

Задача А. Сражение под Шахматском

Имя входного файла: `sum-rooks.in`
 Имя выходного файла: `sum-rooks.out`
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 256 мегабайт

Однажды близ славного города Шахматск, что жил мирною жизнью в Тридевятиом царстве, был замечен вражеский флот из соседней провинции Крысинск.

Было принято решение срочно собирать силы для обороны единственного порта в городе. Гавань в городе представляет собой отрезок из N пристаней, к каждой из которых может причалить вражеская Ладья.

Чтобы определить, какое количество войск нужно направить к каждой пристани, необходимо уметь считать количество ладей на отрезке пристаней.

Все пристани в Шахматске пронумерованы числами от 1 до N .

Формат входных данных

Первая строка входного файла содержит два целых числа N и K — количество пристаней в Шахматске и количество запросов ($1 \leq N \leq 100\,000$, $0 \leq K \leq 100\,000$). Следующие K строк содержат следующие запросы:

1. `A i x` — данные разведки сообщили, что у i -ой пристани сейчас стоит ровно x ладей ($1 \leq i \leq n$, $0 \leq x \leq 10^9$);
2. `Q l r` — найти суммарное число ладей в пристанях с номерами от l до r ($1 \leq l \leq r \leq n$).

Изначально в пристанях никого нет.

Формат выходных данных

На каждый запрос вида `Q l r` нужно вывести единственное число — количество ладей на отрезке.

Примеры

<code>sum-rooks.in</code>	<code>sum-rooks.out</code>
5 9	0
A 2 2	2
A 3 1	1
A 4 2	2
Q 1 1	0
Q 2 2	5
Q 3 3	
Q 4 4	
Q 5 5	
Q 1 5	

Задача В. Лодейное поле

Имя входного файла: `rooks.in`
 Имя выходного файла: `rooks.out`
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 64 мегабайта

На шахматной доске 8×8 стоит белый король и несколько черных ладей. Найдите безопасный ход, который может сделать король, либо определите, что королю поставлен мат или пат.

Формат входных данных

В первой строке содержится позиция белого короля. Во второй строке записано число n ($0 \leq N \leq 63$) — количество черных ладей на поле. В последующих N строках записаны позиции ладей.

Формат выходных данных

Выведите слово `Checkmate`, если белому королю поставлен мат, слово `Stalemate`, если позиция патовая, или произвольный безопасный ход белого короля в противном случае.

Примеры

<code>rooks.in</code>	<code>rooks.out</code>
a1 2 b8 h2	Stalemate
e4 4 e3 e5 d4 f4	Checkmate
h8 1 g7	g7

Задача С. Кони ходят по очереди

Имя входного файла: `knight3.in`
Имя выходного файла: `knight3.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Переведите каждого из двух коней из одной клетки в другую за наименьшее общее число ходов. Два коня не могут одновременно находиться в одной клетке. **Ходы коней должны чередоваться.**

Формат входных данных

Во входном файле записаны координаты первого и второго коня, затем координаты клеток, куда нужно их переместить.

Формат выходных данных

Программа должна вывести последовательность ходов коней в виде нескольких строк. Первым символом в строке должен быть номер коня (1 или 2), затем, через пробел, координаты клетки, в которую он переставляется. Необходимо вывести любое из возможных оптимальных решений. Кони должны ходить по очереди, первым может ходить любой из коней, кони могут сделать различное число ходов.

Примеры

<code>knight3.in</code>	<code>knight3.out</code>
<code>a1</code>	<code>1 b3</code>
<code>c2</code>	<code>2 a1</code>
<code>c2</code>	<code>1 d4</code>
<code>a1</code>	<code>2 b3</code>
	<code>1 c2</code>
	<code>2 a1</code>

Задача D. Короля — в угол

Имя входного файла: `king1.in`
Имя выходного файла: `king1.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В левом нижнем углу доски $m \times n$ стоит король. Двое игроков по очереди ходят королем, перемещая его на одну клетку по вертикали вверх, по горизонтали вправо или по диагонали вправо-вверх. Проигрывает тот, кто не может сделать ход. Определите, какой из игроков имеет выигрышную стратегию.

