

## Задача А. Окончание сессии

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

После успешной сдачи последнего экзамена студент Вася решил позвать своих одноклассников в кафе, чтобы вместе отметить окончание сессии молочным коктейлем. Вася решил заказать молочный коктейль, состоящий из смеси  $N$  различных ингредиентов.

Известно, что одна порция коктейля весит  $M$  грамм. Вася определился с процентным содержанием каждого выбранного ингредиента в порции, и теперь он хочет посчитать, сколько граммов каждого ингредиента содержится в его порции.

Помогите Васе посчитать вес каждого выбранного им ингредиента коктейля.

### Формат входного файла

Первая строка входного файла содержит два целых числа, разделенных пробелом:  $N, M$  ( $1 \leq N \leq 100, 1 \leq M \leq 10^{16}$ ) — количество различных ингредиентов и общее число граммов в порции.

Во второй строке через пробелы перечислены  $N$  чисел — процентные содержания каждого ингредиента (целые числа от 0 до 100). Сумма всех процентных содержаний равно 100.

### Формат выходного файла

В выходной файл выведите  $N$  целых чисел — количество граммов каждого ингредиента в одной порции коктейля. Порядок чисел должен соответствовать порядку, заданному во входном файле.

### Примеры

| <code>session.in</code> | <code>session.out</code> |
|-------------------------|--------------------------|
| 1 100<br>100            | 100                      |
| 3 1200<br>50 25 25      | 600 300 300              |

## Задача В. Черно-белый Paint

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

Малыш Илюша очень сообразительный и любит познавать мир вокруг себя. Недавно он начал знакомиться с компьютерами и даже смог освоить программу «Paint», научившись рисовать надписи черным цветом в ней.

# Hello World!

Пример творчества Илюши.

У Илюши есть брат Лёша. Когда Илюша отлучается от компьютера, брат любит портить рисунки, заливая весь фон черным цветом. В итоге от символов остаются лишь замкнутые внутренние области.



Пример творчества Лёши.

Любой малыш на месте Илюши расстроился бы, если его произведения были бы испорчены, но Илюша, как уже говорилось, очень сообразительный, и решил использовать проступки брата с пользой! После очередного вандализма Лёши рисунок представляет из себя набор областей белого цвета. В зависимости от надписи, которая была сделана Илюшей, количество областей может быть разным. Польза же здесь в том, что Илюше нравится считать и узнавать новые числа, являющиеся суммой белых областей на рисунке, испорченном Лёшей.

А чтобы проверить правильность своего счёта, он попросил Вас написать программу, которая по заданному набору символов вычисляет количество замкнутых областей внутри них.

### Формат входного файла

Входной файл состоит из одной строки, содержащей большие и маленькие латинские символы, цифры, пробелы и знаки пунктуации (: ; . , ! ? -). Строка непустая и ее длина не превышает 99999 символов. Строка во входном файле всегда оканчивается символом конца строки. Можно считать, что буквы выглядят так: AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz, а цифры так: 0123456789.

### Формат выходного файла

В выходной файл выведите одно число — количество замкнутых внутренних областей во введённой строке.

### Примеры

| <code>paint.in</code>                          | <code>paint.out</code> |
|------------------------------------------------|------------------------|
| Hello World!                                   | 4                      |
| F - 0, Q - 1, g - 2.                           | 4                      |
| Ololo pysch pysch!!!                           | 5                      |
| Ilya is 2 years old, Lesha is 2 years old too. | 13                     |
| Black-white Paint                              | 6                      |

## Задача С. Видеодрайвер

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

Чёрный Кот обожает всем известную свободную операционную систему. К сожалению, драйверы для многих редких устройств в этой системе отсутствуют, поэтому Кот пишет их сам. Недавно Кот обнаружил в шкафу старый компьютер. Конечно же, первым делом он попробовал установить туда свою любимую операционную систему, но его ждала неудача — видеокарта этого древнего компьютера поддерживает только текстовый режим экрана, а установщик системы работает в графическом режиме. Кот не сдавался и уже через некоторое время экран компьютера радовал его приветствием `forest:~#`.

