

Задача А. Наибольший общий делитель

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

Наибольшим общим делителем двух натуральных чисел a и b называется наибольшее натуральное число d такое, что a и b делятся на d нацело без остатка.

Найдите наибольший общий делитель двух данных чисел.

Формат входного файла

В первой строке входного файла заданы через пробел два натуральных числа a и b ($1 \leq a, b \leq 100$).

Формат выходного файла

В первой строке выходного файла выведите одно число — наибольший общий делитель a и b .

Примеры

gcd.in	gcd.out
2 3	1
4 2	2

Задача В. Наименьшее общее кратное

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

Наименьшим общим кратным двух натуральных чисел a и b называется наименьшее натуральное число l такое, что оно делится на a и на b нацело без остатка.

Найдите наименьшее общее кратное двух данных чисел.

Формат входного файла

В первой строке входного файла заданы через пробел два натуральных числа a и b ($1 \leq a, b \leq 100$).

Формат выходного файла

В первой строке выходного файла выведите одно число — наименьшее общее кратное a и b .

Примеры

lcm.in	lcm.out
2 3	6
4 2	4

Задача С. Соотношение Безу

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

Наибольшим общим делителем двух натуральных чисел a и b называется наибольшее натуральное число d такое, что a и b делятся на d нацело без остатка.

Соотношение Безу — это представление наибольшего общего делителя в виде $d = xa + yb$, где x и y — некоторые целые числа.

По данным a и b найдите d и такие числа x и y , что $d = xa + yb$.

Формат входного файла

В первой строке входного файла заданы через пробел два натуральных числа a и b ($1 \leq a, b \leq 100$).

Формат выходного файла

В первой строке выходного файла выведите одно число d — наибольший общий делитель a и b . Во второй строке выведите через пробел два целых числа — x и y . Эти числа не должны превышать по модулю 100. Если правильных ответов несколько, можно вывести любой из них.

Примеры

bezout.in	bezout.out
2 3	1 -1 1
4 2	2 2 -3

Задача D. Проверка на простоту

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

Натуральное число называется *простым*, если оно делится нацело только на себя и на 1. Единица простым числом не считается.

Дано число p . Определите, простое ли оно.

Формат входного файла

В первой строке входного файла задано натуральное число p ($2 \leq p \leq 100$).

Формат выходного файла

В первой строке выходного файла выведите "YES", если число простое, и "NO" в противном случае.

Примеры

prime.in	prime.out
3	YES
6	NO

Задача E. Проверка на простоту 2

Имя входного файла: prime2.in
Имя выходного файла: prime2.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Натуральное число называется *простым*, если оно делится нацело только на себя и на 1. Единица простым числом не считается.

Дано число p . Определите, простое ли оно.

Формат входного файла

В первой строке входного файла задано натуральное число p ($1 \leq p \leq 10^9$).

Формат выходного файла

В первой строке выходного файла выведите "YES", если число простое, и "NO" в противном случае.

Примеры

prime2.in	prime2.out
3	YES
6	NO

Задача F. Проверка на простоту 3

Имя входного файла: prime3.in
Имя выходного файла: prime3.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Натуральное число называется *простым*, если оно делится нацело только на себя и на 1. Единица простым числом не считается.

Дано число p . Определите, простое ли оно.

Формат входного файла

В первой строке входного файла задано натуральное число p ($1 \leq p \leq 10^{18}$).

Формат выходного файла

В первой строке выходного файла выведите "YES", если число простое, и "NO" в противном случае.

Примеры

prime3.in	prime3.out
3	YES
6	NO

Задача G. Площадь треугольника

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

Даны три точки на плоскости. Какова площадь треугольника с вершинами в этих точках?

Формат входного файла

В первых трёх строках входного файла заданы координаты точек. Каждая из этих строк содержит два целых числа x_i и y_i , разделённых пробелом ($|x_i|, |y_i| \leq 100$). Учтите, что вершины треугольника могут лежать на одной прямой и даже совпадать.

Формат выходного файла

В первой строке выходного файла выведите одно число — площадь треугольника с вершинами в данных точках. Выводите ответ с точностью не менее шести знаков после запятой.

