Задача A. Максимальный поток минимальной стоимости

Имя входного файла:	`mincost.in`
Имя выходного файла:	`mincost.out`
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

mincost.in	mincost.out
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:	2 секунды
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 минимизируемая 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. Аарельские горы

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

Вы — могучий волшебник, путешествующий по Аарельскому королевству. Ваш путь лежит через крутые холмы и горы. Горная цепочка может быть представлена, как последовательность n регионов; каждый регион имеет свою *высоту* h_i.

К сожалению, у вас нет никакого скалолазного оборудования (и, честно говоря, никакого умения его использовать), и ваши заклинания полета требуют перезарядки. Однако, вы сможете пересечь эту область, если высоты регионов будут образовывать неубывающую последовательность, то есть $h_i \leqslant h_{i+1}$ для всех i от 1 до $(n-1)$ (вы путешествуете справа налево, и падения с высокой высоты для вас опасности не представляют).

Единственное заклинание, которое может вам помочь, это заклинание Землетрясение. Это заклинание влияет на несколько соседних регионов и позволяет вам изменить их высоты. Вы знаете m видов Землетрясения; каждое заклинание имеет свою длину l_i (количество соседних регионов, затронутых этих заклинанием), свою энергетическую стоимость c_i и либо увеличивает, либо уменьшает высоты всех затронутых регионов (одно заклинание может либо повышать, либо понижать регионы, но не и то, и то). Каждое заклинание может быть применено к любому отрезку длины l_i, полностью лежащему внутри области из n регионов, и может быть применено сколько угодно раз (но его энергетическую стоимость придется платить при каждом применении). Разрешено, чтобы какие-то регионы имеют отрицательную высоту после применения заклинания.

Определите, возможно ли изменить ландшафт так, что вы сможете пересечь заданную область, и, если это возможно, найдите минимальную суммарную энергию, которую придется потратить на применение заклинаний.

Формат входных данных

Первая строка содержит два числа n и m, разделенных пробелами ($1 \leqslant n, m \leqslant 200$).

Вторая строка содержит n чисел h_i — изначальную высоту регионов ($0 \leqslant h_i \leqslant 10^6$).

Следующих m строчек описывают виды заклинаний Землетрясение, которые вы знаете, i-я из этих строк содержит описание заклинания в формате "t_i l_i c_i"; где t_i это символ "+" если заклинание позволяет увеличить высоту на 1, или "-" если оно позволяет уменьшить высоту на 1; l_i и c_i — длина и энергетическая стоимость заклинания, соответственно ($1 \leqslant l_i \leqslant n$, $1 \leqslant c_i \leqslant 10^6$).

Формат выходных данных

Если вы можете пройти через эту область, выведите минимальную возможную суммарную энергию, потраченную на заклинания, иначе, выведите -1.

Примеры

aarelia.in	aarelia.out
3 2 3 2 1 + 1 1 - 1 1	2
3 1 3 2 1 + 2 1	-1

Задача E. Посвящение

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

В этот раз, чтобы стать ЛКШонком, нужно пройти страшный-страшный лабиринт. Лабиринт настолько запутанный и опасный, что школьников в него надо пускать парами. Конечно же, пара должна состоять из мальчика и девочки. Поскольку в ЛКШ разное количество мальчиков и девочек, кому-то придётся проходить лабиринт несколько раз (главное, чтобы школьник прошёл его хотя бы раз).

Для каждой пары мальчик-девочка, которые дружат между собой, культорги знают время, за которое эта парочка найдёт выход из лабиринта. Помогите им провести всех детей через лабиринт за минимально возможное время.

Формат входных данных

Первая строка входного файла содержит два целых числа n и m — количество мальчиков и девочек в ЛКШ соответственно ($1 \leqslant n, m \leqslant 100$). вторая строка содержит число r — количество пар, которых можно пускать вместе ($1 \leqslant r \leqslant 1000$). Следующие r строк содержат по три числа каждая: a_i, b_i и c_i. Эти числа означают, что мальчик с номером a_i может пойти в лабиринт с девочкой с номером b_i, и они пробудет там вместе c_i секунд ($1 \leqslant c_i \leqslant 1000$). Гарантируется, что у кажого школьника есть друг/подруга, с которым/ой она/он может пойти в лабиринт.

Формат выходных данных

На первой строке выходного файла выведите минимальное время, за которое можно провести посвящение. На второй строке выведите k — количество пар, которые следует пустить в лабиринт. Третья строка должна содержать k целых чисел — номера этих пар, как они даны во входном файла.

Примеры

initiation.in	initiation.out
3 3 7 1 1 3 1 2 2 1 3 4 2 1 3 2 2 9 3 1 2 3 3 11	11 4 2 3 4 6