

## Задача А. Флойд

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

Полный ориентированный взвешенный граф задан матрицей смежности. Постройте матрицу кратчайших путей между его вершинами.

Гарантируется, что в графе нет циклов отрицательного веса.

### Формат входных данных

В первой строке вводится единственное число  $N$  ( $1 \leq N \leq 100$ ) — количество вершин графа. В следующих  $N$  строках по  $N$  чисел задается матрица смежности графа ( $j$ -ое число в  $i$ -ой строке — вес ребра из вершины  $i$  в вершину  $j$ ). Все числа по модулю не превышают 100. На главной диагонали матрицы — всегда нули.

### Формат выходных данных

Выведите  $N$  строк по  $N$  чисел — матрицу расстояний между парами вершин, где  $j$ -ое число в  $i$ -ой строке равно весу кратчайшего пути из вершины  $i$  в  $j$ .

### Примеры

floyd.in	floyd.out
4	0 5 7 13
0 5 9 100	12 0 2 8
100 0 2 8	11 16 0 7
100 100 0 7	4 9 11 0
4 100 100 0	

## Задача В. Расстояние между вершинами

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

Дан неориентированный взвешенный граф.

Найти вес минимального пути между двумя вершинами.

### Формат входных данных

Первая строка входного файла содержит натуральные числа  $N$ ,  $M$ , вторая строка числа  $S$  и  $F$  ( $N \leq 5000$ ,  $M \leq 100000$ ,  $1 \leq S, F \leq N$ ,  $S \neq F$ ) — количество вершин и ребер графа а также номера вершин, длину пути между которыми требуется найти. Следующие  $M$  строк по три натуральных числа  $b_i$ ,  $e_i$  и  $w_i$  — номера концов  $i$ -ого ребра и его вес соответственно ( $1 \leq b_i, e_i \leq n$ ,  $0 \leq w_i \leq 100000$ ).

### Формат выходных данных

Первая строка должна содержать одно натуральное число — вес минимального пути между вершинами  $S$  и  $F$ . Во второй строке через пробел выведите вершины на кратчайшем пути из  $S$  в  $F$  в порядке обхода. Если путь из  $S$  в  $F$  не существует, выведите  $-1$ .

### Примеры

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

## Задача С. Цикл отрицательного веса

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

Дан ориентированный граф. Определите, есть ли в нем цикл отрицательного веса, и если да, то выведите его.

### Формат входных данных

Во входном файле в первой строке число  $N$  ( $1 \leq N \leq 100$ ) — количество вершин графа. В следующих  $N$  строках находится по  $N$  чисел — матрица смежности графа. Все веса ребер не превышают по модулю 10 000. Если ребра нет, то соответствующее число равно 100 000.

### Формат выходных данных

В первой строке выходного файла выведите «YES», если цикл существует или «NO» в противном случае. При его наличии выведите во второй строке количество вершин в искомом цикле и в третьей строке — вершины входящие в этот цикл в порядке обхода.

### Примеры

<code>negcycle.in</code>	<code>negcycle.out</code>
2	YES
0 -1	2
-1 0	2 1

## Задача D. Защищенное соединение

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

В свете недавних новостей о прослушке каналов связи, два непримиримых интернет-гиганта Урагании «Laim.UR» и «Xenda» решили подписать соглашение об установлении защищенного канала связи между дата-центрами друг друга. В Урагании  $n$  городов, но, к сожалению, ни в одном городе нет дата-центров обоих гигантов. Поэтому для формирования защищенного канала придется прокладывать междугородные линии связи.

Специалисты компаний определили  $m$  пар городов, которые можно соединить, проложив сегмент канала связи, и оценили стоимость создания такого сегмента для каждой из этих пар.

Результирующий канал может состоять из нескольких сегментов. Он должен начинаться в одном из городов, где находится дата-центр первой компании, может проходить через промежуточные города и должен заканчиваться в городе, где находится дата-центр второй компании.

Теперь необходимо определить минимальную стоимость защищенного канала, соединяющего два дата-центра компаний.

### Формат входных данных

В первой строке находятся целые числа  $n$  и  $m$  ( $2 \leq n \leq 5000$ ,  $1 \leq m \leq 10^5$ ) — количество городов и количество пар городов, которые можно соединить сегментом канала связи.

Во второй строке находятся  $n$  целых чисел  $a_i$  ( $0 \leq a_i \leq 2$ ). Если  $a_i = 0$ , то в  $i$ -м городе нет дата-центра ни одного из гигантов. Если  $a_i = 1$ , то в  $i$ -м городе есть дата-центр «Laim.UR», а если  $a_i = 2$ , то в  $i$ -м городе находится дата-центр «Xenda». Гарантируется, что среди этих чисел есть как минимум одна единица и одна двойка.

В каждой из следующих  $m$  строк находится по три целых числа —  $s_i$ ,  $t_i$  и  $c_i$ , которые означают, что города  $s_i$  и  $t_i$  ( $1 \leq s_i, t_i \leq n$ ,  $s_i \neq t_i$ ) можно соединить сегментом канала связи стоимостью  $c_i$  ( $1 \leq c_i \leq 10^5$ ). Каждую пару городов можно соединить не более чем одним сегментом канала.

### Формат выходных данных

Если соединить защищенным каналом связи два дата-центра разных интернет-гигантов возможно, то выведите в выходной файл три числа:  $x$ ,  $y$  и  $d$ , означающие, что между городами  $x$  и  $y$  возможно провести канал связи суммарной стоимостью  $d$ . В городе  $x$  должен находиться дата-центр «Laim.UR», в городе  $y$  — дата-центр «Xenda». Если существует несколько оптимальных ответов, выведите любой. Если провести искомый канал невозможно, выведите  $-1$ .

### Примеры

<code>secure.in</code>	<code>secure.out</code>
6 7	3 4 5
1 0 1 2 2 0	
1 3 3	
1 2 4	
2 3 3	
2 4 2	
1 6 5	
3 5 6	
5 6 1	