

А. Чёрные начинают и...

Вам задано расположение белого короля и белой ладьи, а также чёрного короля на шахматной доске. Расположение фигуры задаётся двумя символами — буквой от “a” до “h”, задающей вертикаль, и цифрой от 1 до 8, задающей горизонталь. В данный момент очередь хода за чёрными.

Напомним, что ладья атакует все поля, находящиеся с ней на одной горизонтали или вертикали, кроме случаев, когда между данным полем и ладьёй в соответствующем направлении есть другая фигура. Король атакует все поля, соседние с ним по стороне или по углу и может делать ход (как со взятием занимающей его фигуры другого цвета, так и без) на любое из таких полей, которое не находится под атакой фигуры другого цвета.

Требуется классифицировать позицию в соответствии со следующими вариантами:

- странный — чёрный король находится под атакой белого,
- обычная позиция — чёрный король не находится под атакой и может сделать ход,
- шах — чёрный король находится под атакой и может сделать ход,
- пат — чёрный король не находится под атакой и не может сделать ход,
- мат — чёрный король находится под атакой и не может сделать ход.

Формат ввода

В первой строке заданы три пары символов. Первый символ каждой пары — буква от “a” до “h”, второй — цифра от 1 до 8. Эти три пары задают позиции белого короля, белой ладьи и чёрного короля соответственно. Соседние пары символов разделены пробелами. Гарантируется, что никакие две фигуры не занимают одну и ту же позицию.

Формат вывода

В случае, если позиция является странной, выведите “Strange”, в случае, если обычной — “Normal”, в случае, если является шахом — “Check”, если является патом — “Stalemate”, если является матом — “Checkmate”.

Примеры

Ввод

b2 a3 c4

a1 b2 e2

Вывод

Normal

Check

В. Байтландские шаманы

Шаманы племён, обитавших в древности на территории нынешней Байтландии, практиковали следующий способ построения заклинаний.

В основу брались две ключевых фразы — A и B . Каждая фраза могла состоять из нескольких слов. После этого строилась такая последовательность заклинаний: $T_0 = A$, $T_1 = B$, $T_2 = A B$, $T_3 = B A B$, ..., $T_n = T_{n-2} T_{n-1}$.

Таким образом, каждое заклинание T_i , начиная с $i = 2$, состояло из последовательного исполнения заклинаний T_{i-2} и T_{i-1} . При последовательном исполнении заклинаний между ними делалась небольшая пауза, так что новые слова при конкатенации образовываться не могли.

В разных случаях использовались разные заклинания: например, для проведения обряда инициации шаман должен был произнести заклинание так, чтобы имя духа S , встречающееся как минимум в одной из ключевых фраз, было произнесено ровно K раз.

Антропологи обратились к вам с просьбой: зная начальные ключевые фразы A и B , имя духа S , встречающееся в одной или обеих из них, а также число K , определить число N — наименьший номер заклинания T_N , в котором слово S встречается ровно K раз.

Формат ввода

Первая строка содержит ключевую фразу A — строку из не более 255 ASCII-символов с кодами от 32 до 127 включительно. Фраза содержит от одного до десяти слов.

Словом считается последовательность английских букв, ограниченная символами-разделителями, к которым относятся все остальные символы. При этом большие и маленькие английские буквы не различаются.

Во второй строке в том же формате задана фраза B .

В третьей строке задано имя духа S — слово, представляющее собой последовательность маленьких английских букв. Гарантируется, что слово S содержится как минимум в одной из ключевых фраз A или B .

Четвёртая строка содержит одно целое положительное число K . Длина строки, содержащей число K , не превосходит 3000 символов.

Формат вывода

Требуется вывести единственное число N — наименьший номер заклинания, в котором слово S встречается ровно K раз. Если такого номера не существует, следует вывести "Impossible".

Ввод

```
Python is great!  
Python-Die-Grosse-Shlange  
Python  
5
```

Вывод

4

```
Great Python, answer to me!  
Help us, Python  
python  
4
```

Impossible

С. Окружности-2

Задано n треугольников, длины сторон которых являются целыми числами. Найдите наибольшее количество треугольников, имеющих один и тот же радиус вписанной окружности.

Формат ввода

В первой строке входного файла задано целое число n ($1 \leq n \leq 1000$) — количество треугольников. Каждая из последующих n строк содержит три целых числа a_i , b_i и c_i — длины сторон i -го треугольника. Гарантируется, что треугольники невырождены и $1 \leq a_i, b_i, c_i \leq 1000$.

Формат вывода

Выведите одно целое число — количество треугольников, имеющих наиболее часто встречающийся в наборе радиус вписанной окружности.

Пример

Ввод

```
3
3 4 5
5 4 3
10 10 10
```

Вывод

```
2
```

Д. После финала

Финал Чемпионата Байтландии по программированию впервые проводился в городе $\mathcal{I}$. Дорожная сеть города $\mathcal{I}$ представляет собой N перекрёстков, соединённых M дорогами с двусторонним движением. Программа финала была столь насыщенной, что участники не успели осмотреть город. Более того, церемония закрытия затянулась, так что при отъезде команды-победителя времени оставалось только на то, чтобы добраться до аэропорта кратчайшим по суммарной длине путём. При этом, чтобы хотя бы немного посмотреть город, из всех таких маршрутов выбрали тот, который включает в себя целиком наибольшее количество дорог.

