

## Задача А. Произведение цифр

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

По заданным  $n, k$  найдите количество чисел от 1 до  $n$  с произведением цифр не более  $k$ .

### Формат входных данных

В первой и единственной строке ввода находятся два целых числа  $n$  и  $k$  ( $1 \leq n \leq 10^{18}, 1 \leq k \leq 10^9$ ).

### Формат выходных данных

Выведите единственное число — ответ на задачу.

### Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 13 2             | 5                 |
| 100 80           | 99                |

### Замечание

Интересно, а сколько различных произведений бывает.

## Задача В. Путь на Манхеттене

Имя входного файла: стандартный ввод  
 Имя выходного файла: стандартный вывод  
 Ограничение по времени: 0.6 секунд  
 Ограничение по памяти: 256 мегабайт

Пусть  $dist(A, B) = |A_x - B_x| + |A_y - B_y|$ .

Дано  $n$  точек. Ваша задача — обойти их все **по порядку** кроме, может быть, одной точки.

Есть  $q$  запросов. Каждый из них имеет вид `update i x y`. Запрос означает, что  $i$ -я точка теперь имеет координаты  $(x, y)$ .

Найдите наименьшее расстояние, которое вам придется пройти в каждом из случаев.

### Формат входных данных

В первой строке находится число  $n$ . ( $1 \leq n \leq 10^5$ ).

В следующих  $n$  строках находится по 2 числа — координаты точек.

В  $(n + 2)$ -й строке находится число  $q$ . ( $1 \leq q \leq 10^5$ ).

В следующих  $q$  строках находятся сами запросы.

### Формат выходных данных

После каждого изменения выведите ответ на задачу.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3                | 2                 |
| 0 0              | 2                 |
| 2 2              | 0                 |
| 2 0              |                   |
| 3                |                   |
| update 2 1 100   |                   |
| update 2 2 2     |                   |
| update 3 0 0     |                   |

## Задача С. Комбинации в видеоигре

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

Бесси играет в компьютерную игру! В игре есть только три допустимые кнопки:  $A$ ,  $B$  и  $C$ . Бесси может нажимать кнопки в любом порядке, но есть только  $N$  различных последовательностей нажатия ( $1 \leq N \leq 20$ ), за которые дают баллы. Комбинация  $i$  представлена строкой  $S_i$  длиной от 1 до 15 символов, содержащей только буквы  $A$ ,  $B$  и  $C$ .

Когда Бесси нажимает кнопки в одной из этих последовательностей, она получает одно очко за эту последовательность. Последовательности могут перекрываться или заканчиваться одновременно! Например, если  $N = 3$  и три балловые последовательности —  $ABA$ ,  $CB$  и  $ABACB$ , а Бесси нажимает  $ABACB$ , она получит 3 очка. Бесси может получить очки за одну последовательность несколько раз.

Конечно же, Бесси хочет заработать как можно больше очков. Если она нажимает ровно  $K$  кнопок ( $1 \leq K \leq 1000$ ), какое максимальное количество очков она может заработать?

### Формат входных данных

В первой строке находятся два целых числа, разделенных пробелом:  $N$  и  $K$ .

В каждой из следующих  $N$  строк находится очередная балловая последовательность.

### Формат выходных данных

В единственной строке выведите одно целое число — максимальное количество очков, которое Бесси может получить.

### Пример

| стандартный ввод          | стандартный вывод |
|---------------------------|-------------------|
| 3 7<br>ABA<br>CB<br>ABACB | 4                 |

### Замечание

Оптимальная последовательность нажатий кнопок в этом случае —  $ABACB$ , что дает 4 очка — 1 за  $ABA$ , 1 за  $ABACB$  и 2 за  $CB$ .

## Задача D. Маршрут доставки

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

После нескольких лет рекордного производства молока, Фермер Джон теперь управляет целой сетью из  $N$  ферм ( $1 \leq N \leq 100$ ). Ферма  $i$  находится в позиции  $(x_i, y_i)$  на плоскости, отличной от всех остальных ферм, где  $x_i$  и  $y_i$  - целые числа.

ФД нуждается в вашей помощи в планировании его ежедневного маршрута доставки для поставки товаров на  $N$  ферм. Начиная с фермы 1, он планирует посещать фермы последовательно (ферма 1, затем ферма 2, затем ферма 3 и т. д.), наконец, возвращаясь на ферму 1 после посещения фермы  $N$ . ФД требуется одна минута, чтобы сделать один шаг на север, юг, восток или запад. Кроме того, ФД хочет посетить каждую ферму ровно один раз во время всего своего путешествия (кроме фермы 1, которую он посещает дважды).

Пожалуйста, помогите ФД определить минимальное количество времени, которое ему потребуется, чтобы завершить весь маршрут.

