

## Задача А. Планарность

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

Задан неориентированный граф и гамильтонов цикл в нём. Требуется расположить граф на плоскости без самопересечений таким образом, чтобы рёбра изображались отрезками и дугами окружностей.

### Формат входных данных

В первой строке входного файла содержится число вершин графа  $n$  ( $1 \leq n \leq 100$ ) и число рёбер  $m$ . В последующих  $m$  строках заданы рёбра, каждое номерами двух вершин. Далее дана последовательность вершин  $p_1, \dots, p_n$  в порядке их обхода по гамильтонову циклу.

### Формат выходных данных

Если расположить граф требуемым образом невозможно, выдать в выходной файл слово «NO». Иначе в первой строке выходного файла выдать слово «YES», во второй строке — координаты вершин графа, а в последующих  $m$  — координаты середины каждого из рёбер. Вершины и рёбра должны идти в той же последовательности, что и во входном файле.

### Примеры

стандартный ввод	стандартный вывод
3 3	YES
1 2	0 0 1 0 2 0
2 3	0.5 0
1 3	1.5 0
1 2 3	1 11

## Задача В. Площади

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

Даны  $n$  прямых на плоскости. Они делят плоскость на части, некоторые из которых конечны, некоторые — бесконечны. Найдите площади всех конечных частей.

### Формат входных данных

Первая строка содержит  $n$  — число прямых ( $1 \leq n \leq 80$ ). Каждая из следующих  $n$  строк содержит четыре целых числа  $x_1, y_1, x_2$  и  $y_2$  — координаты двух различных точек на очередной прямой. Координаты не превышают 100 по абсолютной величине. Прямые попарно различны.

### Формат выходных данных

В первой строке выведите  $k$  — число конечных частей. В следующих  $k$  строках выведите их площади в неубывающем порядке. Точность должна быть не хуже  $10^{-4}$ . Не рассматривайте части, имеющие площадь меньшую  $10^{-8}$ .

### Примеры

стандартный ввод	стандартный вывод
5 0 0 1 0 1 0 1 1 1 1 0 1 0 1 0 0 0 0 1 1	2 0.5 0.5

## Задача С. Чернослив

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

Сливочная состоит из  $m$  канав, расположенных на плоскости, и представляющих собой отрезки прямых, соединяющих между собой сливочные узлы. Канавы не пересекаются друг с другом, кроме как в сливочных узлах, которые являются концами канав. Для каждой канавы известна её пропускная способность — сколько по ней может течь единиц жидкости в единицу времени. В некотором сливочном узле  $t$  расположен общий сток сливочной. Жидкость по канаве может течь в любом направлении, но только в одном в любой конкретный момент времени. Ясно, что канавы делят плоскость на несколько частей — рабочих площадок (внешняя часть тоже считается рабочей площадкой). Вася сейчас находится в одной из частей, на границе которой находится общий сток, но по техническим причинам не может воспользоваться стоком. Однако ему необходимо срочно организовать слив масла таким образом, чтобы масло по канавам попало в сливочный узел  $t$ . Для этого Вася собирается сливать масло в некоторый фиксированный узел  $s$ . Считается, что масло распространяется моментально, и по каждой канаве может течь единиц масла не более, чем её пропускная способность. Требуется определить, какое максимальное количество единиц масла сможет сливать Вася в единицу времени.

### Формат входных данных

В первой строке входного файла задано число сливочных узлов  $n$ , число канав  $m$ , номер узла, которым собирается воспользоваться Вася  $s$  и номер сливочного узла с общим стоком  $t$  ( $1 \leq n \leq 50\,000$ ,  $1 \leq m \leq 3n$ ,  $1 \leq s, t \leq n$ ,  $s \neq t$ ). Далее следуют  $n$  строк — координаты сливочных узлов  $x_i y_i$  — целые числа, не превосходящие по модулю  $10^9$ . Никакие два сливочных узла не находятся в одной точке. Далее следуют  $m$  строк — описания канав, проведённых между сливочными узлами в виде  $u_j v_j w_j$  ( $1 \leq u_j, v_j \leq n$ ,  $u_j \neq v_j$ ,  $1 \leq w_j \leq 10^6$ ), описывающих канаву между узлами  $u_j$  и  $v_j$ , имеющую целую пропускную способность  $w_j$ . Гарантируется, что канавы не пересекаются во внутренних точках, и что сливочные узлы  $s$  и  $t$  доступны со сливочной площадки, на которой находится Вася, при нормальном функционировании сливочной. Между двумя узлами находится не более одной канавы.

### Формат выходных данных

Необходимо вывести максимальное количество масла, которое может сливать Вася в единицу времени.

### Примеры

стандартный ввод	стандартный вывод
4 5 1 4 0 1 1 0 1 2 2 1 1 2 1000 1 3 1000 2 3 1 3 4 1000 2 4 1000	2000

## Задача D. Crossroads

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

Ведущий байтландский производитель автомобилей — компания «<CDR Cars>» — планирует строительство нового завода. Завод будет состоять из  $n$  цехов. Некоторые цеха планируется соединить дорогами с двусторонним движением, причём любые два цеха планируется соединить не более, чем одной дорогой. Руководство «<CDR Cars>» интересуется, можно ли расположить цеха и дороги таким образом, чтобы любые две дороги пересекались только в случае, если они выходят из одного цеха (и только в точке, соответствующей этому цеху).

Заметим, что дороги не обязаны быть прямолинейными.

### Формат входных данных

В первой строке содержится целое число  $t$  ( $1 \leq t \leq 500$ ) — количество тестовых примеров. В каждой из следующей строк описано расположение цехов и дорог. Пусть у нас есть  $n$  цехов ( $1 \leq n \leq 6$ ). Тогда описание состоит из  $\frac{n(n-1)}{2}$  символов '0' или '1', обозначающих наличие (1) или отсутствие (0) дорог между цехами 1 и 2, 1 и 3, 2 и 3, 1 и 4 и так далее (если использовать термины теории графов, то задаётся нижний треугольник матрицы смежности, выписанный слитно построчно).

### Формат выходных данных

Для каждого тестового примера выведите «YES», если цеха можно расположить так, чтобы никакие две дороги не пересекались во внутренних точках, и «NO» в противном случае.

### Примеры

planaritycheck.in	planaritycheck.out
3	NO
1111111111	NO
000111111011100	YES
111111	