

## Задача А. Это как пятый, но на один пониже

Имя входного файла: floor4.in  
Имя выходного файла: floor4.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Знаете ли вы, почему четвертый этаж заперт и там не останавливается лифт? Потому что на самом деле четвертый, запертый, этаж, где не останавливается лифт, содержит бесконечное количество комнат, пронумерованных натуральными числами. На этот этаж регулярно приезжают дети, каждый из которых заранее выбрал, в какую комнату он хочет заселиться. Если выбранная комната оказывается свободна, то ребенок занимает ее, в противном случае он занимает первую свободную комнату с большим номером.

Кроме того, некоторые дети уезжают в середине смены. Сразу после отъезда ребенка его комната становится доступна для заселения следующего.

Промоделируйте работу преподавателей, ответственных за четвертый этаж, и научитесь быстро сообщать приезжающим детям, какую комнату им следует занимать.

### Формат входного файла

Первая строка входного файла содержит натуральное число  $n$  — количество прибытий и отъездов, происходящих в течение смены ( $n \leq 100\,000$ ).

Следующие  $n$  строк содержат информацию об детях. Число  $a > 0$  обозначает, что приехал школьник, желающий занять комнату номер  $a$  ( $a \leq 100\,000$ ). Число  $a < 0$  обозначает, что с комнаты номер  $|a|$  уехал школьник. (Гарантируется, что эта комната не была пуста).

### Формат выходного файла

Для каждого приезжающего школьника выведите одно натуральное число — номер комнаты, в которую он поселится.

### Пример

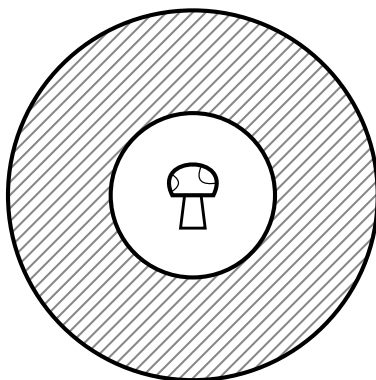
| floor4.in | floor4.out |
|-----------|------------|
| 6         | 5          |
| 5         | 6          |
| 5         | 7          |
| 5         | 6          |
| -6        | 8          |
| 5         |            |
| 5         |            |

## Задача В. Грибы

|                         |               |
|-------------------------|---------------|
| Имя входного файла:     | mushrooms.in  |
| Имя выходного файла:    | mushrooms.out |
| Ограничение по времени: | 1 секунда     |
| Ограничение по памяти:  | 256 мегабайт  |

В очень большом лесу раскинулась огромная поляна. После проливного дождя на поляне последовательно вырастают грибы в течении  $n$  минут. На первой минуте в центре поляны вырастает гриб высотой  $h_1$  сантиметров. На второй минуте на расстоянии 1 метр вокруг первого гриба вырастают грибы высотой  $h_2$ . На  $n$ -ой минуте на расстоянии 1 метр вокруг уже существующих грибов вырастают грибы высотой  $h_n$ . Грибы растут кольцами.

Разумеется, что такой урожай не мог остаться незамеченным. Поэтому была сформирована Регистрационная Единая Государственная Грибная Инспекция, которая занимается исследованием высоты грибов выросших после дождя. А именно, инспекцию интересует, какой самый высокий гриб находится в кольце от  $r$  до  $R$ . Будем называть кольцом область заштрихованную на рисунке (включая границы). Причем  $r$  - радиус меньшей окружности, а  $R$  - радиус большей окружности, центры окружностей находятся в точке, где вырос первый гриб.



Так как поляна может быть достаточно большой, то инспекция должна проверить несколько колец. Причем, если на  $i - 1$ -ой проверке было выявлено, что самый высокий гриб имеет высоту  $H$  (в кольце от  $r$  до  $R$ ), то на  $i$ -ой проверке необходимо выявить гриб в кольце от  $\min(R, (r \cdot H + H^2) \bmod n)$  до  $\max(R, (r \cdot H + H^2) \bmod n)$ . Для отчета инспекция хочет узнать только сумму высот грибов полученных, в результате всех проверок.

