

Задача 1. Компьютерная графика

Имя входного файла:	bw.in
Имя выходного файла:	bw.out
Максимальное время работы на одном тесте:	2 секунды
Максимальный объем используемой памяти:	64 мегабайта

Одна из базовых задач компьютерной графики – обработка черно-белых изображений. Изображения можно представить в виде прямоугольников шириной w и высотой h , разбитых на $w \times h$ единичных квадратов, каждый из которых имеет либо белый, либо черный цвет. Такие единичные квадраты называются *пикселями*. В памяти компьютера сами изображения хранятся в виде прямоугольных таблиц, содержащих нули и единицы.

Во многих областях очень часто возникает задача комбинации изображений. Одним из простейших методов комбинации, который используется при работе с черно-белыми изображениями, является попиксельное применение некоторой логической операции. Это означает, что значение пиксела результата получается применением этой логической операции к соответствующим пикселям аргументов. Логическая операция от двух аргументов обычно задается таблицей истинности, которая содержит значения операции для всех возможных комбинаций аргументов. Например, для операции «исключающее ИЛИ» эта таблица выглядит так.

Первый аргумент	Второй аргумент	Результат
0	0	0
0	1	1
1	0	1
1	1	0

Требуется написать программу, которая вычислит результат попиксельного применения заданной логической операции к двум черно-белым изображениям одинакового размера.

Формат входных данных

Первая строка входного файла содержит два целых числа w и h ($1 \leq w, h \leq 100$). Последующие h строк описывают первое изображение и каждая из этих строк содержит w символов, каждый из которых равен нулю или единице. Далее следует описание второго изображения в аналогичном формате. Последняя строка входного файла содержит описание логической операции в виде четырех чисел, каждое из которых – ноль или единица. Первое из них есть результат применения логической операции в случае, если оба аргумента – нули, второе – результат в случае, если первый аргумент – ноль, второй – единица, третье – результат в случае, если первый аргумент – единица, второй – ноль, а четвертый – в случае, если оба аргумента – единицы.

Формат выходных данных

В выходной файл необходимо вывести результат применения заданной логической операции к изображениям в том же формате, в котором изображения заданы во входном файле.

Пример входных и выходных данных

bw.in	bw.out
5 3	11110
01000	11100
11110	11110
01000	
10110	
00010	
10110	
0110	

Задача 2. Уход за газоном

Имя входного файла: lawn.in
Имя выходного файла: lawn.out
Максимальное время работы на одном тесте: 2 секунды
Максимальный объем используемой памяти: 64 мегабайта

Фермер Иван с юности следит за своим газоном. Газон можно считать плоскостью, на которой в каждой точке с целыми координатами растет один пучок травы.

В одно из воскресений Иван воспользовался газонокосилкой и постриг некоторый прямоугольный участок газона. Стороны этого участка параллельны осям координат, а две противоположные вершины расположены в точках (x_1, y_1) и (x_2, y_2) . Следует отметить, что пучки травы, находящиеся на границе этого прямоугольника, также были пострижены.

Довольный результатом Иван купил и установил на газоне дождевальную установку. Она была размещена в точке с координатами (x_3, y_3) и имела радиус действия струи r . Таким образом, установка начала поливать все пучки, расстояние от которых до точки (x_3, y_3) не превышало r .

Все было хорошо, но Ивана заинтересовал следующий вопрос: сколько пучков травы оказалось и пострижено, и полито в это воскресенье?

Требуется написать программу, которая позволит дать ответ на вопрос Ивана.

Формат входных данных

В первой строке входного файла содержатся четыре целых числа x_1, y_1, x_2, y_2 ($-100\,000 \leq x_1 < x_2 \leq 100\,000$; $-100\,000 \leq y_1 < y_2 \leq 100\,000$).

Во второй строке входного файла содержатся три целых числа x_3, y_3, r ($-100\,000 \leq x_3, y_3 \leq 100\,000$; $1 \leq r \leq 100\,000$)

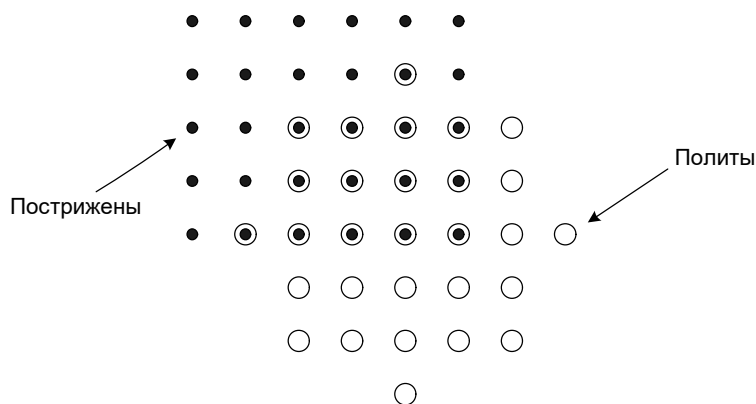
Формат выходных данных

В выходной файл необходимо вывести одно целое число — число пучков травы, которые были и пострижены, и политы.

