

Задача А. Ничего и не терялось

| | |
|-------------------------|-------------------------|
| Имя входного файла: | <code>giggle.in</code> |
| Имя выходного файла: | <code>giggle.out</code> |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

Разработка новой поисковой системы, которая ведется группой компаний *Giggle*, проходит под лозунгом «Ничего и не терялось». Вы работаете в отделе передовых разработок, и на сегодняшний день вашей задачей является разработка тестовой версии поискового «движка».

Тестовая версия реализует лишь часть функциональности полной версии. В частности, отсутствуют такие функции, как использование логических выражений в запросах, перевод найденных страниц с одного языка на другой и т.д.

Возможности тестовой версии поисковой системы ограничиваются обработкой трех видов запросов: запросов на добавление, на удаление и на поиск. Система работает следующим образом. В любой момент времени существует множество известных системе *сайтов*, причем для каждого сайта известно множество *ключевых слов*, встречающихся на нем.

Запрос на добавление содержит ключевое слово и название сайта. При его выполнении ключевое слово добавляется в множество ключевых слов, присутствующих на данном сайте. Если этого слова в соответствующем множестве еще нет, то результатом запроса является `OK`, в противном случае — `Already exists`.

Запрос на удаление содержит ключевое слово и название сайта. При его выполнении ключевое слово исключается из множества ключевых слов, присутствующих на данном сайте. Если этого слова в соответствующем множестве нет, то результатом запроса является `Not found`, в противном случае — `OK`.

Запрос на поиск содержит только ключевое слово. Результатом запроса является лексикографически отсортированный список сайтов, содержащих данное ключевое слово. При этом в результат выводятся только первые 10 сайтов из этого списка.

Задан список запросов. Необходимо вывести результат их последовательного выполнения.

Формат входного файла

Первая строка входного файла содержит целое число n — количество запросов, которые необходимо обработать ($0 \leq n \leq 2500$). Каждая из последующих n строк содержит запрос.

Запрос на добавление имеет следующий формат:

`Add keyword "<keyword>" to <site>`, где `<keyword>` — ключевое слово `<site>` — название сайта, на котором появилось это ключевое слово.

Запрос на удаление имеет следующий формат:

`Remove keyword "<keyword>" from <site>`, где `<keyword>` — ключевое слово `<site>` — название сайта, на котором появилось это ключевое слово.

Запрос на поиск имеет следующий формат:

`Search "<keyword>"`, где `<keyword>` — ключевое слово.

Все ключевые слова (`<keyword>`) состоят из строчных букв латинского алфавита. Длины ключевых слов не превосходят 30 символов.

Все названия сайтов (`<site>`) состоят из строчных букв латинского алфавита, символов «косая черта» (`</>`) и точек (`<.>`). Длины названий сайтов не превосходят 100 символов.

Формат выходного файла

Для каждого запроса выведите его результат. При этом придерживайтесь формата, приведенного в примерах. Не забудьте обратить внимание на второй пример.

Результаты запросов разделяйте строкой из пяти символов «равно» (`<=>`).

Примеры

| giggle.in |
|---|
| 12 Add keyword "olympiads" to neerc.ifmo.ru/school/io Add keyword "neerc" to neerc.ifmo.ru Search "olympiads" Search "neerc" Add keyword "olympiads" to neerc.ifmo.ru Search "olympiads" Add keyword "olympiads" to neerc.ifmo.ru/school/io Remove keyword "olympiads" from neerc.ifmo.ru/school/io Search "olympiads" Remove keyword "olymp" from neerc.ifmo.ru Remove keyword "olympiads" from neerc.ifmo.ru Search "olympiads" |
| giggle.out |
| OK ===== OK ===== Results: 1 site(s) found 1) neerc.ifmo.ru/school/io ===== Results: 1 site(s) found 1) neerc.ifmo.ru ===== OK ===== Results: 2 site(s) found 1) neerc.ifmo.ru 2) neerc.ifmo.ru/school/io ===== Already exists ===== OK ===== Results: 1 site(s) found 1) neerc.ifmo.ru ===== Not found ===== OK ===== Results: 0 site(s) found |

giggle.in

```
12
Add keyword "keyword" to site01
Add keyword "keyword" to site02
Add keyword "keyword" to site03
Add keyword "keyword" to site04
Add keyword "keyword" to site05
Add keyword "keyword" to site06
Add keyword "keyword" to site07
Add keyword "keyword" to site08
Add keyword "keyword" to site09
Add keyword "keyword" to site10
Add keyword "keyword" to site11
Search "keyword"
```

giggle.out

```
OK
=====
OK
=====
OK
=====
OK
=====
OK
=====
OK
=====
OK
=====
OK
=====
OK
=====
OK
=====
OK
=====
Results: 11 site(s) found
1) site01
2) site02
3) site03
4) site04
5) site05
6) site06
7) site07
8) site08
9) site09
10) site10
```

Задача В. Мафия в городе

Имя входного файла: `mafia.in`
Имя выходного файла: `mafia.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Об этом еще никто не знает, но многие догадываются — мафия уже в городе. Поговаривают, что в планах главы мафиозного клана захват контроля над всем городом, однако поначалу он решил ограничиться захватом основных линий связи города.

