

## Задача А. Пересечение отрезков

Имя входного файла: `segments.in`  
Имя выходного файла: `segments.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Даны два отрезка:  $AB$  и  $CD$ . Определите, какое множество точек является пересечением этих отрезков.

### Формат входного файла

Программа получает на вход восемь целых чисел, по абсолютной величине не превосходящих  $10^4$  — координаты точек  $A$ ,  $B$ ,  $C$ ,  $D$ . Точки могут совпадать (в том числе могут совпадать и концы одного отрезка).

### Формат выходного файла

Если указанные отрезки не пересекаются, то выведите строку «`Empty`». Если отрезки пересекаются в одной точке, то выведите два числа — координаты точки пересечения. Если пересечением является отрезок, то выведите четыре числа — координаты двух концов отрезка в лексикографическом порядке (то есть сначала нужно вывести ту точку, у которой меньше координата  $x$ , а если у них равны координаты  $x$ , то ту, у которой меньше координата  $y$ ). Все числа следует выводить с точностью не менее 6 знаков после запятой.

### Примеры

<code>segments.in</code>	<code>segments.out</code>
0 0 9 9 9 5 0 5	5.0000000000 5.0000000000
0 0 9 9 15 15 7 7	7.0000000000 7.0000000000 9.0000000000 9.0000000000
0 0 9 9 10 10 10 10	<code>Empty</code>

## Задача В. Поиск цикла

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

Дан ориентированный невзвешенный граф. Необходимо определить есть ли в нём циклы, и если есть, то вывести любой из них.

### Формат входного файла

В первой строке входного файла находятся два натуральных числа  $N$  и  $M$  ( $1 \leq N \leq 100\,000$ ,  $M \leq 100\,000$ ) — количество вершин и рёбер в графе соответственно. Далее в  $M$  строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

### Формат выходного файла

Если в графе нет цикла, то вывести «NO», иначе — «YES» и затем перечислить все вершины в порядке обхода цикла.

### Примеры

<code>cycle.in</code>	<code>cycle.out</code>
2 2 1 2 2 1	YES 1 2
2 2 1 2 1 2	NO

## Задача С. Подстроки

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

Дано  $K$  строк из маленьких латинских букв. Требуется найти их наибольшую общую подстроку.

### Формат входного файла

В первой строке число  $K$  ( $1 \leq K \leq 10$ ). В следующих  $K$  строках — собственно  $K$  строк (длины строк от 1 до 10 000).

### Формат выходного файла

Наибольшая общая подстрока.

### Примеры

<code>substr.in</code>	<code>substr.out</code>
3 abacaba mycabarchive acabistrue	cab

## Задача D. Островные государства — 2

Имя входного файла: island2.in  
Имя выходного файла: island2.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Суровые феодальные времена переживала некогда великая островная страна Байтландия. За главенство над всем островом борются два самых сильных барона. Таким образом, каждый город страны контролируется одним из правителей. Как водится издревле, некоторые из городов соединены двусторонними дорогами. Бароны очень не любят друг друга и стараются делать как можно больше пакостей. В частности, теперь для того, чтобы пройти по дороге, соединяющей города различных правителей, надо заплатить пошлину — один байтландский рубль. Кроме этого, за выезд из городов с чётными номерами берется удвоенная пошлина.

Программист Вася живет в городе номер 1. С наступлением лета он собирается съездить в город  $N$  на Всебайтландское собрание программистов. Разумеется, он хочет затратить при этом как можно меньше денег, и помочь ему здесь, как обычно, предлагается вам.

### Формат входного файла

В первой строке входного файла записаны два числа  $N$  и  $M$  ( $1 \leq N, M \leq 100\,000$ ) — количество городов и количество дорог соответственно.

В следующей строке содержится информация о городах —  $N$  чисел 1 или 2 — какому из баронов принадлежит соответствующий город.

В последних  $M$  строках записаны пары  $1 \leq a, b \leq N, a \neq b$ . Каждая пара означает наличие дороги из города  $a$  в город  $b$ . По дорогам Байтландии можно двигаться в любом направлении.

### Формат выходного файла

Если искомого пути не существует, выведите единственное слово «impossible». В противном случае в первой строке напишите минимальную стоимость и количество посещённых городов, а во вторую выведите эти города в порядке посещения. Если минимальных путей несколько, выведите любой.

### Примеры

island2.in	island2.out
7 8 1 1 1 1 2 2 1 1 2 2 5 2 3 5 4 4 3 4 7 1 6 6 7	0 5 1 2 3 4 7
5 5 1 2 1 1 2 1 2 2 3 3 5 1 4 4 5	2 3 1 4 5

## Задача Е. Выпуклая оболочка

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  $x$  и  $y$  ( $-10^9 \leq x, y \leq 10^9$ ) — 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

<code>hull.in</code>	<code>hull.out</code>
5	4
0 0	0 0
2 0	0 2
0 2	2 2
1 1	2 0
2 2	