Формат входных данных

На вход программе подается два натуральных числа m и n , не превосходящих 10^9 .

Формат выходных данных

Программа должна вывести номер игрока (1 или 2), который имеет выигрышную стратегию.

Примеры

<code>king1.in</code>	<code>king1.out</code>
3 4	1

Задача Е. Александрийские шахматы

Имя входного файла:	alex-chess.in
Имя выходного файла:	alex-chess.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Ученики параллели В' крайне разносторонние личности и добиваются успеха не только в программировании. Например, многие из них замечательно играют в шахматы, и, проводя за игрой все вечера, нередко опаздывают на жёсткий отбой.

Школьница Сашенька тоже любит шахматы, но абсолютно не разбирается в правилах — слишком много фигур, слишком мало абстракции. Поэтому она придумала свою собственную игру — Александрийские шахматы, которая, благодаря её богатой фантазии, получилась абсолютно не похожей на оригинал.

Правила Александрийских шахмат очень просты - игра происходит на бесконечной доске шириной 1, клетки которой пронумерованы натуральными числами, и допускает два вида ходов:

1. Поставить пешку на клетку с номером i , а если эта клетка уже занята, на первую свободную клетку с бóльшим номером.
2. Снять с доски пешку, стоящую на клетке с номером i .

Сашенька хочет воплотить свою игру в жизнь, но, хотя она и учится в В', её навыки в программировании далеки от идеала, так что она просит вас помочь ей.

Формат входных данных

Первая строка входного файла содержит натуральное число n — количество ходов, сделанных в игре ($1 \leq n \leq 100\,000$).

Следующие n строк содержат информацию о ходах. Число $a > 0$ обозначает, что на доску ставится новая пешка в клетку с номером a ($1 \leq a \leq 100\,000$). Число $a < 0$ обозначает, что с доски убирается пешка, стоявшая в клетке номер $|a|$. (Гарантируется, что эта клетка не была пуста).

Формат выходных данных

Для каждой добавляемой на доску пешки выведите одно натуральное число — номер клетки, на которой она будет стоять.

Примеры

alex-chess.in	alex-chess.out
6	5
5	6
5	7
5	6
-6	8
5	
5	

Задача F. Обход доски

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Дана шахматная доска размером $n \times m$. Необходимо построить обход всей доски ходом коня так, чтобы конь побывал во всех клетках доски ровно по одному разу. В задачах F1–F6 достаточно построить незамкнутый путь, а в задачах F7–F9 необходимо построить циклический обход, начальная и конечная клетка которого также должны быть связаны ходом коня.

Вы должны сдать на проверку текстовый файл, содержащий nm строчек. Каждая строчка должна содержать координаты ровно одной клетки. Две соседние координаты должны быть связаны ходом коня и каждая из nm клеток доски должна встречаться в этом файле ровно один раз. Каждая клетка записывается в виде “a1”, где сначала записывается одна из первых n букв латинского алфавита, затем — число от 1 до m . Например, для доски 4×5 сданный файл может быть таким:

```
a1
c2
d4
b5
a3
b1
d2
c4
a5
b3
c1
a2
b4
d5
c3
d1
b2
a4
c5
d2
```

Формат входных данных

Список файлов для сдачи:

- F1: путь на доске 5×5 ;
- F2: путь на доске 7×7 ;
- F3: путь на доске 6×12 ;
- F4: путь на доске 13×14 ;
- F5: путь на доске 20×20 ;
- F6: путь на доске 26×26 ;
- F7: цикл на доске 6×6 ;
- F8: цикл на доске 8×8 ;
- F9: цикл на доске 7×16 .

Кроме этого, сдайте программу, которую вы использовали для нахождения ответов. Эта программа не будет проверяться автоматически. Сдать эту программу необходимо как задачу “F”.