

**Задача А. Выпуклая оболочка (12,5 баллов)**

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

Дано  $N$  точек на плоскости. Нужно построить их выпуклую оболочку. Гарантируется, что выпуклая оболочка не вырождена.

**Формат входного файла**

На первой строке число  $N$  ( $3 \leq N \leq 10^5$ ). Следующие  $N$  строк содержат пары целых чисел  $x, y$  ( $-10^9 \leq x, y \leq 10^9$ ) — точки.

Будьте аккуратны! Точки произвольны: бывают совпадающие, бывают лежащие на одной прямой в большом количестве.

**Формат выходного файла**

В первой строке выведите  $N$  — число вершин выпуклой оболочки. Следующие  $N$  строк должны содержать координаты вершин в порядке обхода. Никакие три подряд идущие точки не должны лежать на одной прямой.

**Примеры**

| hull.in | hull.out |
|---------|----------|
| 5       | 4        |
| 0 0     | 0 0      |
| 2 0     | 0 2      |
| 0 2     | 2 2      |
| 1 1     | 2 0      |
| 2 2     |          |

**Задача В. Разрезание графа (12,5 баллов)**

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

Дан неориентированный граф. Над ним в заданном порядке производят операции следующих двух типов:

- **cut** — разрезать граф, то есть удалить из него ребро;
- **ask** — проверить, лежат ли две вершины графа в одной компоненте связности.

Известно, что после выполнения всех операций типа **cut** рёбер в графе не осталось. Найдите результат выполнения каждой из операций типа **ask**.

**Формат входного файла**

Первая строка входного файла содержит три целых числа, разделённые пробелами — количество вершин графа  $n$ , количество рёбер  $m$  и количество операций  $k$  ( $1 \leq n \leq 50\,000$ ,  $0 \leq m \leq 100\,000$ ,  $m \leq k \leq 150\,000$ ).

Следующие  $m$  строк задают рёбра графа;  $i$ -я из этих строк содержит два числа  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq n$ ), разделённые пробелами — номера концов  $i$ -го ребра. Вершины нумеруются с единицы; граф не содержит петель и кратных рёбер.

Далее следуют  $k$  строк, описывающих операции. Операция типа **cut** задаётся строкой «**cut**  $u$   $v$ » ( $1 \leq u, v \leq n$ ), которая означает, что из графа удаляют ребро между вершинами  $u$  и  $v$ . Операция типа **ask** задаётся строкой «**ask**  $u$   $v$ » ( $1 \leq u, v \leq n$ ), которая означает, что необходимо узнать, лежат ли в данный момент вершины  $u$  и  $v$  в одной компоненте связности. Гарантируется, что каждое ребро графа встретится в операциях типа **cut** ровно один раз.

**Формат выходного файла**

Для каждой операции **ask** во входном файле выведите на отдельной строке слово «**YES**», если две указанные вершины лежат в одной компоненте связности, и «**NO**» в противном случае. Порядок ответов должен соответствовать порядку операций **ask** во входном файле.

**Пример**

| cutting.in | cutting.out |
|------------|-------------|
| 3 3 7      | YES         |
| 1 2        | YES         |
| 2 3        | NO          |
| 3 1        | NO          |
| ask 3 3    |             |
| cut 1 2    |             |
| ask 1 2    |             |
| cut 1 3    |             |
| ask 2 1    |             |
| cut 2 3    |             |
| ask 3 1    |             |

**Задача С. Лабиринт знаний (25 баллов)**

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

В ЛКШ построили аттракцион «Лабиринт знаний». Лабиринт представляет собой  $N$  комнат, занумерованных от 1 до  $N$ , между некоторыми из которых есть двери. Когда человек проходит через дверь, показатель его знаний изменяется на определённую величину, фиксированную для данной двери. Вход в лабиринт находится в комнате 1, выход — в комнате  $N$ . Каждый ЛКШонок проходит лабиринт ровно один раз и попадает в

группу в зависимости от набранных знаний (при входе в лабиринт этот показатель равен нулю). Ваша задача показать наилучший результат.

### Формат входного файла

Первая строка входного файла содержит целые числа  $N$  ( $1 \leq N \leq 2000$ ) — количество комнат и  $M$  ( $0 \leq M \leq 10000$ ) — количество дверей. В каждой из следующих  $M$  строк содержится описание двери — номера комнат, из которой она ведёт и в которую она ведёт, а также целое число, которое прибавляется к количеству знаний при прохождении через дверь (это число по модулю не превышает 10000). Двери могут вести из комнаты в неё саму, между двумя комнатами может быть более одной двери.

### Формат выходного файла

В выходной файл выведите «:» — если можно получить неограниченно большой запас знаний, «:(» — если лабиринт пройти нельзя, и максимальное количество набранных знаний в противном случае.

### Примеры

| maze.in               | maze.out |
|-----------------------|----------|
| 2 2<br>1 2 3<br>1 2 7 | 7        |