

Задача А. Флойд

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
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 -е число в i -й 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 -е число в i -й строке равно весу кратчайшего пути из вершины i в j .

Пример

стандартный ввод	стандартный вывод
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 С. Цикл

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

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

Формат входных данных

В первой строке содержится число n ($1 \leq n \leq 100$) – количество вершин графа. В следующих n строках находится по n чисел – матрица смежности графа. Веса ребер по модулю меньше 10^5 . Если ребра нет, соответствующее значение равно 10^5 .

Формат выходных данных

В первой строке выведите «YES», если цикл существует, или «NO» в противном случае. При наличии цикла выведите во второй строке количество вершин в нем (считая одинаковые – первую и последнюю), а в третьей строке – вершины, входящие в этот цикл, в порядке обхода. Если циклов несколько, то выведите любой из них.

Пример

стандартный ввод	стандартный вывод
3 100000 100000 -51 100 100000 100000 100000 -50 100000	YES 4 2 1 3 2

Задача D. Егор и граф

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

У Егора есть взвешенный ориентированный граф, состоящий из n вершин. В этом графе между любой парой различных вершин есть ребро в обоих направлениях. Егор любит играть с графом, и сейчас он придумал новую игру:

- Игра состоит из n шагов.
- На i -том шаге Егор удаляет из графа вершину номер x_i . Удаляя вершину, Егор удаляет все ребра, которые входили в данную вершину и которые выходили из нее.
- Перед выполнением каждого шага, Егор хочет знать сумму длин кратчайших путей между всеми парами оставшихся вершин. Кратчайший путь может проходить через любую оставшуюся вершину. Другими словами, если обозначить как $d(i, v, u)$ кратчайший путь между вершинами v и u в графе, который получился до удаления вершины x_i , то Егор хочет знать значение следующей суммы: $\sum_{v, u, v \neq u} d(i, v, u)$.

Помогите Егору, выведите значение искомой суммы перед каждым шагом.

Формат входных данных

В первой строке содержится целое число n ($1 \leq n \leq 500$) — количество вершин в графе.

В следующих n строках содержится по n целых чисел — матрица смежности графа: j -тое число в i -той строке a_{ij} ($1 \leq a_{ij} \leq 10^5$, $a_{ii} = 0$) обозначает вес ребра, ведущего из вершины i в вершину j .

В следующей строке содержится n различных целых чисел: $x_1, x_2, \dots, x_n$ ($1 \leq x_i \leq n$) — вершины, которые удаляет Егор.

Формат выходных данных

Выведите n целых чисел — i -тое число равно искомой сумме перед i -тым шагом.

Пожалуйста, не используйте спецификатор `%lld` для чтения или записи 64-х битовых чисел на `C++`. Рекомендуется использовать потоки `cin`, `cout` или спецификатор `%I64d`.

Примеры

stdin	stdout
1 0 1	0
2 0 5 4 0 1 2	9 0
4 0 3 1 1 6 0 400 1 2 4 0 1 1 1 1 0 4 1 2 3	17 23 404 0

Задача Е. Флойд

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дан ориентированный граф, в котором могут быть кратные ребра и петли. Каждое ребро имеет вес, выражающийся целым числом (возможно, отрицательным). Гарантируется, что циклы отрицательного веса отсутствуют.

Требуется посчитать матрицу длин кратчайших путей от каждой вершины до всех остальных вершин.

Формат входных данных

Программа получает сначала число N ($1 \leq N \leq 100$) — количество вершин графа и число M ($0 \leq M \leq 10^4$) — количество ребер.

В следующих строках идет M троек чисел, описывающих ребра: начало ребра, конец ребра и вес (вес — целое число от -100 до 100).

Формат выходных данных

Требуется вывести матрицу N на N . В j -м столбце i -й строки следует вывести длину кратчайшего пути от вершины i до вершины j , либо число 30 000, если не существует пути между данными вершинами.

Пример

стандартный ввод	стандартный вывод
6 4	0 10 20 30000 30000 30000
1 2 10	30000 0 10 30000 30000 30000
2 3 10	30000 30000 0 30000 30000 30000
1 3 100	30000 30000 30000 0 -10 30000
4 5 -10	30000 30000 30000 30000 0 30000
	30000 30000 30000 30000 30000 0

Задача F. Pink Floyd

Имя входного файла: floyd.in
Имя выходного файла: floyd.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

256 мегабайт

Группа Pink Floyd собирается отправиться в новый концертный тур по всему миру. По предыдущему опыту группа знает, что солист Роджер Уотерс постоянно нервничает при перелетах. На некоторых маршрутах он теряет вес от волнения, а на других — много ест и набирает вес.

Известно, что чем больше весит Роджер, тем лучше выступает группа, поэтому требуется спланировать перелеты так, чтобы вес Роджера на каждом концерте был максимально возможным.

Группа должна посещать города в том же порядке, в котором она дает концерты. При этом между концертами группа может посещать промежуточные города.

Формат входных данных

Первая строка входного файла содержит три натуральных числа n , m и k — количество городов в мире, количество рейсов и количество концертов, которые должна дать группа соответственно ($n \leq 100$, $m \leq 10\,000$, $2 \leq k \leq 10\,000$). Города пронумерованы числами от 1 до n .

Следующие m строк содержат описание рейсов, по одному на строке. Рейс номер i описывается тремя числами b_i , e_i и w_i — номер начального и конечного города рейса и предполагаемое изменение веса Роджера в миллиграммах ($1 \leq b_i, e_i \leq n$, $-100\,000 \leq w_i \leq 100\,000$).

Последняя строка содержит числа $a_1, a_2, \dots, a_k$ — номера городов, в которых проводятся концерты ($a_i \neq a_{i+1}$). В начале концертного тура группа находится в городе a_1 .

Гарантируется, что группа может дать все концерты.

Формат выходных данных

Первая строка выходного файла должна содержать число l — количество рейсов, которые должна сделать группа. Вторая строка должна содержать l чисел — номера используемых рейсов.

Если существует такая последовательность маршрутов между концертами, что Роджер будет набирать вес неограниченно, то первая строка выходного файла должна содержать строку “infinitely kind”.

Пример

floyd.in	floyd.out
4 8 5	6
1 2 -2	5 6 5 7 2 3
2 3 3	
3 4 -5	
4 1 3	
1 3 2	
3 1 -2	
3 2 -3	
2 4 -10	
1 3 1 2 4	