

## Задача А. Цикл отрицательного веса

Имя входного файла: `negcycle.in`  
Имя выходного файла: `negcycle.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

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

### Формат входных данных

Во входном файле в первой строке число  $N$  ( $1 \leq N \leq 100$ ) — количество вершин графа. В следующих  $N$  строках находится по  $N$  чисел — матрица смежности графа. Все веса ребер не превышают по модулю 10 000. Если ребра нет, то соответствующее число равно 100 000.

### Формат выходных данных

В первой строке выходного файла выведите «YES», если цикл существует или «NO» в противном случае. При его наличии выведите во второй строке количество вершин в искомом цикле и в третьей строке — вершины входящие в этот цикл в порядке обхода.

### Пример

<code>negcycle.in</code>	<code>negcycle.out</code>
2	YES
0 -1	2
-1 0	2 1

## Задача В. НВП

Имя входного файла: `lis.in`  
Имя выходного файла: `lis.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Числовая последовательность задана рекуррентной формулой:  $a_{i+1} = (ka_i + b) \bmod m$ . Найдите её наибольшую возрастающую подпоследовательность.

### Формат входных данных

Программа получает на вход пять целых чисел: длину последовательности  $n$  ( $1 \leq n \leq 10^5$ ), начальный элемент последовательности  $a_1$ , параметры  $k$ ,  $b$ ,  $m$  для вычисления последующих членов последовательности ( $1 \leq m \leq 10^4$ ,  $0 \leq k < m$ ,  $0 \leq b < m$ ,  $0 \leq a_1 < m$ ).

### Формат выходных данных

Требуется вывести длину НВП.

### Пример

<code>lis.in</code>	<code>lis.out</code>
5 41 2 1 100	3

### Замечание

В данном примере последовательность состоит из 5 элементов:  $a_1 = 41$ ,  $a_{i+1} = (2a_i + 1) \bmod 100$ , то есть последовательность имеет вид 41, 83, 67, 35, 71.

## Задача С. Минимальное остовное дерево

Имя входного файла: `mst.in`  
Имя выходного файла: `mst.out`  
Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

256 мегабайт

Дан связный неориентированный взвешенный граф. Необходимо выбрать в этом графе некоторое подмножество рёбер минимального суммарного веса таким образом, чтобы между любыми двумя вершинами графа существовал путь из выбранных рёбер.

Очевидно, что выбранное подмножество рёбер должно быть деревом (для минимальности суммарного веса ребер), и такое подмножество называется *минимальным остовным деревом* или *минимальным каркасом*.

### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количество вершин и ребер графа соответственно ( $1 \leq n \leq 20\,000$ ,  $0 \leq m \leq 100\,000$ ). Следующие  $m$  строк содержат описание рёбер по одному на строке. Ребро номер  $i$  описывается тремя натуральными числами  $b_i$ ,  $e_i$  и  $w_i$  — номера концов ребра и его вес соответственно ( $1 \leq b_i, e_i \leq n$ ,  $0 \leq w_i \leq 100\,000$ ).

Гарантируется, что данный граф является связным.

### Формат выходных данных

Программа должна вывести одно целое число — вес минимального остовного дерева.

### Пример

<code>mst.in</code>	<code>mst.out</code>
4 4 1 2 1 2 3 2 3 4 5 4 1 4	7

## Задача D. Следующее сочетание

Имя входного файла: `nextcomb.in`  
Имя выходного файла: `nextcomb.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Дано множество целых чисел от 1 до  $n$ . Рассмотрим подмножество этого множества, состоящее из  $k$  элементов, в возрастающем порядке.

Выведите следующее в лексикографическом порядке подмножество из  $k$  элементов.

### Формат входных данных

В первой строке входного файла содержатся целые положительные числа  $n$  и  $k$  ( $1 \leq k \leq n \leq 50$ ). Во второй строке содержится  $k$  целых чисел от 1 до  $n$  в возрастающем порядке — подмножество из  $k$  элементов.

### Формат выходных данных

Выведите следующее в лексикографическом порядке после данного подмножество из  $k$  элементов. Если следующего подмножества нет, выведите 0.