### Формат входных данных

В первой строке находится единственное число  $N$ .

В следующих  $N$  строках находятся по два числа  $x_i$  и  $y_i$  ( $1 \leq x_i, y_i \leq 1,000,000$ ).

### Формат выходных данных

Выведите минимальное количество минут, которое ФД нужно для завершения его маршрута доставки, или -1, если подходящего маршрута не существует.

### Пример

| стандартный ввод              | стандартный вывод |
|-------------------------------|-------------------|
| 4<br>2 2<br>2 4<br>2 1<br>1 3 | 12                |

## Задача Е. Дерево? Дерево!

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

Дано дерево на  $N$  вершинах с весами на ребрах, подвешенное за вершину 1.  
Для каждой вершины дерева  $v$  найдите количество его вершин  $u$  таких, что:

- $dist(u, v) \leq L$
- все вершины на пути из  $v$  в  $u$  лежат дальше от корня дерева, чем  $v$ .

### Формат входных данных

В первой строке находятся два числа  $N$  и  $L$ ,  $1 \leq N \leq 200000$ ,  $1 \leq L \leq 10^{18}$ .

Следующие  $N - 1$  строк содержат информацию о ребрах дерева. Каждая строка содержит пару чисел  $p_{i+1}, l_{i+1}$  — предка вершины  $(i + 1)$  и длина ребра из неё в предка.  $1 \leq p_i \leq i$ ,  $1 \leq l_{i+1} \leq 10^{12}$ .

### Формат выходных данных

Выведите  $N$  чисел — ответ на задачу.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 4 5              | 3                 |
| 1 4              | 2                 |
| 2 3              | 1                 |
| 1 5              | 1                 |

## Задача F. Аппарат Пина

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

После инцидента с морковкой Крош стал избегать азартных игр. Но недавно Пин построил совсем новый аппарат, в этот раз максимально честный и с понятными правилами. Аппарат принимает массив натуральных чисел, затем вычисляет побитовое **and** всех чисел и в качестве приза выдает столько конфет, сколько показывает результат. Если на вход дали пустой массив, то аппарат не дает конфет. Само собой, Крош не мог пропустить возможность получить много конфет. Для этого он нашел дома массив, который ему подарили на прошлый День Рождения. Но, как хороший друг, Крош решил отдать несколько элементов массива Ёжику, чтобы он тоже смог получить много конфет. Также Крош хочет, чтобы каждый элемент попал в аппарат. Но понять, какие элементы нужно отдать, оказалось непростой задачей, помогите им решить этот вопрос, чтобы друзья смогли выиграть вместе как можно больше конфет.

Побитовое **and** — бинарная операция, действие которой эквивалентно применению логического «И» к каждой паре битов, которые стоят на одинаковых позициях в двоичных представлениях операндов. Например, побитовое **and** 13 и 6 = 4. Данная операция существует во всех современных языках программирования, например, в языках C++, Java и Python она обозначена как «&», в Pascal — как «and».

### Формат входных данных

В первой строке вводится число  $n$  — количество элементов в массиве Кроша ( $2 \leq n \leq 10^5$ ).

Во второй строке вводится  $n$  целых чисел — массив  $a_1, a_2, \dots, a_n$ , который подарили Крошу ( $0 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите, какое максимальное суммарное количество могут получить друзья, после того как Крош отдаст некоторые элементы Ёжику.

### Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 4<br>3 1 5 2     | 5                 |
| 3<br>5 5 2       | 7                 |

### Замечание

Обозначим за  $\wedge$  операцию побитового **and**.

В первом тесте Крош может отдать Ёжику элементы 1, 3, 2, тогда аппарат Крошу даст 5 конфет, а Ёжику  $1 \wedge 3 \wedge 2 = 0$  конфет, суммарно они получают 5 конфет.

Во втором тесте Крош может отдать Ёжику элементы 5, 5, тогда аппарат Крошу даст 2 конфет, а Ёжику  $5 \wedge 5 = 5$  конфет, суммарно они получают 7 конфет.

## Задача G. Разговор с бабой Ниной

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

Ночью Михаил разговаривал с бабой Ниной по телефону, и неожиданно она сказала, чтобы продолжить разговор реши задачку: "Дано  $n$  выделите в клике размера  $n$  максимальное кол-во реберно непересекающихся остовных деревьев"

Михаил в час ночи совсем не может решать задачи, помогите ему пж.