Кот решил немного доработать видеодрайвер, чтобы иметь возможность рисовать окна при помощи символов псевдографики. Однако в знакогенераторе видеокарты обнаружились только символы `|` (ASCII-код 124), `+` (ASCII-код 43) и `-` (ASCII-код 45). Кот решил, что для начала этого будет достаточно и он ограничится только вертикальными и горизонтальными линиями.

Вам требуется помочь Коту написать фрагмент драйвера, который по набору выводимых на экран вертикальных и горизонтальных линий будет генерировать матрицу символов для видеокарты. Матрица символов генерируется по следующим правилам:

1. Каждая линия задаётся позициями двух символов в матрице. Для горизонтальной линии эти символы должны быть расположены в одной строке, для вертикальной — в одном столбце. Эти символы будем называть начальным и конечным соответственно.
2. Горизонтальная линия «проходит» через все символы в строке, расположенные между начальным и конечным (включительно).
3. Вертикальная линия «проходит» через все символы в столбце, расположенные между начальным и конечным (также включительно).
4. Если через символ «проходит» горизонтальная линия, в матрице символов для видеокарты требуется выдать символ `-`.
5. Если через символ «проходит» вертикальная линия, в матрице символов для видеокарты требуется выдать символ `|`.
6. Если через символ «проходит» и горизонтальная, и вертикальная линия. В матрице символов для видеокарты требуется выдать символ `+`.
7. Если через символ не «проходит» ни одна линия, в матрице символов для видеокарты требуется вывести пробел.

### Формат входного файла

В первой строке входного файла находится единственное число  $N$  — количество линий (горизонтальных и вертикальных) ( $1 \leq N \leq 10^4$ ). Во второй строке находятся два числа  $R$  и  $C$  — количество строк и столбцов в матрице символов видеокарты ( $1 \leq R, C \leq 100$ ). Последующие  $N$  строк содержат по четыре числа  $r_1, c_1, r_2, c_2$ , описывающие выводимые на экран линии ( $1 \leq r_1, r_2 \leq R, 1 \leq c_1, c_2 \leq C$ ),  $r_1, c_1$  — номера строки и столбца начального символа линии,  $r_2, c_2$  — соответственно номера строки и столбца конечного символа линии. Гарантируется, что каждая линия является либо горизонтальной (в таком случае  $r_1 = r_2$ ), либо вертикальной (в таком случае  $c_1 = c_2$ ). Гарантируется, что начальная и конечная точки не совпадают.

### Формат выходного файла

Выведите  $R$  строк по  $C$  символов в каждой — искомую матрицу символов.

## Примеры

| driver.in | driver.out |
|-----------|------------|
| 4         | +---+      |
| 5 5       | ...        |
| 1 1 1 5   | ...        |
| 1 1 5 1   | ...        |
| 1 5 5 5   | +---+      |
| 5 1 5 5   |            |

Примечание: в примере пробелы заменены на точки для большей наглядности. Вам следует выводить именно пробелы. Позаботьтесь о том, чтобы ни в какой строке не было бы лишних пробелов.

## Задача D. Тёплый шарф

Имя входного файла: `scarf.in`  
Имя выходного файла: `scarf.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Малыш Илюша очень любит тепло одеваться. Так получилось, что сегодня у него День рождения и мама подарила ему тёплый шарф длиной  $K$  сантиметров и толщиной  $d$  сантиметров. Теперь Илюшу очень интересует вопрос: какое максимальное число раз он сможет полностью обмотать шарф вокруг своей шеи, представляющей из себя окружность радиуса  $R$  сантиметров, и какой длины останется не намотанная часть.

При подсчете оборотов шарфа вокруг шеи нужно считать, что шея длиннее ширины шарфа, а после полного оборота шарфа вокруг шеи ее радиус просто увеличивается на толщину шарфа, а шарф начинает наматываться заново на шею нового радиуса. Шарф не может растягиваться, однако может сжиматься. Таким образом при самом первом обороте шарфа вокруг шеи, на полный оборот будет израсходована длина шарфа, соответствующая радиусу  $R + d$ , а не радиусу  $R$ .

Так как Илюша ещё слишком маленький, и у него не получается аккуратно намотать шарф, он попросил Вас ему помочь.

### Формат входного файла

Входной файл состоит из одной строки, содержащей три вещественных числа, разделённых пробелами:  $K, d, R$  ( $1 \leq K \leq 10^{13}, 0.000001 \leq d \leq 10^6, 1 \leq R \leq 10^6$ ) — длина шарфа, толщина шарфа и радиус шеи. Все размеры даны в сантиметрах и имеют не более чем 6 знаков после запятой.