В городе находятся n базовых телефонных станций, некоторые пары которых соединены двусторонними каналами связи. Для удобства, занумеруем базовые станции целыми числами от 1 до n , канал связи в этом случае задается парой чисел (u, v) — номерами станций, которые он соединяет.

Будем говорить, что канал связи (u, v) *контролируется* мафией, если захвачена либо станция u , либо станция v (либо обе).

Глава мафиозного клана хочет контролировать все каналы связи, захватив при этом как можно меньше базовых станций. Ваша задача — помочь службе безопасности телефонной компании, составив возможный план захвата и определив количество таких планов.

Формат входного файла

Первая строка входного файла содержит два целых числа: n и m ($2 \leq n \leq 18, 0 \leq m$). Каждая из последующих m строк описывает один канал связи и содержит по два целых числа: u и v ($1 \leq u, v \leq n, u \neq v$) — номера базовых станций, соединенных этим каналом связи. Любая пара станций соединена не более, чем одним каналом.

Формат выходного файла

В первой строке входного файла выведите два числа: k и c — соответственно, минимальное количество базовых станций, которые необходимо захватить для того, чтобы контролировать все каналы связи, и число способов захватить такое количество станций, так чтобы контролировать все каналы связи.

Во второй строке входного файла выведите k чисел — номера базовых станций, соответствующих одному из способов захвата.

Примеры

| <code>mafia.in</code> | <code>mafia.out</code> |
|---------------------------------|------------------------|
| 3 3 1 2 2 3 3 1 | 2 3 1 2 |
| 5 4 1 2 1 3 1 4 1 5 | 1 1 1 |

В первом примере существует три способа захватить две станции так, чтобы контролировать все каналы связи: $\{1, 2\}$, $\{1, 3\}$, $\{2, 3\}$.

Задача С. Трубы

Имя входного файла: `pipe.in`
Имя выходного файла: `pipe.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Для поддержания температуры и чистоты воды в бассейн постоянно наливают новую воду и выливают старую. Иногда это происходит одновременно. Сотрудники бассейна составили план налива/слива воды. В плане перечислены напор слива/налива воды и интервал времени, в которое он будет происходить. Вода может выливаться (а так же наливаться) одновременно из нескольких труб. Необходимо узнать объем воды в бассейна через заданное время T . Стоит заметить, что вода не может выливаться из пустого бассейна.

Формат входного файла

В первой строке записано целое число N — количество записей в плане ($1 \leq N \leq 1000$). Далее следует N строк, содержащих по три целых числа l_i , r_i и v_i — начало работы трубы, конец работы трубы и напор, с которой вода поступает в бассейн (объем воды, поступающий в бассейн за единицу времени). Если напор отрицателен, то вода вытекает из бассейна ($0 \leq l_i, r_i, |v_i| \leq 1000, l_i \leq r_i$). В последней строке записано целое число T — время, через которое необходимо определить объем воды в бассейне ($0 \leq T \leq 10000$).

Формат выходного файла

В выходной файл выведите ответ на задачу — объем воды в бассейне через время T , если изначально бассейн пуст.

Примеры

| <code>pipe.in</code> | <code>pipe.out</code> |
|--|-----------------------|
| 3 0 10 1 5 10 1 10 15 -1 100 | 10 |
| 3 0 10 1 5 10 1 10 15 -1 10 | 15 |
| 3 0 10 -10 5 10 1 10 15 1 15 | 5 |

Задача D. Несложная сортировка

Имя входного файла: `sort.in`
Имя выходного файла: `sort.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Пусть x — целое положительное число, а k — целое положительное число от 1 до 10. Пусть $s(x, k)$ равно сумме цифр числа x , представленного в системе счисления по основанию k .

Задано n чисел a_1, a_2, \dots, a_n . Необходимо вычислить последовательность b_i по формуле $b_i = s(a_i, k_1) \cdot s(a_i, k_2)$, после чего отсортировать последовательность b_i по неубыванию.

Формат входного файла

Первая строка входного файла содержит три целых числа: n, k_1, k_2 ($1 \leq n \leq 1000, 2 \leq k_1, k_2 \leq 10$).

Вторая строка входного файла содержит n целых чисел: a_i ($1 \leq a_i \leq 10^9$).

Формат выходного файла

В выходной файл выведите n чисел — b_i в требуемом порядке.

Примеры

| sort.in | sort.out |
|--|-------------------------|
| 9 10 10 1 2 3 4 5 6 7 9 8 | 1 4 9 16 25 36 49 64 81 |
| 10 2 2 1 2 4 8 16 32 64 128 256 512 | 1 1 1 1 1 1 1 1 1 1 |

Задача Е. Строки

Имя входного файла: `strings.in`
Имя выходного файла: `strings.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Определим *расстояние* между равными по длине строками S_A и S_B (обозначим $d(S_A, S_B)$) как сумму для всех $1 \leq i \leq |S_A|$ кратчайших расстояний между буквами $S_A(i)$ и $S_B(i)$ в циклически замкнутом латинском алфавите (т.е. после буквы «а» идет буква «b», ..., после буквы «z» идет «а»). Например $d(aba, aca) = 1$, а $d(aba, zbz) = 2$.