Примеры

tarea.in	tarea.out
0 0 0 1 1 1	0.5
-2 -2 0 0 2 3	1

Задача H. Площадь четырёхугольника

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

На плоскости задан выпуклый четырёхугольник. Найдите его площадь.

Формат входного файла

В первых четырёх строках входного файла заданы координаты вершин четырёхугольника. Каждая из этих строк содержит два целых числа x_i и y_i , разделённых пробелом ($|x_i|, |y_i| \leq 100$). Вершины находятся в попарно различных точках и перечислены в порядке обхода по часовой стрелке. Гарантируется, что четырёхугольник выпуклый, то есть все его внутренние углы строго меньше 180° .

Формат выходного файла

В первой строке выходного файла выведите одно число — площадь заданного четырёхугольника. Выводите ответ с точностью не менее шести знаков после запятой.

Примеры

qarea.in	qarea.out
0 0 0 1 1 1 1 0	1
-1 -1 0 0 2 0 4 -1	3.5

Задача I. Площадь четырёхугольника 2

Имя входного файла: qarea2.in
Имя выходного файла: qarea2.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Кривая на плоскости называется *простой*, если она не имеет самопересечений и самокасаний. Такая кривая, если она замкнута, делит плоскость ровно на две части. Фигура, ограниченная простой кривой, также называется простой.

На плоскости задан простой четырёхугольник. Найдите его площадь.

Формат входного файла

В первых четырёх строках входного файла заданы координаты вершин четырёхугольника. Каждая из этих строк содержит два целых числа x_i и y_i , разделённых пробелом ($|x_i|, |y_i| \leq 100$). Вершины находятся в попарно различных точках и перечислены в порядке обхода против часовой стрелки. Гарантируется, что четырёхугольник является простым, то есть его граница не имеет самопересечений и самокасаний.

Формат выходного файла

В первой строке выходного файла выведите одно число — площадь заданного четырёхугольника. Выводите ответ с точностью не менее шести знаков после запятой.

Примеры

qarea2.in	qarea2.out
0 0 1 0 1 1 0 1	1
-1 2 0 0 2 2 0 1	1.5

Задача J. Треугольник и точка

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

Даны треугольник и точка на плоскости. Определите, лежит ли точка внутри, на границе или вне треугольника.

Формат входного файла

В первых трёх строках входного файла заданы координаты вершин треугольника. Каждая из этих строк содержит два целых числа x_i и y_i , разделённых пробелом. В четвёртой строке заданы координаты точки в аналогичном формате. Все координаты не превосходят по модулю 100. Учтите, что вершины треугольника могут лежать на одной прямой и даже совпадать.

Формат выходного файла

Если точка лежит на стороне или в вершине треугольника, выведите "BORDER". Если точка лежит строго внутри треугольника, выведите "INSIDE". Наконец, если точка лежит вне треугольника, выведите "OUTSIDE".

Примеры

<code>inside.in</code>	<code>inside.out</code>
0 0 0 2 1 1 0 1	BORDER
0 0 0 2 2 1 1 1	INSIDE
0 0 0 1 1 1 3 2	OUTSIDE

Задача K. Пути на доске

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

Рассмотрим бесконечную клетчатую доску.

Назовём *путём* из одной клетки в другую последовательность клеток, в которой каждые две идущие подряд клетки являются соседними по стороне. Длина пути — это количество клеток в нём, не считая начальную.

Назовём путь *простым*, если в нём не встречается двух одинаковых клеток.

Зафиксируем какую-то клетку на доске. Сколько существует простых путей заданной длины, начинающихся в этой клетке?

Формат входного файла

В первой строке входного файла задано целое число n ($1 \leq n \leq 10$).

Формат выходного файла

В первой строке выходного файла выведите одно число — количество путей длины n из этой клетки.

Примеры

<code>paths.in</code>	<code>paths.out</code>
1	4
2	12

Задача L. Пути на доске 2

Имя входного файла: `paths2.in`
Имя выходного файла: `paths2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Рассмотрим бесконечную клетчатую доску.

Назовём *путём* из одной клетки в другую последовательность клеток, в которой каждые две идущие подряд клетки являются соседними по стороне. Длина пути — это количество клеток в нём, не считая начальную.