### Формат выходного файла

В выходной файл выведите два числа разделённых пробелом — максимальное число полных оборотов шарфа вокруг шеи и длину остатка шарфа в сантиметрах с точностью не менее 8 знаков после запятой. Ваш ответ будет признан правильным, если относительная или абсолютная погрешность не превышает  $10^{-6}$ .

### Примеры

| <code>scarf.in</code>        | <code>scarf.out</code> |
|------------------------------|------------------------|
| 110 1 13                     | 1 22.0354056995        |
| 10 0.001 1                   | 1 3.7105315075         |
| 17.342 0.00342 1.14501       | 2 2.8889145016         |
| 123.456789 0.123456 1.234567 | 10 3.2233257581        |
| 31.415927 2 3                | 1 0.0000004641         |

Примечание: решения, работающие правильно при ( $1 \leq K \leq 10^8, 0.001 \leq d \leq 10^6, 1 \leq R \leq 10^6$ ) будут оцениваться исходя из 50 баллов.

## Задача Е. Мониторы

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

Чёрный Кот подрабатывает созданием сайтов на заказ. Одной из необходимых операций для создания сайта является вёрстка макета — когда макет дизайна, нарисованный в графическом редакторе, переводят в HTML, чтобы потом на основе этого HTML-документа создать шаблон для системы управления содержимым сайта.

Конечно, при вёрстке HTML-шаблона, Кот проверяет правильность его отображения в различных условиях: в разных браузерах, с разным разрешением экрана и т.п. Обычно всё проходит достаточно гладко, однако в этот раз заказчик хочет, чтобы сайт корректно отображался на очень широких мониторах (более 2000 пикселей по горизонтали). Однако у Кота нет такого широкого монитора и казалось бы, он не сможет протестировать сайт. Но Кот не был бы Котом, если бы не придумал решение. Он понял, что если подключить к компьютеру несколько мониторов и расположить их по горизонтали, можно получить достаточно широкий виртуальный экран, на котором уже можно будет протестировать сайт. Кот выписал разрешения всех мониторов, которые есть у него и его друзей и теперь хочет узнать, какое же максимальное разрешение по горизонтали он может получить, используя эти мониторы.

Кот может объединять только мониторы, имеющие одинаковое разрешение по вертикали, поскольку в противном случае работать будет очень неудобно. Также Кот не может менять разрешение монитора, поскольку они все жидкокристаллические и при изменении разрешения с оптимального, изображение сильно портится.

### Формат входного файла

В первой строке входного файла находится единственное число  $N$  — количество мониторов, имеющихся у Кота и его друзей ( $1 \leq N \leq 100000$ ). Далее в  $N$  строках следует по два числа  $w_i, h_i$  — разрешение очередного монитора по горизонтали и вертикали соответственно ( $1 \leq w_i, h_i \leq 10^9$ ).

### Формат выходного файла

Выведите единственное число — максимальное разрешение по горизонтали, которое может получить Кот используя имеющиеся мониторы.

### Примеры

| <code>monitors.in</code>                            | <code>monitors.out</code> |
|-----------------------------------------------------|---------------------------|
| 3<br>1024 768<br>1024 768<br>1600 1200              | 2048                      |
| 4<br>1024 768<br>1024 768<br>1600 1200<br>1600 1200 | 3200                      |
| 3<br>640 480<br>800 600<br>400 768                  | 800                       |

Примечание: в первом примере Коту следует объединить два первых монитора. Третий монитор добавить к ним нельзя, поскольку у него другое разрешение по вертикали. Во втором примере можно объединить либо два первых монитора, либо два последних. Два последних объединить выгоднее.

В третьем примере никакие два монитора объединить нельзя — Коту придётся просто использовать самый широкий монитор.

Решения, работающие при  $N \leq 1000$  будут оцениваться из 50 баллов.

Решения, работающие при  $h_i, w_i \leq 1000$  будут оцениваться из 50 баллов.

Решения, работающие в случае, когда все  $h_i$  различны, будут оцениваться из 20 баллов. Учтите, что в этом случае всё равно требуется пройти тесты из условия.