### Формат входных данных

Первая строка входных данных содержит натуральное число  $n \leq 1000$

### Формат выходных данных

выведите  $k$  - число деревьев В  $i$ -й последовательности строк – описании  $i$ -го остовного дерева – перечислите все ребра, которые входят в это дерево. Каждое ребро  $v_i$  и  $u_i$  опишите в отдельной строке в формате " $u_i v_i$ " (без кавычек). Последовательности строк, относящиеся к разным планам, разделите пустыми строками

### Примеры

| стандартный ввод | стандартный вывод                               |
|------------------|---|
| 2                | 1<br>1 2  |
| 4                | 2<br>1 2<br>1 3<br>2 4<br><br>3 4<br>1 4<br>2 3 |

## Задача Н. Шахматы

|                         |                   |
|-------------------------|-------------------|
| Имя входного файла:     | стандартный ввод  |
| Имя выходного файла:    | стандартный вывод |
| Ограничение по времени: | 2 секунды         |
| Ограничение по памяти:  | 256 мегабайт      |

Однажды мальчик Петя стал чемпионом Вселенной по шахматам, и он решил, что эта игра слишком простая для него. Поэтому он решил играть не на поле  $8 \times 8$ , а на поле  $10^9 \times 10^9$ . Прежде чем начать всех побеждать и здесь, он решил начать с чего-то простого. Для начала он расставил  $N$  пешек на этом поле. Затем он решил  $Q$  раз **представить** (то есть один запрос не влияет на все последующие), что на клетке  $(x, y)$  стоит одна из шахматных фигур и узнать, сколько пешек она бьет. Он, конечно, с этой задачей справился, а вы справитесь?

Обратите внимание как ходят фигуры.

- Фигура *слон* бьет все пешки, которые с ней находятся на одной диагонали.
- Фигура *ладья* бьет все пешки, которые с ней находятся в одной строке или в одном столбце.
- Фигура *конь* бьет все пешки, которые находятся либо в двух клетках по горизонтали и одной по вертикали, либо в двух клетках по вертикали и одной по горизонтали.
- Фигура *пешка* бьет все пешки, которые находятся в следующей строке и в соседнем столбце.
- Фигура *ферзь* бьет одновременно как слон и ладья.

Важно, что фигуры могут бить пешки **насквозь**, то есть, если на пути от фигуры до пешки стоит другая пешка, то она все равно бьет первую пешку.

### Формат входных данных

В первой строке вводится целое число  $N$  ( $1 \leq N \leq 10^5$ ) — количество пешек на поле.

В следующих  $N$  строках находятся по два целых числа  $x_i, y_i$  ( $1 \leq x_i, y_i \leq 10^9$ ) — координаты клетки, где находится  $i$ -я пешка. **Гарантируется**, что нет двух пешек, которые находятся в одной клетке. Клетка с координатами  $(1, 1)$  соответствует нижнему левому углу поля. Ось  $Ox$  направлена вдоль нижней стороны поля, ось  $Oy$  направлена вдоль левой стороны поля.

В следующей строке вводится количество запросов  $Q$  ( $1 \leq Q \leq 10^5$ ).

В  $i$ -й из следующих  $Q$  строк находится  $i$ -й запрос. Он состоит из символа  $t$  и двух целых чисел  $x$  и  $y$  ( $1 \leq x, y \leq 10^9$ ), где  $x, y$  — координаты, на которые мы предположительно ставим фигуру, а  $t$  — заглавная буква английского алфавита, описывающая тип фигуры ( $K$  — конь,  $P$  — пешка,  $B$  — слон,  $Q$  — ферзь,  $R$  — ладья). **Гарантируется**, что в клетке  $(x, y)$  изначально не стоит пешка.

### Формат выходных данных

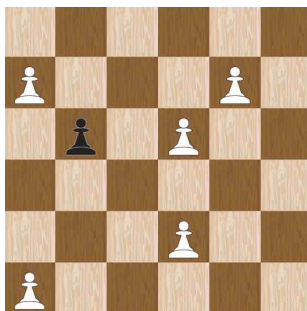
Для каждого запроса выведите количество пешек, которые будет бить фигура из запроса.

### Пример

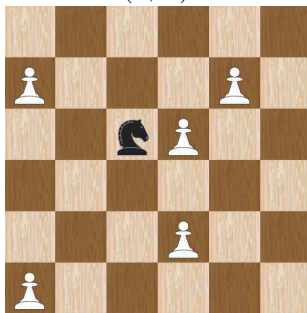
| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 5                | 1                 |
| 1 1              | 3                 |
| 1 5              | 5                 |
| 4 2              | 3                 |
| 5 5              | 3                 |
| 4 4              |                   |
| 5                |                   |
| P 2 4            |                   |
| K 3 4            |                   |
| Q 3 3            |                   |
| R 1 2            |                   |
| B 2 2            |                   |



