**Практическая работа 3**

**Питон. Двумерные списки. Кортежи.**

**Вариант 1**

**Задача 1**

Известный художник решил написать новый шедевр. После многих дней усердной работы он захотел исследовать свое творение. Художник вспомнил, что картина писалась следующим образом: сначала был взят белый холст, имеющий форму прямоугольника шириной w и высотой h. Затем художник нарисовал на этом холсте n прямоугольников со сторонами, параллельными сторонам холста и вершинами, расположенными в целочисленных координатах. Помогите художнику определить площадь незакрашенной части холста.

***Входные данные***

Первая строка входных данных содержит два натуральных числа w и h (1 ≤ w, h ≤ 100). Во второй строке записано целое число n (0 ≤ n ≤ 5000) – количество прямоугольников. Следующие n строк содержат информацию о всех прямоугольниках. Каждая строка описывает один прямоугольник в виде четырех чисел x1, y1, x2, y2 , где (x1, y1) и (x2, y2) – координаты левого верхнего и правого нижнего угла прямоугольника соответственно.

***Выходные данные***

Выведите одно целое число – площадь незакрашенной части холста.

***Примеры***

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 5 5  2  1 1 3 3  2 2 4 4 | 18 |
| 2 | 6 7  3  0 0 5 5  1 1 4 4  2 2 3 3 | 17 |

**Задача 2**

На хоккейном стадионе в одном большом городе расположено большое прямоугольное табло. Оно имеет n строк и m столбцов (то есть состоит из n x m ячеек). Во время хоккейного матча это табло служит для отображения счета и времени, прошедшего с начала тайма, а в перерывах на нем показывают различную рекламу.

В связи с этим возникла задача проверки возможности показа на этом табло определенной рекламной заставки. Заставка также, как и табло, имеет размер n строк на m столбцов. Каждая из ячеек заставки окрашена в один из четырех цветов - трех основных: красный - R, зеленый - G, синий - B и черный - .(точка).

Каждая из ячеек табло характеризуется своими цветопередаточными возможностями. Любая из ячеек табло может отображать черный цвет - это соответствует тому, что на нее вообще не подается напряжение. Также каждая из ячеек может отображать некоторое подмножество множества основных цветов. В этой задаче эти подмножества будут кодироваться следующим образом:

0 - ячейка может отображать только черный цвет;

1 - ячейка может отображать только черный и синий цвета;

2 - ячейка может отображать только черный и зеленый цвета;

3 - ячейка может отображать только черный, зеленый и синий цвета;

4 - ячейка может отображать только черный и красный цвета;

5 - ячейка может отображать только черный, красный и синий цвета;

6 - ячейка может отображать только черный, красный и зеленый цвета;

7 - ячейка может отображать только черный, красный, зеленый и синий цвета.

Напишите программу, которая по описанию табло и заставки определяет: возможно ли на табло отобразить эту заставку.

***Входные данные***

Первая строка входных данных содержит целые числа n и m (1 ≤ n, m ≤ 100). Далее идут n строк по m символов каждая - описание заставки. Каждый из символов описания заставки принадлежит множеству {R, G, B, .} . Их значения описаны выше.

После этого идет описание табло. Оно содержит n строк по m чисел, разделенных пробелами. Значения чисел описаны выше.

***Выходные данные***

Выведите YES, если на табло возможно отобразить заставку и NO - в противном случае.

***Примеры***

|  |  |  |
| --- | --- | --- |
| **№** | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 3 3 .GB R.B RG. 0 1 2 3 4 5 6 7 0 | NO |
| 2 | 2 3 RGB .G. 7 7 7 7 7 7 | YES |

**Вариант 2**

**Задача 1**

«Морской бой» - игра для двух участников, в которой игроки по очереди называют координаты на неизвестной им карте соперника. Если у соперника по этим координатам имеется корабль, то корабль или его часть «топится», а попавший получает право сделать еще один ход. Цель игрока - первым поразить все корабли противника.

«Морской бой» очень популярен среди учеников одной физико-математической школы. Ребята очень любят в него играть на переменах. Вот и сейчас ученики Иннокентий и Емельян начали новую партию.

Правила, по которым ребята расставляют корабли перед началом партии, несколько отличаются от классических. Во-первых, игра происходит на поле размером N×M, а не 10×10. Во-вторых, число кораблей, их размер и форма выбираются ребятами перед партией - так играть намного интереснее.

Емельян уже расставил все свои корабли, кроме одного однопалубного. Такой корабль занимает ровно одну клетку.

Задана расстановка кораблей Емельяна. Найдите число способов поставить оставшийся однопалубный корабль. При этом учитывайте, что по правилам его можно ставить только в ту клетку, все соседние с которой не заняты. В этой задаче соседними считаются клетки, имеющие общую сторону.

