

Задача А. Золотой песок

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

Во время ограбления магазина вор обнаружил N ящиков с золотым песком. В ящичек под номером i песок имеет стоимость v_i и вес w_i . Чтобы унести награбленное, вор использует рюкзак. Требуется определить наибольшую суммарную стоимость песка, который может унести грабитель, если грузоподъемность рюкзака ограничена величиной W .

Из ящиков можно пересыпать любое количество песка, тогда отношение стоимости отсыпанного песка к стоимости всего ящичка будет равна отношению объема пересыпанного песка к объему всего ящичка.

Формат входного файла

В первой строке входного файла записаны два числа — N и W ($1 \leq N \leq 1000$, $1 \leq W \leq 10^6$). Далее следует N строк по два целых числа в каждой. В i -ой строке записана стоимость v_i и вес w_i песка в i -ом ящичке. Все числа неотрицательные и не превосходят 10^6 .

Формат выходного файла

Выполните искомую максимальную стоимость с точностью до 3 знаков после запятой.

Пример

dust.in	dust.out
3 50	
60 20	
100 50	
120 30	

Задача В. От списка ребер к матрице смежности

Имя входного файла: **e2m.in**
Имя выходного файла: **e2m.out**
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Простой неориентированный граф задан списком ребер, выведите его представление в виде матрицы смежности.

Формат входного файла

Входной файл содержит числа N ($1 \leq N \leq 100$) — число вершин в графе и M ($1 \leq M \leq \frac{n(n-1)}{2}$) — число ребер. Затем следуют M пар чисел — ребра графа.

Формат выходного файла

Выполните в выходной файл матрицу смежности заданного графа.

Пример

e2m.in	e2m.out
3 3	0 1 1
1 2	1 0 1
2 3	1 1 0
1 3	

Задача С. Рюкзак

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

Найдите максимальный вес золота, который можно унести в рюкзаке вместимостью S , если есть N золотых слитков с заданными весами.

Формат входного файла

В первой строке входного файла записаны два числа — S и N ($1 \leq S \leq 10\,000$, $1 \leq N \leq 300$). Далее следует N неотрицательных целых чисел, не превосходящих 100 000 — веса слитков.

Формат выходного файла

Выполните искомый максимальный вес.

Пример

knapsack.in	knapsack.out
10 3	
5 7 4	9

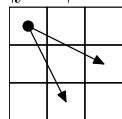
Задача D. Ход конём

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

Дана прямоугольная доска $N \times M$ (N строк и M столбцов). В левом верхнем

углу находится шахматный конь, которого необходимо переместить в правый нижний угол доски.

При этом конь может ходить следующим образом:



Необходимо определить, сколько существует различных маршрутов, ведущих из левого верхнего в правый нижний угол.

Формат входного файла

В первой строке входного файла находятся два натуральных числа N и M ($1 \leq N, M \leq 50$).

Формат выходного файла

В выходной файл выведите единственное число — количество способов добраться конём до правого нижнего угла доски.

Пример

knight.in	knight.out
3 2	1
31 34	293930

Задача Е. Максимум по минимуму

Имя входного файла: maxmin.in

Имя выходного файла: maxmin.out

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

Дан ориентированный невзвешенный граф. В нём необходимо найти вершину, кратчайшее расстояние от которой до заданной максимально.

Формат входного файла

В первой строке входного файла содержится три натуральных числа N , M и S ($1 \leq S \leq N \leq 1000$, $1 \leq M \leq 10000$) — количество вершин и рёбер в графе и номер заданной вершины соответственно. Далее в M строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

Формат выходного файла

Вывести одно целое число — искомое кратчайшее расстояние.

Пример

maxmin.in	maxmin.out
3 5 3 1 2 2 1 3 1 2 3 3 3	2

Задача F. Точки и отрезки

Имя входного файла: segments.in

Имя выходного файла: segments.out

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

Дано n отрезков на числовой прямой и m точек на этой же прямой. Для каждой из данных точек определите, скольким отрезкам она принадлежит. Точка x считается принадлежащей отрезку с концами a и b , если выполняется двойное неравенство $\min(a, b) \leq x \leq \max(a, b)$.

Формат входного файла

Первая строка содержит два целых числа n ($1 \leq n \leq 10^5$) — число отрезков и m ($1 \leq m \leq 10^5$) — число точек. В следующих n строках по два целых числа a_i и b_i — координаты концов соответствующего отрезка. В последней строке m целых чисел — координаты точек. Все числа во входном файле не превосходят по модулю 10^9 .

Формат выходного файла

В выходной файл выведите m чисел — для каждой точки количество отрезков, в которых она содержится.

Примеры

segments.in	segments.out
2 2 0 5 7 10 1 6	1 0
1 3 -10 10 -100 100 0	0 0 1

Задача G. Наибольшая последовательнократная подпоследовательность

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

Для заданной числовой последовательности a_1, a_2, \dots, a_n требуется найти длину максимальной последовательнократной подпоследовательности.

Для последовательнократной подпоследовательности $a_{k_1}, a_{k_2}, \dots, a_{k_t}$ ($k_1 < k_2 < \dots < k_t$) верно, что $a_{k_i}|a_{k_j}$ при $1 \leq i < j \leq t$ (утверждение « $a|b$ » эквивалентно « b кратно a »). Подпоследовательность из одного элемента полагается последовательнократной по определению.

Формат входного файла

В первой строке входного файла записано одно натуральное число N ($1 \leq N \leq 1000$) — количество чиел в исходной последовательности. Далее следует N натуральных чисел, не превосходящих 10^9 — сама последовательность.

Формат выходного файла

Вывести единственное число, равное искомому количеству.

Пример

sequence.in	sequence.out
4	
3 6 5 12	3

Задача H. Лесопосадки

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

Дан неориентированный невзвешенный граф. Необходимо определить, является ли он деревом.

Формат входного файла

В первой строке входного файла содержится одно натуральное число N ($N \leq 100$) — количество вершин в графе. Далее в N строках по N чисел — матрица смежности графа: в i -ой строке на j -ом месте стоит 1, если вершины i и j соединены ребром, и 0, если ребра между ними нет. На главной диагонали матрицы стоят нули. Матрица симметрична относительно главной диагонали.

Формат выходного файла

Вывести YES, если граф является деревом, NO иначе.

Пример

tree.in	tree.out
6 0 1 1 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0	NO
3 0 1 0 1 0 1 0 1 0	YES