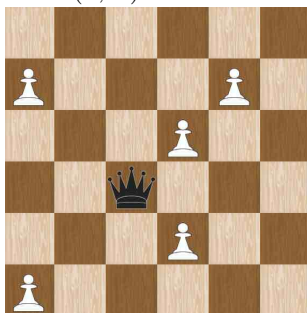
## Замечание



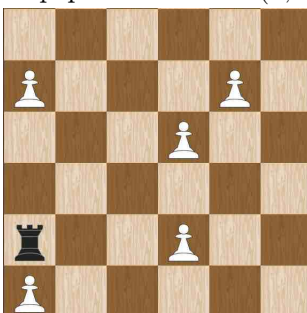
В первом запросе пешка на клетке (2, 4) бьёт только пешку на клетке (1, 5)



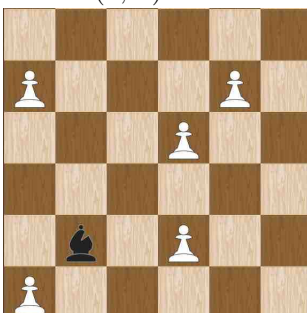
Во втором запросе конь на клетке (3, 4) бьёт пешки на клетках (1, 5), (5, 5), (4, 2)



В третьем запросе ферзь на клетке (3, 3) бьёт все пешки



В четвертом запросе ладья на клетке (1, 2) бьет пешки на клетках (1, 1), (1, 5), (4, 2)



В пятом запросе слон на клетке (2, 2) бьет пешки на клетках (1, 1), (4, 4), (5, 5)

## Задача I. Бамбук Гомори-Ху

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

Добрый Бобр съездил в ЛКШ.зима 2022-2023 и узнал про дерево Гомори-Ху. Оказывается для любого неориентированного графа можно построить дерево с таким же множеством вершин, такое что размер максимального потока между любыми двумя вершинами исходного графа совпадает с максимальным потоком между этими же вершинами в дереве. Бобр был в шоке от такого алгоритма, но вскоре понял, что ему этого мало. Он хочет, чтобы дерево к тому же было бамбуком.

Бобр не очень трудолюбивый, поэтому он не хочет писать новый код. Он уже сдал в констест учебную задачу на Гомори-Ху, поэтому код, который строит дерево по графу у него уже есть. Теперь он хочет дописать алгоритм, который по дереву построит бамбук, в котором максимальный поток между любой парой вершин совпадает с максимальным потоком между этими вершинами в дереве. Сейчас бобр ушел играть в доту, поэтому код придется писать вам.

### Формат входных данных

В первой строке входных данных содержится целое число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — количество вершин в дереве.

В каждой из следующих  $n - 1$  строк содержится по три целых числа  $u, v, w$  ( $1 \leq u, v \leq n$ ,  $1 \leq w \leq 10^9$ ), которые означают, что есть ребро между  $u$  и  $v$  пропускной способности  $w$ .

Гарантируется, что эти ребра образуют дерево. **Гарантируются, что  $w_i$  различны.**

### Формат выходных данных

В первой строке выходных данных выведите перестановку  $p_1, p_2, \dots, p_n$  — последовательность вершин в бамбуке.

Во второй строке выведите последовательность пропускных способностей  $w_1, w_2, \dots, w_{n-1}$  — последовательность пропускных способностей ребер бамбука. Ребро пропускной способности  $w_i$  соединяет вершины  $p_i$  и  $p_{i+1}$ .

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 4                | 1 4 3 2           |
| 1 2 1            | 3 2 1             |
| 1 3 2            |                   |
| 1 4 3            |                   |

## Задача J. короткая задача

Имя входного файла: стандартный ввод  
 Имя выходного файла: стандартный вывод  
 Ограничение по времени: 6 секунд  
 Ограничение по памяти: 1024 мегабайта

Вам дано целое число  $n$ , массив целых чисел  $a_1, a_2, \dots, a_n$  и простое число  $p$ . Найдите кол-во пар  $i, j$  таких, что существует натуральное число  $k$ , такое что  $a_i^k = a_j \pmod{p}$

### Формат входных данных

первая строка содержит 2 числа  $n$  ( $2 \leq n \leq 2 \cdot 100000$ ) и  $p$  ( $2 \leq p \leq 10^{13}$ ) вторая строка содержит  $n$  чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq p$ )

### Формат выходных данных

выведите 1 число - ответ на задачу

### Примеры

| стандартный ввод  | стандартный вывод |
|---|-------------------|
| 5 2<br>1 1 1 1 1  | 25                |
| 3 13<br>2 3 5   | 5                 |
| 10 999999999971<br>141592653589 793238462643 383279502884<br><br>197169399375<br><br>105820974944 592307816406 286208998628<br><br>34825342117 67982148086 513282306647 | 63                |