Назовём путь *простым*, если в нём не встречается двух одинаковых клеток.

Зафиксируем какую-то клетку на доске. Сколько существует простых путей заданной длины, начинающихся в этой клетке?

Формат входного файла

В первой строке входного файла задано целое число n ($0 \leq n \leq 20$).

Формат выходного файла

В первой строке выходного файла выведите одно число — количество путей длины n из этой клетки.

Примеры

<code>paths2.in</code>	<code>paths2.out</code>
1	4
2	12

Задача M. Самая большая буква

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

Дана строка. Найдите в ней букву, алфавитный номер которой — самый большой среди букв этой строки.

Формат входного файла

В первой строке входного файла задана строка длины от 1 до 100 символов, включительно. Строка состоит из строчных букв латинского алфавита.

Формат выходного файла

В первой строке выходного файла выведите одну букву — ту букву строки, алфавитный номер которой максимален.

Примеры

<code>greatest.in</code>	<code>greatest.out</code>
aab	b
dabux	y

Задача N. Различные буквы

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

Дана строка. Сколько различных букв она содержит?

Формат входного файла

В первой строке входного файла задана строка длины от 1 до 100 символов, включительно. Строка состоит из строчных букв латинского алфавита.

Формат выходного файла

В первой строке выходного файла выведите одно число — количество различных букв в этой строке.

Примеры

letters.in	letters.out
aab	2
dabux	5

Задача O. Подстрока

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

Подстрокой строки $S = s_1s_2\dots s_n$ называется непрерывная подпоследовательность символов этой строки $s_i s_{i+1} s_{i+2} \dots s_{j-1} s_j$.

Даны две строки. Сколько раз вторая из них встречается в первой как подстрока?

Формат входного файла

В первой строке входного файла задана строка длины от 1 до 100 символов, включительно. Во второй строке входного файла задана строка длины от 1 до 10 символов, включительно. Строки состоят из строчных букв латинского алфавита.

Формат выходного файла

В первой строке выходного файла выведите одно число — количество вхождений второй строки в первую как подстроки.

Примеры

substr.in	substr.out
aab ab	1
abababc aba	2

Задача P. Различные подстроки

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

Подстрокой строки $S = s_1s_2\dots s_n$ называется непрерывная подпоследовательность символов этой строки $s_i s_{i+1} s_{i+2} \dots s_{j-1} s_j$.

Дана строка. Сколько различных подстрок, не считая пустой, она содержит?

Формат входного файла

В первой строке входного файла задана строка длины от 1 до 100 символов, включительно. Строка состоит из строчных букв латинского алфавита.

Формат выходного файла

В первой строке выходного файла выведите одно число — количество различных подстрок данной строки, не считая пустой.

Примеры

unequal.in	unequal.out
aab	5
dabux	15

Задача Q. Компоненты связности

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

Назовём компонентой связности в неориентированном графе произвольное множество вершин такое, что из любой вершины этого множества существует путь в любую другую вершину этого множества, и не существует другого множества с аналогичным свойством, содержащего это множество.

Дан неориентированный граф. Найдите количество различных компонент связности в нём.

Формат входного файла

В первой строке входного файла заданы через пробел два целых числа n и m — количество вершин и рёбер в графе, соответственно ($1 \leq n \leq 10$, $0 \leq m \leq 45$). Следующие m строк описывают рёбра графа; i -ая из этих строк содержит два числа u_i и v_i через пробел — номера концов i -го ребра ($1 \leq u_i, v_i \leq n$). Гарантируется, что граф не содержит петель и кратных рёбер.

Формат выходного файла

В первой строке выходного файла выведите одно число — количество компонент связности данного графа.

Примеры

connect.in	connect.out
3 2 1 2 2 3	1
5 4 3 1 1 2 2 3 4 5	2

Задача R. Компоненты связности 2

Имя входного файла: connect2.in
Имя выходного файла: connect2.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Назовём компонентой связности в неориентированном графе произвольное множество вершин такое, что из любой вершины этого множества существует путь в любую другую вершину этого множества, и не существует другого множества с аналогичным свойством, содержащего это множество.

Дан неориентированный граф. Найдите количество различных компонент связности в нём.

