

4B. Задача о назначениях

ограничение по времени на тест: 1.5 секунд

ограничение по памяти на тест: 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.

Входные данные

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

Выходные данные

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.

Пример

входные данные	Скопировать
<pre>3 3 2 1 1 3 2 2 1 3</pre>	
выходные данные	Скопировать
<pre>3 2 1 3 2 1 3</pre>	

Примечание

Обратите внимание, что эту задачу можно сдать с помощью алгоритма за $O(n^3)$, **без использования** Венгерского алгоритма, а используя только максимальный поток минимальной стоимости.

4Е. Автоматное программирование

ограничение по времени на тест: 5 секунд

ограничение по памяти на тест: 256 мегабайт

В один замечательный день в компанию «Х» 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.

Входные данные

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

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

Выходные данные

Выведите n целых чисел x_1, x_2, \dots, 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

входные данные	Скопировать
<pre>3 1 2 7 5 1 3 3 4 1 3</pre>	
выходные данные	Скопировать
<pre>0 1 1</pre>	

входные данные	Скопировать
<pre>5 2 1 5 4 1 4 5 1 3 2 4 1 2 5 6 1</pre>	
выходные данные	Скопировать
<pre>1 1 0 0 1</pre>	

7A. Диофантово уравнение

ограничение по времени на тест: 0.25 секунд

ограничение по памяти на тест: 256 мегабайт

Даны натуральные числа a , b и c . Решите в целых числах уравнение $ax + by = c$. Среди множества решений следует выбрать такое, где x имеет наименьшее неотрицательное значение.

Входные данные

Первая строка содержит три целых числа a и b и c ($1 \leq a, b, c \leq 10^9$).

Выходные данные

Выведите искомые x и y через пробел. Если решения не существует, выведите одну строку «Impossible».

Примеры

входные данные	Скопировать
1 2 3	
выходные данные	Скопировать
1 1	

входные данные	Скопировать
10 6 8	
выходные данные	Скопировать
2 -2	

7B. Система линейных уравнений

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Дана система из двух линейных сравнений: $x \equiv a \pmod{n}$ и $x \equiv b \pmod{m}$, где числа n и m не обязательно взаимно простые. Решите эту систему или определите, что она не имеет решений.

Входные данные

В первой строке входного файла записано единственное число $1 \leq t \leq 100\,000$. В следующих t строках содержатся по четыре целых числа a, b, n, m , задающих одну систему сравнений. Все числа не превосходят по модулю 10^4 , $n > 1, m > 1$.

Выходные данные

Программа должна вывести t строк, по одной на каждую систему.

В случае, если система не имеет решений, выведите строку «NO» (без кавычек).

В случае, если решение есть, то необходимо вывести слово «YES» (без кавычек) и два таких целых числа x_0 и p , $0 \leq x_0 < p$, таких, что множество чисел $x = x_0 + kp$ (где k — произвольное целое число) является решением данной системы, и все решения системы описаны этой парой чисел.

Пример

входные данные	Скопировать
<pre>3 3 2 5 9 1 1 5 9 7 13 20 24</pre>	
выходные данные	Скопировать
<pre>YES 38 45 YES 1 45 NO</pre>	

7G. Проверка на простоту

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Дано n натуральных чисел a_i . Определите для каждого числа, является ли оно простым.

Входные данные

Программа получает на вход число n , $1 \leq n \leq 1000$ и далее n чисел a_i , $1 \leq a_i \leq 10^{18}$.

Выходные данные

Если число a_i простое, программа должна вывести YES, для составного числа программа должна вывести NO.

Пример

входные данные	Скопировать
4 1 5 10 239	
выходные данные	Скопировать
NO YES NO YES	

8А. Задача для второклассника

ограничение по времени на тест: 2.5 секунд

ограничение по памяти на тест: 256 мегабайт

Вам даны два числа. Необходимо найти их произведение.

Входные данные

Входные данные состоят из двух строк, на каждой из которых находится целое одно **целое** число, длина которого не превосходит двухсот пятидесяти тысяч символов.

Выходные данные

Выведите произведение данных чисел.

Примеры

входные данные	Скопировать
2 2	
выходные данные	Скопировать
4	

входные данные	Скопировать
1 -1	
выходные данные	Скопировать
-1	

входные данные	Скопировать
-1 -239	
выходные данные	Скопировать
239	

входные данные	Скопировать
0 -1	
выходные данные	Скопировать
0	

1В. Покрытие путями

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 64 мегабайта

{64 мегабайта}

Задан ориентированный ациклический граф. Требуется определить минимальное количество не пересекающихся по вершинам путей, покрывающих все вершины.

Входные данные

Первая строка входного файла содержит целые числа n и m — количества вершин и рёбер графа соответственно ($2 \leq n \leq 1\,000$, $0 \leq m \leq 10^5$). В следующих m строках содержатся по два натуральных числа — номера вершин u и v , которые соединяет ребро (u, v) .