Пример входных и выходных данных

lawn.in	lawn.out
0 0 5 4 4 0 3	14

Иллюстрация к примеру



Задача 3. Городской трамвай

Имя входного файла:	tram.in
Имя выходного файла:	tram.out
Максимальное время работы на одном тесте:	5 секунд
Максимальный объем используемой памяти:	64 мегабайта

С окраины в центр города каждое утро по одному маршруту едут в трамвае N человек. За долгое время поездок они достаточно хорошо узнали друг друга. Чтобы никому не было обидно, они захотели решить, кто из них и между какими остановками маршрута должен сидеть, а кто должен стоять. Все остановки пронумерованы от 1 до P .

Один из пассажиров оказался знатоком теории математического моделирования. Он предложил рассмотреть значение *суммарного удовлетворения* пассажиров. Для каждого i -го пассажира он оценил две величины — a_i и b_i . Если в течение одного переезда между остановками пассажир сидит, то к суммарному удовлетворению прибавляется a_i , если же он стоит, то прибавляется b_i .

Всего в трамвае M сидячих мест. Встать и садиться пассажиры могут мгновенно на любой остановке. Кроме того, некоторые пассажиры предпочитают ехать стоя, даже если в трамвае есть свободные места (для них $a_i < b_i$).

Требуется написать программу, которая вычисляет значение максимально достижимого суммарного удовлетворения, если для каждого i -го пассажира известны величины a_i и b_i , а также номера остановок, на которых он садится и выходит из трамвая.

Формат входных данных

Первая строка входного файла содержит разделенные пробелом три целых числа N , M и P — число пассажиров, число сидячих мест и число остановок на маршруте соответственно ($1 \leq N, M, P \leq 100\,000$; $2 \leq P$).

Каждая из следующих N строк содержит информацию об очередном пассажире в виде четырех целых чисел a_i, b_i, c_i, d_i , где первые два числа определяют вклад в параметр счастья, третье — номер остановки, на которой пассажир садится в трамвай, и последнее — номер остановки, на которой он выходит из трамвая ($-10^6 \leq a_i, b_i \leq 10^6$; $1 \leq c_i < d_i \leq P$).

Формат выходных данных

В выходной файл необходимо вывести одно целое число — максимальное суммарное удовлетворение, которого могут добиться пассажиры.

Пример входных и выходных данных

tram.in	tram.out
4 2 4	28
10 -10 2 3	
-1 -3 1 4	
6 -6 1 3	
7 4 2 4	

Комментарий к примеру

Максимальное суммарное удовлетворение достигается следующим образом:

- На первой остановке входят и садятся второй и третий пассажиры;
- На второй остановке входят первый и четвертый пассажиры, второй уступает место первому;
- На третьей остановке встают и выходят первый и третий пассажиры, второй и четвертый садятся на их места;
- На четвертой остановке выходят первый и третий пассажиры.

Задача 4. Треугольники и симметрия

Имя входного файла:	triangle.in
Имя выходного файла:	triangle.out
Максимальное время работы на одном тесте:	3 секунды
Максимальный объем используемой памяти:	64 мегабайта

Роман достаточно давно занимается в математическом кружке, поэтому он уже успел узнать не только правила выполнения простейших операций, но и о таком достаточно сложном понятии как симметрия. Для того чтобы лучше изучить симметрию, Роман решил начать с наиболее простых геометрических фигур – треугольников. Он скоро понял, что осевой симметрией обладают так называемые равнобедренные треугольники. Напомним, что треугольник называется *равнобедренным*, если его площадь положительна, и у него есть хотя бы две равные стороны.

Недавно Роман, зайдя в класс, увидел, что на доске нарисовано n точек. Разумеется, он сразу задумался, сколько существует троек из этих точек, которые являются вершинами равнобедренных треугольников.

Требуется написать программу, решающую указанную задачу.

Формат входных данных

Входной файл содержит в первой строке целое число n ($3 \leq n \leq 1500$). Каждая из последующих строк содержит по два разделенных пробелом целых числа x_i и y_i , определяющих координаты i -ой точки. Все координаты точек не превосходят 10^9 по абсолютной величине. Среди заданных точек нет совпадающих.

Формат выходных данных

В выходной файл необходимо вывести ответ на вышеназванную задачу.

Пример входных и выходных данных

triangle.in	triangle.out
3 0 0 2 2 -2 2	1
4 0 0 1 1 1 0 0 1	4