Задача A. Максимальный поток минимальной стоимости

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

Задан ориентированный граф, каждое ребро которого обладает пропускной способностью и стоимостью. Найдите максимальный поток минимальной стоимости из вершины с номером 1 в вершину с номером n.

Формат входных данных

Первая строка входного файла содержит n и m — количество вершин и количество ребер графа ($2 \leqslant n \leqslant 100$, $0 \leqslant m \leqslant 1000$). Следующие m строк содержат по четыре целых числа числа: номера вершин, которые соединяет соответствующее ребро графа, его пропускную способность и его стоимость. Пропускные способности и стоимости не превосходят 10^5.

Формат выходных данных

В выходной файл выведите одно число — цену максимального потока минимальной стоимости из вершины с номером 1 в вершину с номером n. Ответ не превышает $2^{63} - 1$. Гарантируется, что в графе нет циклов отрицательной стоимости.

Примеры

стандартный ввод	стандартный вывод
4 5 1 2 1 2 1 3 2 2 3 2 1 1 2 4 2 1 3 4 2 3	12

Замечание

В этой задаче достаточно несколько раз пустить Форд-Беллмана...

Задача B. Задача о назначениях

Имя входного файла:	`assignment.in`
Имя выходного файла:	`assignment.out`
Ограничение по времени:	0.4 секунда
Ограничение по памяти:	256 мегабайт

Дана целочисленная матрица C размера $n \times n$. Требуется выбрать n ячеек так, чтобы в каждой строке и каждом столбце была выбрана ровно одна ячейка, а сумма значений в выбранных ячейках была минимальна.

Формат входных данных

Первая строка входного файла содержит n ($2 \leqslant n \leqslant 300$). Каждая из последующих n строк содержит по n чисел: C_{ij} Все значения во входном файле неотрицательны и не превосходят 10^6.

Формат выходных данных

В первую строку выходного файла выведите одно число — искомая минимизуруемая величина. Далее выведите n строк по два числа в каждой — номер строки и столбца клетки, участвующей в оптимальном назначении.

Пары чисел можно выводить в произвольном порядке.

Примеры

assignment.in	assignment.out
3 3 2 1 1 3 2 2 1 3	3 2 1 3 2 1 3

Задача C. Автоматное программирование

Имя входного файла:	`schedule.in`
Имя выходного файла:	`schedule.out`
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

В один замечательный день в компанию «X» завезли k автоматов. И не простых автоматов, а автоматов-программистов! Это был последний неудачный шаг перед переходом на андроидов-программистов, но это уже совсем другая история.

В компании сейчас n задач, для каждой из которых известно время начала ее выполнения s_i, длительность ее выполнения t_i и прибыль компании от ее завершения c_i. Любой автомат может выполнять любую задачу, ровно одну в один момент времени. Если автомат начал выполнять задачу, то он занят все моменты времени с s_i по $s_i + t_i - 1$ включительно и не может переключиться на другую задачу.

Вам требуется выбрать набор задач, которые можно выполнить с помощью этих k автоматов и который принесет максимальную суммарную прибыль.

Формат входных данных

В первой строке записаны два целых числа n и k ($1 \leqslant n \leqslant 1000$, $1 \leqslant k \leqslant 50$) — количество задач и количество автоматов, соответственно.

В следующих n строках через пробелы записаны тройки целых чисел s_i, t_i, c_i ($1 \leqslant s_i, t_i \leqslant 10^9$, $1 \leqslant c_i \leqslant 10^6$), s_i — время начала выполнения i-го задания, t_i — длительность i-го задания, а c_i — прибыль от его выполнения.

Формат выходных данных

Выведите n целых чисел $x_1, x_2, \ldots, x_n$. Число x_i должно быть равно 1, если задачу i следует выполнить, и 0 в противном случае.

Если оптимальных решений несколько, то выведите любое из них.

Примеры

schedule.in	schedule.out
3 1 2 7 5 1 3 3 4 1 3	0 1 1
5 2 1 5 4 1 4 5 1 3 2 4 1 2 5 6 1	1 1 0 0 1

Задача D. Банах

Имя входного файла:	`banach.in`
Имя выходного файла:	`banach.out`
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Стефан, Дэвид и Феликс готовят соревнование по правилам ICPC. Стефан предлагают следующую задачу:

> Дано N точек на плоскости, а также N векторов. Найдите однозначное соответствие между векторами и точками так, что если передвинуть каждую точку на соответствующий вектор, то расстояние между **любой** парой точек не уменьшится.

Дэвид справился с решением этой задачи довольно быстро и утверждает, что эта задача слишком проста для этого соревнования. Для того, чтобы убедить его, Феликс предложил следующее усложнение: среди всех возможных решений, найдите то, которое максимизирует сумму квадратов всех результирующих попарных расстояний.

Дэвиду все еще кажется, что задача слишком простая, а вам?

Формат входных данных

Первая строка входного файла содержит число N, количество точек и вектором ($1 \leqslant N \leqslant 500$).

Следующие N строк описывают точки. Каждая из них содержит два числа px_i и py_i ($0 \leqslant |px_i|, |py_i| \leqslant 10\,000$).

Затем следуют N строк с описанием векторов. Каждый вектор задан двумя числами vx_i и vy_i ($0 \leqslant |vx_i|, |vy_i| \leqslant 10\,000$).

Формат выходных данных

Если способ сопоставить точки и вектора таким образом, чтобы все попарные расстояния не уменьшились существует, выведите “`Yes`” в первой строке выходного файла. На следующей строке, выведите N различных чисел от 1 до N, i-е из которых равно номеру вектора, сопоставленного i-й точке.

Не забудьте выбрать ответ с максимально возможной суммой квадратов всех результирующих попарных расстояний. Если существует несколько возможных решений, выведите любое из них.

Если требование выполнить невозможно, выведите “`No`” в единственной строке выходного файла.

Примеры

banach.in	banach.out
2 0 0 1 0 2 0 -2 0	Yes 2 1
2 2 2 2 2 -1 -1 -1 -1	Yes 1 2

Замечание

В первом примере, существует только два возможных способа сопоставить векторы точкам. В обоих сопоставлениях, “`1 2`” и “`2 1`”, расстояние между каждой парой точек не уменьшается. В первом случае сумма квадратов расстояний между всеми парами точек равна 9, но во втором случае она равна 25. Поэтому единственным правильным решением является “`2 1`”.

Во втором примере, опять же, только два случая установить соответствие. В обоих случаях, расстояние между всеми парами точек не уменьшается, и сумма квадратов попарных расстояний равна 0. Поэтому, оба ответа корректны.