### Формат входного файла

В первой строке входного файла задано целое число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) – количество минут в течении которых вырастают грибы. Во второй строке задано  $n$  целых чисел  $h_i$  – высота грибов, выросших на  $i$  минуте ( $1 \leq h_i \leq 10^9$ ). В третьей строке задано число  $m$  ( $1 \leq m \leq 5 \cdot 10^6$ ) – количество проверок. Последняя строка содержит два целых числа  $r$  и  $R$  ( $0 \leq r \leq R < n$ ) – описание первой проверки.

### Формат выходного файла

Выведите ответ на задачу – сумму высот грибов, полученных в результате всех проверок.

## Пример

| mushrooms.in                   | mushrooms.out |
|--------------------------------|---------------|
| 5<br>1 2 3 4 5<br>3<br>1 4     | 15            |
| 7<br>1 1 4 2 3 5 1<br>1<br>0 6 | 5             |

## Задача С. Художник

Имя входного файла: `painter.in`  
Имя выходного файла: `painter.out`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 256 мегабайт

Не успев дорисовать свой гениальный футуристический шедевр, М. Калевич увлекся рисованием одномерных черно-белых картин. Он пытается найти оптимальное местоположение и количество черных участков картины. Для этого он проводит на прямой белые и черные отрезки, и после каждой из таких операций хочет знать количество черных отрезков на получившейся картине и их суммарную длину.

Изначально прямая — белая. Ваша задача — написать программу, которая после каждой такой операции выводит в выходной файл интересующие художника данные.

### Формат входного файла

В первой строке входного файла содержится общее количество нарисованных отрезков ( $1 \leq N \leq 100\,000$ ). В последующих  $N$  строках содержится описание операций. Каждая операция описывается строкой вида  $s\ x\ l$ , где  $s$  — цвет отрезка ( $W$  для белых отрезков,  $B$  для черных), а сам отрезок имеет вид  $[x; x + l]$ , причем координаты обоих концов — целые числа, не превосходящие по модулю 500 000. Длина задается положительным целым числом.

### Формат выходного файла

После выполнения каждой из операций необходимо вывести в выходной файл на отдельной строке количество черных отрезков на картине и их суммарную длину, разделенные одним пробелом.

### Пример

| <code>painter.in</code> | <code>painter.out</code> |
|-------------------------|--------------------------|
| 7                       | 0 0                      |
| W 2 3                   | 1 2                      |
| B 2 2                   | 1 4                      |
| B 4 2                   | 1 4                      |
| B 3 2                   | 2 6                      |
| B 7 2                   | 3 5                      |
| W 3 1                   | 0 0                      |
| W 0 10                  |                          |

## Задача D. Горы

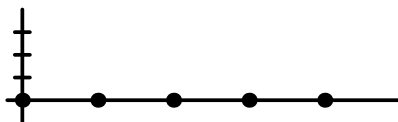
Имя входного файла: `mountains.in`  
Имя выходного файла: `mountains.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

В парке развлечений «Ай-ой-ай» открылся новейший аттракцион: польские горки. Трек состоит из  $n$  рельс, присоединённых одна к концу другой. Начало первой рельсы находится на высоте 0. Оператор Петя может конфигурировать аттракцион, изменяя по своему желанию подъём нескольких последовательных рельс. При этом подъём всех остальных рельс не изменяется. При каждом изменении конфигурации рельс положение следующих за изменяемыми подбирается таким образом, чтобы весь трек оставался связным.