***Входные данные***

Первая строка входных данных содержит два числа: N и M (1 ≤ N, M ≤ 100). Последующие N строк описывают игровое поле - каждая из них содержит M символов. Символом «.» (точка) обозначена свободная клетка, символом «\*» (звездочка) - занятая кораблем.

***Выходные данные***

Выведите ответ на задачу.

***Примеры***

|  |  |  |
| --- | --- | --- |
| **№** | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 4 4 \*\*\*\* \*\*.. \*... \*... | 4 |
| 2 | 4 3 \*\*\* ... ... \*\*\* | 0 |

**Задача 2**

Из шахматной доски по границам клеток выпилили связную (не распадающуюся на части) фигуру без дыр. Требуется определить ее периметр.

***Входные данные***

Во входных данных сначала записано число N (1 ≤ N ≤ 64) – количество выпиленных клеток. В следующих N строках указаны координаты выпиленных клеток, разделенные пробелом (номер строки и столбца – числа от 1 до 8). Каждая выпиленная клетка указывается один раз.

***Выходные данные***

Выведите одно число – периметр выпиленной фигуры (сторона клетки равна единице).

***Примеры***

|  |  |  |
| --- | --- | --- |
| **№** | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 3 1 1 1 2 2 1 | 8 |
| 2 | 1 8 8 | 4 |

**Вариант 3**

**Задача 1**

Миша уже научился хорошо фотографировать и недавно увлекся программированием. Первая программа, которую он написал, позволяет формировать негатив бинарного черно-белого изображения.

Бинарное черно-белое изображение – это прямоугольник, состоящий из пикселей, каждый из которых может быть либо черным, либо белым. Негатив такого изображения получается путем замены каждого черного пикселя на белый, а каждого белого пикселя – на черный.

Миша, как начинающий программист, написал свою программу с ошибкой, поэтому в результате ее исполнения мог получаться некорректный негатив. Для того чтобы оценить уровень несоответствия получаемого негатива исходному изображению, Миша начал тестировать свою программу.

В качестве входных данных он использовал исходные изображения. Сформированные программой негативы он начал тщательно анализировать, каждый раз определяя число пикселей негатива, которые получены с ошибкой.

Требуется написать программу, которая в качестве входных данных использует исходное бинарное черно-белое изображение и полученный Мишиной программой негатив, и на основе этого определяет количество пикселей, в которых допущена ошибка.

***Входные данные***

Первая строка входных данных содержит целые числа n и m (1 ≤ n, m ≤ 100) – высоту и ширину исходного изображения (в пикселях). Последующие n строк содержат описание исходного изображения. Каждая строка состоит из m символов «B» и «W». Символ «B» соответствует черному пикселю, а символ «W» – белому. Далее следует пустая строка, а после нее – описание выведенного Мишиной программой изображения в том же формате, что и исходное изображение.

***Выходные данные***

Необходимо вывести число пикселей негатива, которые неправильно сформированы Мишиной программой.

***Примеры***

|  |  |  |
| --- | --- | --- |
| **№** | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 3 4 WBBW BBBB WBBW  BWWW WWWB BWWB | 2 |
| 2 | 2 2 BW BB  WW BW | 2 |

**Задача 2**

Мальчику Васе очень нравится известная игра "Сапер" ("Minesweeper").

В "Сапер" играет один человек. Игра идет на клетчатом поле (далее будем называть его картой) N×M (N строк, M столбцов). В K клетках поля стоят мины, в остальных клетках записано либо число от 1 до 8 — количество мин в соседних клетках, либо ничего не написано, если в соседних клетках мин нет. Клетки являются соседними, если они имеют хотя бы одну общую точку, в одной клетке не может стоять более одной мины. Изначально все клетки поля закрыты. Игрок за один ход может открыть какую-нибудь клетку. Если в открытой им клетке оказывается мина — он проигрывает, иначе игроку показывается число, которое стоит в этой клетке, и игра продолжается. Цель игры — открыть все клетки, в которых нет мин.

У Васи на компьютере есть эта игра, но ему кажется, что все карты, которые в ней есть, некрасивые и неинтересные. Поэтому он решил нарисовать свои. Однако фантазия у него богатая, а времени мало, и он хочет успеть нарисовать как можно больше карт. Поэтому он просто выбирает N, M и K и расставляет мины на поле, после чего все остальные клетки могут быть однозначно определены. Однако на определение остальных клеток он не хочет тратить свое драгоценное время. Помогите ему!

По заданным N, M, K и координатам мин восстановите полную карту.

***Входные данные***

В первой строке входных данных содержатся числа N, M и K (1 ≤ N ≤ 200, 1 ≤ M ≤ 200, 0 ≤ K ≤ N×M). Далее идут K строк, в каждой из которых содержится по два числа, задающих координаты мин. Первое число в каждой строке задает номер строки клетки, где находится мина, второе число — номер столбца. Левая верхняя клетка поля имеет координаты (1,1), правая нижняя — координаты (N,M).