По заданной карте города вычислите длину маршрута и количество дорог, которое удалось посмотреть команде.

Формат ввода

Первая строка содержит два целых числа N и M — количество перекрёстков в городе $\mathcal{I}$ и количество дорог соответственно ($2 \leq N \leq 1000$, $1 \leq M \leq 2 \cdot 10^5$).

Каждая из следующих M строк содержит три целых числа a_i , b_i и c_i ($1 \leq a_i, b_i \leq N$, $a_i \neq b_i$, $1 \leq c_i \leq 1000$) — номера перекрёстков, соединяемых i -й дорогой, и длину этой дороги. Два различных перекрёстка могут быть соединены более чем одной дорогой. Место проведения финала находится на перекрёстке с номером 1 , аэропорт — на перекрёстке с номером N .

Гарантируется, что существует хотя бы один путь от места проведения финала в аэропорт.

Формат вывода

Выведите два целых числа P и Q — длину кратчайшего пути от места проведения финала в аэропорт и максимальное число дорог, которые может включать в себя кратчайший путь.

Пример

Ввод

```
3 3
1 2 1
1 3 2
2 3 1
```

Вывод

```
2 2
```

Е. Факториалы

Миша недавно узнал, что факториал числа n — это произведение всех натуральных чисел от 1 до n включительно. Факториал числа n имеет обозначение $n!$; так, например, $1! = 1$, $2! = 1 \cdot 2 = 2$, $3! = 1 \cdot 2 \cdot 3 = 6$. По определению $0! = 1$.

Мише понравилось вычислять факториалы маленьких чисел. Однако он быстро понял, что факториалы чисел, больших десяти, очень сложно вычислять вручную. Тогда он стал искать лишь несколько последних цифр, но обнаружил, что они слишком часто оказываются нулями: к примеру, в числе $25!$, если верить его подсчётам, последние шесть цифр — нули. Гораздо более интересной задачей оказалось найти последнюю ненулевую цифру $n!$, но Миша справился и с ней.

Руководитель математического кружка, в котором учится Миша, узнал о его успехах и дал ему более сложную задачу: найти последнюю ненулевую цифру суммы k -х степеней факториалов целых чисел от 0 до n , где k — неотрицательное целое число. Правда, руководитель сделал скидку на то, что Миша ещё учится в шестом классе, и поставил ограничение $k \leq 3$, $n \leq 100$. Впрочем, считать Мише всё равно придётся очень долго...

Помогите Мише с помощью компьютера решить поставленную задачу. Напишите программу, которая по заданным значениям n и k вычисляет последнюю ненулевую цифру суммы k -х степеней факториалов всех целых чисел в промежутке от 0 до n .

Формат ввода

В первой строке заданы два целых числа — положительное число n , не превосходящее 100, и неотрицательное число k , не превосходящее трёх.

Формат вывода

Выведите одно число — последнюю ненулевую цифру суммы k -х степеней факториалов целых чисел от 0 до n включительно.

Пример 1

Ввод

4 1

Вывод

4

Пример 2

Ввод

3 2

Вывод

2

Г. Противоположная цепная дробь

Конечная цепная дробь — это последовательность вида $[a_0; a_1, a_2, \dots, a_n]$. На элементы цепной дроби и её размер накладываются следующие ограничения:

- n — неотрицательное конечное целое число,
- элементы $a_0, a_1, a_2, \dots, a_n$ — целые числа,
- $a_i > 0$ при $i > 0$,
- $a_n > 1$, если $n > 0$.

Эти ограничения позволяют установить взаимно однозначное соответствие между рациональными числами и конечными цепными дробями: каждому вещественному числу x соответствует единственная цепная дробь $[a_0; a_1, a_2, \dots, a_n]$ такая, что

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\ddots + \frac{1}{a_n}}}}.$$

Для обозначения соответствия используется знак равенства: $x = [a_0; a_1, a_2, \dots, a_n]$.

Дано рациональное число x ($0 < x < 1$), записанное в виде цепной дроби. Требуется найти цепную дробь числа $(1 - x)$.

Формат ввода

В первой строке задано целое число n ($1 \leq n \leq 100\,000$). Во второй строке дано $(n + 1)$ целых чисел: $a_0, a_1, a_2, \dots, a_n$ — элементы исходной цепной дроби. Гарантируется, что цепная дробь задана корректно, то есть заданные числа удовлетворяют всем ограничениям из определения. Заданная цепная дробь соответствует такому рациональному числу x , что $0 < x < 1$, и для каждого i выполнено неравенство $a_i \leq 100$.

Формат вывода

В первой строке выведите целое число m , а во второй строке — $(m + 1)$ положительных целых чисел: $b_0, b_1, b_2, \dots, b_m$ — элементы полученной цепной дроби. Цепная дробь должна быть задана корректно, то есть выведенные числа должны удовлетворять всем ограничениям из определения. Для цепных дробей должно выполняться равенство

$$1 - [a_0; a_1, a_2, \dots, a_n] = [b_0; b_1, b_2, \dots, b_m].$$

Примеры

Ввод

2
0 1 2

2
0 2 3

Вывод

1
0 3

3
0 1 1 3