### Примеры

<code>nextcomb.in</code>	<code>nextcomb.out</code>
6 4 1 4 5 6	2 3 4 5
6 2 5 6	0

## Задача Е. Мирные ферзи

Имя входного файла: `queens.in`  
Имя выходного файла: `queens.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Известно, что на шахматной доске размером  $8 \times 8$  можно расставить 8 ферзей не бьющих друг друга, причем сделать это можно 92 способами.

Дано натуральное  $n$ . Определите сколькими способами на доске  $n \times n$  можно расставить  $n$  мирных ферзей.

### Формат входных данных

Во входном файле содержится одно число  $n$  ( $1 \leq n \leq 12$ ).

### Формат выходных данных

Выведите единственное число — ответ на задачу.

### Пример

<code>queens.in</code>	<code>queens.out</code>
8	92

## Задача F. Кони ходят по очереди

Имя входного файла: knight3.in  
Имя выходного файла: knight3.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

64 мегабайта

Переведите каждого из двух коней из одной клетки в другую за наименьшее общее число ходов. Два коня не могут одновременно находиться в одной клетке. **Ходы коней должны чередоваться.**

### Формат входных данных

Во входном файле записаны координаты первого и второго коня, затем координаты клеток, куда нужно их переместить.

### Формат выходных данных

Программа должна вывести последовательность ходов коней в виде нескольких строк. Первым символом в строке должен быть номер коня (1 или 2), затем, через пробел, координаты клетки, в которую он переставляется. Необходимо вывести любое из возможных оптимальных решений. Кони должны ходить по очереди, первым может ходить любой из коней, кони могут сделать различное число ходов.

### Пример

knight3.in	knight3.out
a1	1 b3
c2	2 a1
c2	1 d4
a1	2 b3
	1 c2
	2 a1

## Задача G. Объединение прямоугольников

Имя входного файла: `union.in`  
Имя выходного файла: `union.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

64 мегабайта

На плоскости задано  $N$  прямоугольников с вершинами в точках с целыми координатами и сторонами, параллельными осям координат. Необходимо найти площадь их объединения.

### Формат входных данных

В первой строке входного файла указано число  $N$  ( $1 \leq N \leq 3500$ ). В следующих  $N$  строках заданы по 4 целых числа  $x_1, y_1, x_2, y_2$  — сначала координаты левого нижнего угла прямоугольника, потом правого верхнего ( $0 \leq x_1 \leq x_2 \leq 10^9, 0 \leq y_1 \leq y_2 \leq 10^9$ ). Обратите внимание, что прямоугольники могут вырождаться в отрезки и даже в точки.

### Формат выходных данных

В выходной файл выведите единственное число — ответ на задачу.

### Пример

<code>union.in</code>	<code>union.out</code>
3 1 1 3 5 5 2 7 4 2 4 6 7	23

## Задача Н. Доминошки

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

Вам дано клетчатое поле размера  $n \times m$  клеток. Некоторые клетки на этом поле отмечены как занятые. На нем необходимо разместить несколько (возможно, 0) доминошек размера  $1 \times 2$ , соблюдая следующие ограничения:

- каждая клетка может быть покрыта не более чем одной доминошкой
- никакие две соседних по стороне клетки не могут быть заняты разными доминошками

Сколькими способами можно это сделать?

### Формат входных данных

В первой строке входного файла содержится два целых числа:  $n$  и  $m$  ( $1 \leq n \leq 6, 1 \leq m \leq 100$ ). В следующих  $n$  строках содержится описание поля. Символ  $X$  означает занятую клетку, символ  $.$  означает свободную клетку.

### Формат выходных данных

Выведите одно целое число — количество способов разместить доминошки, взятое по модулю  $10^9 + 7$ .

### Примеры

dominoes.in	dominoes.out
2 2 .. ..	5
3 3 ... .X. ...	21
5 5 ..... ..... ..X.. ..... .....	5182



## Задача I. Хоть где-то надо подумать

Имя входного файла: dominoes.in  
Имя выходного файла: dominoes.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

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

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

### Формат входных данных

Первая строка входного файла содержит 4 целых числа  $n, m, a, b$  ( $1 \leq n, m \leq 100, |a| \leq 1000, |b| \leq 1000$ ). Каждая из последующих  $n$  строк содержит по  $m$  символов: символ "." (точка) обозначает занятую клетку поля, а символ "\*" (звёздочка) — свободную.

### Формат выходных данных

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

### Пример

dominoes.in	dominoes.out
2 3 3 2 .** .*.	5