# Задача A. 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<br>3 2 1<br>1 3 2<br>2 1 3 | 3<br>2 1<br>3 2<br>1 3 |

# Задача B. 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<br>1 2 1 2<br>1 3 2 2<br>3 2 1 1<br>2 4 2 1<br>3 4 2 3 | 12 |

## Замечание

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

# Задача C. 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<br>0 0<br>1 0<br>2 0<br>-2 0 | Yes<br>2 1 |
| 2<br>2 2<br>2 2<br>-1 -1<br>-1 -1 | Yes<br>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.

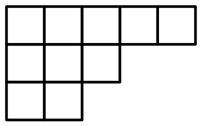
# Задача D. Увидеть Юнга и умереть

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

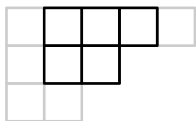
Диаграммы Юнга используются для того, чтобы изобразить разбиение числа на слагаемые. Разбиение числа $n$ на слагаемые представляет собой сумму вида $n = m_1 + m_2 + \ldots + m_k$, где $m_1 \geqslant m_2 \geqslant \ldots \geqslant m_k$.

Диаграмма состоит из $n$ квадратиков, организованных в виде $k$ рядов, где $k$ количество слагаемых в разбиении. Ряд, соответствующий числу $m_i$, содержит $m_i$ квадратиков. Все ряды выровнены по левому краю и упорядочены от более длинного к более короткому.

Например, диаграмма Юнга, приведенная на рисунке, соответствует разбиению $10 = 5 + 3 + 2$.



Иногда можно <u>вписать</u> одну диаграмму Юнга в другую. Диаграмму $X$ можно вписать в диаграмму $Y$, если можно удалить некоторые квадратики из диаграммы $Y$ так, чтобы получилась диаграмма $X$. Отметим, что разрешается только удалять некоторые квадратики, вращать или отражать диаграмму не разрешается. Например, диаграмма для разбиения $5 = 3 + 2$ может быть вписана в диаграмму для разбиения $10 = 5 + 3 + 2$, как показано на рисунке.



С другой стороны, диаграмму для разбиения $8 = 4+4$ нельзя вписать в диаграмму для разбиения $10 = 5 + 3 + 2$.

Для заданного $n$ найдите такое разбиение $n$ на слагаемые, что в соответствующую ему диаграмму Юнга можно вписать максимальное количество различных диаграмм.

Например, в диаграмму для разбиения $10 = 5 + 3 + 2$ можно вписать 36 различных диаграмм. Однако это не максимальное значение. В диаграмму для разбиения $10 = 4 + 2 + 2 + 1 + 1$ можно вписать 41 диаграмму Юнга.

## Формат входных данных

Входной файл содержит целое число $n$ ($1 \leqslant n \leqslant 100$).

## Формат выходных данных

На первой строке выходного файла выведите максимальное число диаграмм Юнга, которые можно вписать в некоторую диаграмму, соответствующую разбиению на слагаемые числа $n$.

На второй строке выведите одно или более целых чисел — количество квадратиков в каждом из рядов оптимальной диаграммы.

## Примеры

| young.in | young.out |
|---|---|
| 10 | 41<br>4 3 2 1 |