

## Задача А. Степени вершин (1 балл) (!)

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Неориентированный граф задан списком ребер. Найдите степени всех вершин графа.

### Формат входного файла

Входной файл содержит числа  $n$  ( $1 \leq n \leq 100$ ) — число вершин в графе и  $m$  ( $1 \leq m \leq n(n-1)/2$ ) — число ребер. Затем следует  $m$  пар чисел — ребра графа. Гарантируется, что граф не содержит петель и кратных ребер.

### Формат выходного файла

Выведите в выходной файл  $n$  чисел — степени вершин графа.

### Пример

<code>input.txt</code>	<code>output.txt</code>
4 4 1 2 1 3 2 3 3 4	2 2 3 1

## Задача В. Остовное дерево (2 балла)

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

Даны точки на плоскости, являющиеся вершинами полного графа. Вес ребра равен расстоянию между точками, соответствующими концам этого ребра. Требуется в этом графе найти остовное дерево минимального веса.

### Формат входного файла

Первая строка входного файла содержит натуральное число  $n$  — количество вершин графа ( $1 \leq n \leq 5000$ ). Каждая из следующих  $n$  строк содержит два целых числа  $x_i, y_i$  — координаты  $i$ -й вершины ( $-10\,000 \leq x_i, y_i \leq 10\,000$ ). Никакие две точки не совпадают.

### Формат выходного файла

Первая строка выходного файла должна содержать одно вещественное число — вес минимального остовного дерева.

### Примеры

<code>spantree.in</code>	<code>spantree.out</code>
3 0 0 1 0 0 1	2

## Задача С. Остовное дерево 3 (3 балла)

Имя входного файла: `spantree3.in`  
Имя выходного файла: `spantree3.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайта

Требуется найти в связном графе остовное дерево минимального веса.

### Формат входного файла

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количество вершин и ребер графа соответственно. Следующие  $m$  строк содержат описание ребер по одному на строке. Ребро номер  $i$  описывается тремя натуральными числами  $b_i$ ,  $e_i$  и  $w_i$  — номера концов ребра и его вес соответственно ( $1 \leq b_i, e_i \leq n$ ,  $0 \leq w_i \leq 100\,000$ ).  $n \leq 50\,000$ ,  $m \leq 200\,000$ .

Граф является связным.

### Формат выходного файла

Первая строка выходного файла должна содержать одно натуральное число — вес минимального остовного дерева.

### Примеры

<code>spantree3.in</code>	<code>spantree3.out</code>
<code>4 4 1 2 1 2 3 2 3 4 5 4 1 4</code>	<code>7</code>

## Задача D. Алгоритм двух китайцев (1 балл) (\*)

Имя входного файла: `chinese.in`  
Имя выходного файла: `chinese.out`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 256 мегабайт

Дан ориентированный взвешенный граф. Покрывающим деревом с корнем в вершине  $u$  назовем множество ребер, таких что из вершины  $u$  достижима любая другая вершина  $v$ , притом единственным образом. Весом дерева назовем сумму весов его ребер.

Требуется определить, существует ли в данном графе покрывающее дерево с корнем в вершине с номером 1. В случае существования требуется определить его минимальный вес.

### Формат входного файла

В первой строке входного файла два числа:  $n$  и  $m$  ( $2 \leq n \leq 1000, 1 \leq m \leq 10000$ ), где  $n$  — количество вершин графа, а  $m$  — количество ребер.

Следующие  $m$  строк содержат описание ребер. Каждое ребро задается стартовой вершиной, конечной вершиной и весом ребра. Вес каждого ребра — целое число, не превосходящее по модулю  $10^9$ .

### Формат выходного файла

Если покрывающее дерево существует, выведите в первой строке выходного файла YES, а во второй строке целое число — его минимальный вес. В противном случае в единственной строке выведите NO.

### Примеры

<code>chinese.in</code>	<code>chinese.out</code>
2 1 2 1 10	NO
4 5 1 2 2 1 3 3 1 4 3 2 3 2 2 4 2	YES 6