Выходные данные

В первой строке выходного файла выведите натуральное число k — минимальное количество путей, необходимых для покрытия всех вершин.

Пример

входные данные	Скопировать
<pre>3 3 1 3 3 2 1 2</pre>	
выходные данные	Скопировать
<pre>1</pre>	

1С. Замощение доминошками

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 64 мегабайта

Дано игровое поле размера $n \times m$, некоторые клетки которого уже замощены. Замостить свободные соседние клетки поля доминошкой размера 1×2 стоит a условных единиц, а замостить свободную клетку поля квадратиком размера 1×1 — b условных единиц.

Определите, какая минимальная сумма денег нужна, чтобы замостить всё поле.

Входные данные

Первая строка входного файла содержит 4 целых числа n, m, a, b ($1 \leq n, m \leq 100$, $|a| \leq 1\,000$, $|b| \leq 1\,000$). Каждая из последующих n строк содержит по m символов: символ "." (точка) обозначает занятую клетку поля, а символ "*" (звёздочка) — свободную.

Выходные данные

В выходной файл выведите одно число — минимальную сумму денег, имея которую можно замостить свободные клетки поля (и только их).

Пример

входные данные	Скопировать
<pre>2 3 3 2 . ** .*.</pre>	
выходные данные	Скопировать
<pre>5</pre>	

4A. Максимальный поток минимальной стоимости

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 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 .

Входные данные

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

Выходные данные

В выходной файл выведите одно число — цену максимального потока минимальной стоимости из вершины с номером **1** в вершину с номером n . Ответ не превышает $2^{63} - 1$.

Пример

входные данные

Скопировать

```
4 5
1 2 1 2
1 3 2 2
3 2 1 1
2 4 2 1
3 4 2 3
```

выходные данные

Скопировать

```
12
```

3А. Максимальный поток на минималках

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Костя поспорил с Ваней, что сможет написать самый примитивный поиск максимального потока за 15 минут. К сожалению, Костя прослушал лекцию о поиске максимального потока, поэтому теперь ему нужна ваша помощь.

Входные данные

В первой строке входного файла содержится два числа: n и m ($2 \leq n \leq 10, 1 \leq m \leq n \cdot (n - 1)$). Это количество вершин и рёбер в графе, в котором вам требуется найти поток. Далее следуют описания рёбер графа, по одному в каждой строке входного файла. Описание ребра состоит из трёх чисел: a, b, c ($1 \leq a, b \leq n, a \neq b, 1 \leq c \leq 100$). Эти числа означают, что из вершины a в вершину b идёт ребро пропускной способности c . Гарантируется, что в графе нет кратных рёбер.

Выходные данные

В единственную строку выходного файла выведите одно число — размер максимального потока из вершины 1 в вершину n .

Пример

входные данные	Скопировать
<pre>4 5 1 2 2 1 3 3 3 2 1 2 4 3 3 4 2</pre>	
выходные данные	Скопировать
<pre>5</pre>	

3В. Максимальный поток

ограничение по времени на тест: 0.25 секунд

ограничение по памяти на тест: 256 мегабайт

Задан ориентированный граф, каждое ребро которого обладает целочисленной пропускной способностью. Найдите максимальный поток из вершины с номером **1** в вершину с номером **n** .

Входные данные

Первая строка входного файла содержит **n** и **m** — число вершин и ребер в графе ($2 \leq n \leq 500$, $1 \leq m \leq 10\,000$). Последующие строки описывают ребра. Каждое ребро задается тремя числами: начальная вершина ребра, конечная вершина ребра и пропускная способность ребра. Пропускные способности не превосходят 10^9 .

Выходные данные

Выведите величину максимального потока между вершинами **1** и **n** .

Пример

входные данные	Скопировать
<pre>4 5 1 2 1 1 3 2 3 2 1 2 4 2 3 4 1</pre>	
выходные данные	Скопировать
<pre>3</pre>	

3C. Разрез

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 64 мегабайта

Дан неориентированный граф. Найдите минимальный разрез между вершинами **1** и **n** .

Входные данные

На первой строке входного файла содержится **n** ($1 \leq n \leq 100$) — число вершин в графе и **m** ($0 \leq m \leq 400$) — количество ребер. На следующих **m** строках входного файла содержится описание ребер. Ребро описывается номерами вершин, которые оно соединяет, и его пропускной способностью (положительное целое число, не превосходящее **10 000 000**), при этом никакие две вершины не соединяются более чем одним ребром.

Выходные данные

На первой строке выходного файла должны содержаться количество ребер в минимальном разрезе и их суммарная пропускная способность. На следующей строке выведите возрастающую последовательность номеров ребер (ребра нумеруются в том порядке, в каком они были заданы во входном файле).