Формат входного файла

В первой строке входного файла заданы через пробел два целых числа n и m — количество вершин и рёбер в графе, соответственно ($0 \leq n \leq 100$, $0 \leq m \leq 10\,000$). Следующие m строк описывают рёбра графа; i -ая из этих строк содержит два числа u_i и v_i через пробел — номера концов i -го ребра ($1 \leq u_i, v_i \leq n$). Граф может содержать петли и кратные рёбра.

Формат выходного файла

В первой строке выходного файла выведите одно число — количество компонент связности данного графа.

Примеры

connect2.in	connect2.out
3 2 1 2 2 3	1
5 4 3 1 1 2 2 3 4 5	2

Задача S. Сильная связность

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

Назовём компонентой сильной связности в ориентированном графе произвольное множество вершин такое, что из любой вершины этого множества существует путь в любую другую вершину этого множества, и не существует другого множества с аналогичным свойством, содержащего это множество.

Дан ориентированный граф. Найдите количество различных компонент сильной связности в нём.

Формат входного файла

В первой строке входного файла заданы через пробел два целых числа n и m — количество вершин и рёбер в графе, соответственно ($1 \leq n \leq 10$, $0 \leq m \leq 90$). Следующие m строк описывают рёбра графа; i -ая из этих строк содержит два числа u_i и v_i через пробел — номера начала и конца i -го ребра, соответственно ($1 \leq u_i, v_i \leq n$). Гарантируется, что граф не содержит петель и кратных рёбер.

Формат выходного файла

В первой строке выходного файла выведите одно число — количество компонент сильной связности данного графа.

Примеры

strong.in	strong.out
3 2 1 2 2 3	3
5 4 3 1 1 2 2 3 4 5	3

Задача T. Сильная связность 2

Имя входного файла: strong2.in
Имя выходного файла: strong2.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Назовём компонентой сильной связности в ориентированном графе произвольное множество вершин такое, что из любой вершины этого множества существует путь в любую другую вершину этого множества, и не существует другого множества с аналогичным свойством, содержащего это множество.

Дан ориентированный граф. Найдите количество различных компонент сильной связности в нём.

Формат входного файла

В первой строке входного файла заданы через пробел два целых числа n и m — количество вершин и рёбер в графе, соответственно ($0 \leq n \leq 100$, $0 \leq m \leq 10\,000$). Следующие m строк описывают рёбра графа; i -ая из этих строк содержит два числа u_i и v_i через пробел — номера начала и конца i -го ребра, соответственно ($1 \leq u_i, v_i \leq n$). Граф может содержать петли и кратные рёбра.

Формат выходного файла

В первой строке выходного файла выведите одно число — количество компонент сильной связности данного графа.

Примеры

strong2.in	strong2.out
3 2 1 2 2 3	3
5 4 3 1 1 2 2 3 4 5	3

Задача U. Кирпичи

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

Дано бесконечное количество кирпичей двух цветов — красного и синего. Красные кирпичи имеют длину 2 дециметра, а синие — 3 дециметра. Сколько различных способов выложить ряд из кирпичей длины n дециметров? Способы считаются различными, если на каком-то одинаковом расстоянии от начала ряда в них лежат кирпичи разного цвета.

Формат входного файла

В первой строке входного файла задано целое число n ($2 \leq n \leq 20$).

Формат выходного файла

В первой строке выходного файла выведите одно число — количество способов выложить ряд из кирпичей длины n дециметров.

Примеры

bricks.in	bricks.out
2	1
6	2

Задача V. Кирпичи 2

Имя входного файла: bricks2.in
Имя выходного файла: bricks2.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дано бесконечное количество кирпичей двух цветов — красного и синего. Красные кирпичи имеют длину 2 дециметра, а синие — 3 дециметра. Сколько различных способов выложить ряд из кирпичей длины n дециметров? Способы считаются различными, если на каком-то одинаковом расстоянии от начала ряда в них лежат кирпичи разного цвета.

Формат входного файла

В первой строке входного файла задано целое число n ($0 \leq n \leq 80$).

Формат выходного файла

В первой строке выходного файла выведите одно число — количество способов выложить ряд из кирпичей длины n дециметров.

Примеры

bricks2.in	bricks2.out
2	1
6	2