Напомним, что *циклическим сдвигом* строки S называется строка (обозначаем, как $S \leftrightarrow k$) $S_k S_{k+1} S_{k+2} \dots S_{|S|} S_1 S_2 \dots S_{k-1}$ для некоторого k , где $|S|$ — длина строки S .

Степенью циклического расстояния между строками S_A и S_B ($|S_A| = |S_B|$) называется сумма:

$$\sum_{i=1}^{|S_A|} \sum_{j=1}^{|S_B|} d(S_A \leftrightarrow i, S_B \leftrightarrow j)$$

Ваша задача посчитать *степень циклического расстояния* заданных строк S_A и S_B .

Формат входного файла

В первой и второй строке расположены две строки равной длины, не превышающей 100000 символов. Строки состоят только из маленьких букв латинского алфавита.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

| <code>strings.in</code> | <code>strings.out</code> |
|-------------------------|--------------------------|
| a b | 1 |
| ab ac | 8 |

Разберем второй пример. Все циклические сдвиги строки «ab» — «ab» и «ba». Все циклические сдвиги строки «ac» — «ac» и «ca». Искомое значение равно $d(ab, ac) + d(ab, ca) + d(ba, ac) + d(ba, ca) = 1 + 3 + 3 + 1 = 8$

Задача F. Сверхстепень

Имя входного файла: `superpow.in`
Имя выходного файла: `superpow.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Назовем значение выражения 2^{2^n} *n*-ой *сверхстепенью* числа 2.

Таким образом, например, третья сверхстепень числа 2 равна $2^{2^3} = 2^8 = 256$.

Ваша задача — вычислить *n*-ую сверхстепень двойки по модулю *m*.

Формат входного файла

Входной файл содержит два целых числа: *n* ($0 \leq n \leq 10^5$) и *m* ($2 \leq m \leq 10^4$).

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

| <code>superpow.in</code> | <code>superpow.out</code> |
|--------------------------|---------------------------|
| 3 1000 | 256 |
| 10 10 | 6 |

Задача G. От перестановки чисел что-то меняется . . .

Имя входного файла: `swap.in`
Имя выходного файла: `swap.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Всем известно, что «от перестановки слагаемых сумма не изменяется». Однако, случается и так, что перестановка двух чисел приводит к более интересным последствиям.

Пусть, например, заданы три числа: a_1, a_2, a_3 . Рассмотрим равенство $a_1 + a_2 = a_3$. Оно может быть не верным (например, если $a_1 = 1, a_2 = 4, a_3 = 3$), однако может стать верным, если поменять некоторые числа местами (например, если поменять местами a_2 и a_3 , оно обратится в равенство $1 + 3 = 4$).

Ваша задача — по заданным трем числам определить, можно ли их переставить так, чтобы сумма первых двух равнялась третьему.

Формат входного файла

Входной файл содержит три целых числа: a_1, a_2, a_3 ($-10^8 \leq a_1, a_2, a_3 \leq 10^8$).

Формат выходного файла

В выходной файл выведите слово **YES**, если заданные числа можно переставить так, чтобы сумма первых двух равнялась третьему. В противном случае выведите в выходной файл слово **NO**.

Примеры

| swap.in | swap.out |
|---------|----------|
| 3 5 2 | YES |
| 2 2 5 | NO |
| 2 2 4 | YES |

Задача Н. Треугольник

Имя входного файла: `triangle.in`
Имя выходного файла: `triangle.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Пусть на плоскости введена прямоугольная декартова система координат и задан треугольник ABC : $A(x_A, y_A)$, $B(x_B, y_B)$, $C(x_C, y_C)$. Центром масс треугольника назовем точку с координатами $((x_A + x_B + x_C)/3, (y_A + y_B + y_C)/3)$.

Заданы координаты центра масс треугольника ABC . Необходимо найти координаты вершин треугольника ABC , удовлетворяющие условиям:

- координаты вершин — целые;
- координаты вершин не превосходят по модулю 10^9 ;
- треугольник ABC имеет ненулевую площадь.

Формат входного файла

Входной файл содержит два целых числа: x и y ($-10^9 \leq x, y \leq 10^9$) — координаты заданной точки.

Формат выходного файла

В первой строке входного файла выведите слово YES, если искомый треугольник существует, и слово NO — в противном случае.

Если ответ положителен, то в последующих трех строках выведите координаты вершин искомого треугольника.

Если ответ неоднозначен, выведите любой.

Примеры

| <code>triangle.in</code> | <code>triangle.out</code> |
|--------------------------|---------------------------|
| 9 4 | YES 3 9 6 3 18 0 |