Пример

входные данные	Скопировать
<pre>6 8 1 2 3 1 3 3 2 4 2 2 5 2 3 4 2 3 5 2 5 6 3 4 6 3</pre>	
выходные данные	Скопировать
<pre>2 6 1 2</pre>	

3D. Улиточки

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Две улиточки Маша и Петя сейчас находятся на лужайке с абрикосами и хотят добраться до своего домика. Лужайки пронумерованы числами от 1 до n и соединены дорожками (может быть несколько дорожек соединяющих две лужайки, могут быть дорожки, соединяющие лужайку с собой же). Ввиду соображений гигиены, если по дорожке проползла улиточка, то вторая по той же дорожке уже ползти не может. Помогите Пете и Маше добраться до домика.

Входные данные

В первой строке файла записаны четыре целых числа — n , m , a и h (количество лужаек, количество дорог, номер лужайки с абрикосами и номер домика).

В следующих m строках записаны пары чисел. Пара чисел (x, y) означает, что есть дорожка с лужайки x до лужайки y (из-за особенностей улиток и местности дорожки односторонние).

Ограничения: $2 \leq n \leq 10^5, 0 \leq m \leq 10^5, a \neq h$

Выходные данные

Если существует решение, то выведите YES и на двух отдельных строчках сначала путь для Машеньки (т.к. дам нужно пропускать вперед), затем путь для Пети. Если решения не существует, выведите NO. Если решений несколько, выведите любое.

Пример

входные данные	Скопировать
<pre>3 3 1 3 1 2 1 3 2 3</pre>	
выходные данные	Скопировать
<pre>YES 1 3 1 2 3</pre>	

3Е. Декомпозиция потока

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Задан ориентированный граф, каждое ребро которого обладает целочисленной пропускной способностью. Найдите максимальный поток из вершины с номером **1** в вершину с номером ***n*** и постройте декомпозицию этого потока.

Входные данные

Первая строка входного файла содержит ***n*** и ***m*** — количество вершин и количество ребер графа ($2 \leq n \leq 500$, $1 \leq m \leq 10000$). Следующие ***m*** строк содержат по три числа: номера вершин, которые соединяет соответствующее ребро графа и его пропускную способность. Пропускные способности не превосходят 10^9 .

Выходные данные

В первую строку выходного файла выведите одно число — количество путей в декомпозиции максимального потока из вершины с номером **1** в вершину с номером ***n***. Следующий строки должны содержать описания элементарных потоков, на который был разбит максимальный. Описание следует выводить в следующем формате: величина потока, количество ребер в пути, вдоль которого течет данный поток и номера ребер в этом пути. Ребра нумеруются с единицы в порядке появления во входном файле.

Пример

входные данные	Скопировать
4 5 1 2 1 1 3 2 3 2 1 2 4 2 3 4 1	
выходные данные	Скопировать
3 1 2 2 5 1 3 2 3 4 1 2 1 4	

1D. Такси

ограничение по времени на тест: 0.5 секунд
ограничение по памяти на тест: 256 мегабайт

Управлять службой такси — совсем не простое дело. Помимо естественной необходимости централизованного управления машинами для того, чтобы обслуживать заказы по мере их поступления и как можно быстрее, нужно также планировать поездки для обслуживания тех клиентов, которые сделали заказы заранее.

В вашем распоряжении находится список заказов такси на следующий день. Вам необходимо минимизировать число машин такси, необходимых чтобы выполнить все заказы.

Для простоты будем считать, что план города представляет собой квадратную решетку. Адрес в городе будем обозначать парой целых чисел: x -координатой и y -координатой. Время, необходимое для того, чтобы добраться из точки с адресом (a, b) в точку (c, d) , равно $|a - c| + |b - d|$ минут. Машина такси может выполнить очередной заказ, либо если это первый ее заказ за день, либо она успевает приехать в начальную точку из предыдущей конечной хотя бы за минуту до указанного срока. Обратите внимание, что выполнение некоторых заказов может окончиться после полуночи.

Входные данные

В первой строке входного файла записано число заказов M ($0 < M < 500$). Последующие M строк описывают сами заказы, по одному в строке. Про каждый заказ указано время отправления в формате $hh:mm$ (в интервале с $00:00$ по $23:59$), координаты (a, b) точки отправления и координаты (c, d) точки назначения. Все координаты во входном файле неотрицательные и не превосходят 200. Заказы записаны упорядоченными по времени отправления.

Выходные данные

В выходной файл выведите единственное целое число — минимальное количество машин такси, необходимых для обслуживания всех заказов.

Примеры

входные данные	Скопировать
2 08:00 10 11 9 16 08:07 9 16 10 11	
выходные данные	Скопировать
1	

входные данные	Скопировать
2 08:00 10 11 9 16 08:06 9 16 10 11	
выходные данные	Скопировать
2	