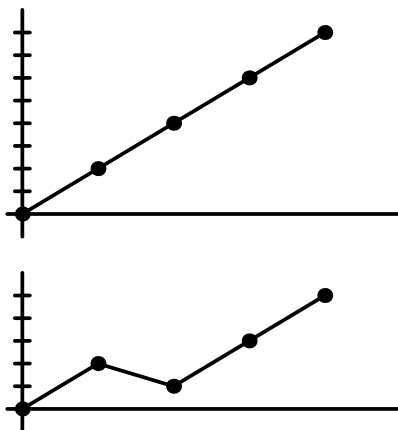
Каждый запуск вагонетки осуществляется с энергией, достаточной для достижения высоты  $h$ . Это значит, что вагонетка будет двигаться до тех пор, пока высота не превысит  $h$ , либо пока не закончится трек.

По записям о всех изменениях конфигурации рельс и временах запусков вагонетки для каждого запуска определите, сколько рельс вагонетка проедет до остановки.

Трек можно представить как последовательность  $n$  подъёмов  $d_i$ , по одному на рельс. Изначально все рельсы горизонтальны, то есть  $d_i = 0$  для всех  $i$ .



Каждое изменение конфигурации определяется числами  $a$ ,  $b$  и  $D$ : все рельсы с  $a$ -й по  $b$ -ю включительно после этого действия имеют подъём  $D$ .



Каждый запуск вагонетки определяется единственным целым числом  $h$  — максимальной высотой, на которую способна подняться вагонетка.

### Формат входного файла

В первой строке записано целое число  $n$  ( $1 \leq n \leq 10^9$ ) — число рельс. Следующие строки содержат запросы трёх видов:

- $I \ a \ b \ D$  — изменение конфигурации. Рельсы с  $a$ -й по  $b$ -ю включительно после выполнения запроса имеют подъём, равный  $D$ .
- $Q \ h$  — запуск вагонетки. Требуется найти число рельс, которое проедет вагонетка, которая способна подняться на высоту  $h$ .

- E — конец ввода. Этот запрос встретится ровно один раз в конце файла.

В любой момент высота любой точки трека лежит от 0 до  $10^9$ . Во вводе не более 100 000 строк.

### Формат выходного файла

Для каждого запроса Q выведите единственное целое число — количество рельс, которое проедет вагонетка.

### Пример

| mountains.in | mountains.out |
|--------------|---------------|
| 4            | 4             |
| Q 1          | 1             |
| I 1 4 2      | 0             |
| Q 3          | 3             |
| Q 1          |               |
| I 2 2 -1     |               |
| Q 3          |               |
| E            |               |

## Задача Е. Золотые рудники

Имя входного файла: `mine.in`  
Имя выходного файла: `mine.out`  
Ограничение по времени: 0.25 секунды  
Ограничение по памяти: 256 мегабайт

Байтмен, один из заслуженных работников компании по добыче золота в Байтленде, собирается в этом году на пенсию. Начальство компании решило вознаградить его за заслуги перед отечеством. Байтмену разрешили присвоить себе прямоугольную часть земли со сторонами  $s$  и  $w$ , параллельными осями координат, со всеми входящими туда рудниками. Положение участка он выбирает сам. Назовём стоимостью участка количество рудников, лежащих внутри него и на границе. Ваша задача — вычислить максимально возможную стоимость такого участка.

### Формат входного файла

В первой строке ввода записаны целые числа  $s$  и  $w$ .  $1 \leq s, w \leq 10\,000$ . Они означают длины сторон участка, параллельных осям  $OX$  и  $OY$ , соответственно.

Во второй строке записано целое число  $N$ : количество рудников.  $1 \leq N \leq 15\,000$ . В следующих  $N$  строках записаны координаты рудников: пары целых чисел, не превышающих 30 000 по абсолютной величине.

### Формат выходного файла

Выведите одно целое число: наибольшее количество рудников на участке.

### Пример

| mine.in   | mine.out |
|---|----------|
| 1 2<br>12<br>0 0<br>1 1<br>2 2<br>3 3<br>4 5<br>5 5<br>4 2<br>1 4<br>0 5<br>5 0<br>2 3<br>3 2 | 4        |