сделать ход (то есть последний предмет заберет на предыдущем ходе его соперник).

Данная игра математически полностью проанализирована. Проигрышными позициями в игре называются такие позиции, что при любых дальнейших ходах игрока, который должен делать ход из данной позиции, его соперник имеет возможность делать такие ходы, которые приведут к проигрышу игрока (то есть соперник имеет выигрышную стратегию). Выигрышными – такие, что, каковы бы ни были в дальнейшем действия соперника, игрок, который выполняет ход из данной позиции, имеет возможность делать такие ходы, которые приведут его к победе (то есть игрок имеет выигрышную стратегию). Известно, что выигрышность позиции определяется ним-суммой: $p_1 \,\mathrm{xor}\, p_2 \,\mathrm{xor} \ldots \mathrm{xor}\, p_N$, где p_i – количество предметов в i-ой кучке, xor – операция побитового исключающего “или” (сложение двоичных представлений чисел без учета переносов в следующие разряды). Если эта сумма отлична от нуля, то позиция выигрышная, если равна нулю – проигрышная.

Здесь мы будем иметь дело с огромным количеством кучек. Ваша задача – по заданным размерам кучек определить количество выигрышных вариантов первого хода первого игрока из начальной позиции (то есть такие ходы, при которых выигрышная стратегия все еще будет оставаться у первого игрока).

Формат входного файла

В первой строке входного файла содержится целое число M, определяющее количество групп кучек. Каждая из последующих M строк описывает соответствующую группу с помощью двух целых чисел a, b. В группе будет $b - a + 1$ кучек с размерами от a до b включительно.

Ограничения

$1 \le M \le 100000$, $0 \le a \le b \le 10^{12}$.

Формат выходного файла

Выведите единственное целое число – количество вариантов первого хода, которое допускает выигрышная стратегия.

Примеры

n.in	n.out
2 3 5 2 6	5
2 1 4 11 14	0

Задача O. K-стороннее домино

Имя входного файла: `o.in`
Имя выходного файла: `o.out`
Ограничение по времени: 1.5 с
Ограничение по памяти: 8 Мб

Стандартный набор домино содержит 28 костяшек. Костяшка представляет собой прямоугольник, разделенный на 2 части. Каждая часть может содержать одно число из множества $\{0, 1, 2, 3, 4, 5, 6\}$. При этом числа на обоих частях могут совпадать. В наборе есть все возможные костяшки и никакие две костяшки не содержат одну и ту же пару чисел.

Рассмотрим набор домино, в котором костяшки разделяются на K частей. Числа, которые содержатся на частях костяшек, будут выбираться из некоторого множества A состоящего из N элементов. Две костяшки считаются одинаковыми, если множества чисел, записанных на них, совпадают с учетом кратности. Набор содержит все возможные костяшки без повторений.

Определите количество костяшек в наборе и общую сумму всех чисел на них.

Формат входного файла

В первой строке задаются два целых числа N, K. Во второй строке задаются числа a_i множества A.

Ограничения

$1 \le K \le 10^4$, $1 \le N \le 10^6$, $0 \le a_i \le 10^9$.
Все a_i различны.

Формат выходного файла

В единственную строку выведите два числа – количество костяшек в наборе и сумму всех чисел на них.

Пример

o.in	o.out
7 2 0 1 2 3 4 5 6	28 168

Задача P. Различные попарные суммы

Имя входного файла:	`p.in`
Имя выходного файла:	`p.out`
Ограничение по времени:	1 с
Ограничение по памяти:	8 Мб

Дано натуральное число n. Требуется построить последовательность различных натуральных чисел $a_1, a_2, \ldots, a_n$, не больших $2n^2 + 4n\sqrt{n}$, такую, что все их попарные суммы различны. Другими словами, все $n(n-1)/2$ чисел вида $a_i + a_j$, $1 \leq i < j \leq n$ должны быть различны. Гарантируется, что такая последовательность существует.

Формат входного файла

В единственной строке входного файла задано натуральное число $n \leq 5000$.

Формат выходного файла

В единственную строку выходного файла выведите через пробел числа $a_1, a_2, \ldots, a_n$. Если решений несколько выведите любое.

Примеры

p.in	p.out
2	1 2
3	3 1 2
8	1 2 3 5 8 13 21 34