***Выходные данные***

Вывод должен содержать N строк по M символов — соответствующие строки карты. j-й символ i-й строки должен содержать символ ‘\*‘ (звездочка) если в клетке (i,j) стоит мина, цифру от 1 до 8, если в этой клетке стоит соответствующее число, либо ‘.‘ (точка), если клетка (i,j) пустая.

***Пример***

|  |  |  |
| --- | --- | --- |
| **№** | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 10 9 23 1 8 2 3 3 2 3 3 4 3 5 7 6 7 7 1 7 2 7 3 7 4 7 5 7 6 7 7 7 8 8 1 8 3 8 5 8 7 9 3 9 5 9 6 9 7 | .111..1\*1 13\*2..111 1\*\*3..... 13\*2.111. .111.2\*2. 233335\*41 \*\*\*\*\*\*\*\*1 \*6\*7\*8\*41 13\*4\*\*\*2. .1122321. |

**Вариант 4**

**Задача 1**

Одна из базовых задач компьютерной графики – обработка черно-белых изображений. Изображения можно представить в виде прямоугольников шириной w и высотой h, разбитых на w×h единичных квадратов, каждый из которых имеет либо белый, либо черный цвет. Такие единичные квадраты называются пикселами. 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 пикселам 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.

## Входные данные

Первая строка входного файла INPUT.TXT содержит два целых числа w и h (1 ≤ w, h ≤ 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.

## Выходные данные

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

## Пример

|  |  |  |
| --- | --- | --- |
| **№** | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 5 3 01000 11110 01000 10110 00010 10110 0110 | 11110 11100 11110 |

**Задача 2**

Судоку размера n называется квадрат со стороной n2, разделенный на n2 средних квадратов со стороной n, каждый из которых разделен на n2 маленьких квадратов. В каждом маленьком квадрате записано число от 1 до n2.

Судоку называется правильным, если в каждом столбце, каждой строке и каждом среднем квадрате встречаются все числа от 1 до n2.

Недавно Вася нарисовал Судоку размера n. Ваша задача – помочь ему определить правильный ли он.

**Входные данные**

В первой строке входных данных содержится число n (1 ≤ n ≤ 10). В следующих n2 строчках содержится по n2 чисел, задающих нарисованный Васей Судоку.

Все числа во входных данных натуральные и не превосходят 100 по модулю.

**Выходные данные**

Если Судоку правильный, то выведите слово «Correct», иначе выведите «Incorrect».

**Примеры**

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 3 1 3 2 5 4 6 9 8 7  4 6 5 8 7 9 3 2 1  7 9 8 2 1 3 6 5 4  9 2 1 4 3 5 8 7 6  3 5 4 7 6 8 2 1 9  6 8 7 1 9 2 5 4 3  5 7 6 9 8 1 4 3 2  2 4 3 6 5 7 1 9 8  8 1 9 3 2 4 7 6 5 | Correct |
| 2 | 1 10 | Incorrect |

**Вариант 5**

**Задача 1**

Рассмотрим бесконечный лист клетчатой бумаги. Закрасим некоторое множество клеток в черный цвет. Теперь мы хотим закрасить минимальное количество клеток, так, чтобы множество черных клеток стало выпуклым.

Напомним, что геометрическая фигура Φ называется выпуклой, если для любых точек A из Φ и В из Φ с вещественными координатами отрезок [AB] принадлежит Φ.

**Входные данные**

В первой строке входных данных содержатся два числа N и M (1 ≤ N, M ≤ 100) — размеры куска бумаги, куда попали все черные клетки. В каждой из следующих N строк содержится М символов «\*» или «.». Символ «\*» обозначает черную клетку, а «.» белую.

**Выходные данные**

Выведите выпуклое множество, содержащее минимальное количество дополнительно покрашенных черных клеток, в ровно N строках по M символов «\*» или «.» в каждой.

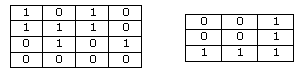
**Примеры**

|  |  |  |
| --- | --- | --- |
| **№** | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 2 4 ..\*. .\*\*. | .\*\*. .\*\*. |
| 2 | 4 3 .\*. .\*. .\*. .\*. | .\*. .\*. .\*. .\*. |

**Задача 2**

Рассмотрим таблицу, содержащую n строк и m столбцов, в каждой клетке которой расположен ноль или единица. Назовем такую таблицу симпатичной, если в ней нет ни одного квадрата 2 на 2, заполненного целиком нулями или целиком единицами.

Так, например, таблица 4 на 4, расположенная слева, является симпатичной, а расположенная справа таблица 3 на 3 - не является.



Задано несколько таблиц. Необходимо для каждой из них выяснить, является ли она симпатичной.

**Входные данные**

Первая строка входных данных содержит количество t (1 ≤ t ≤ 10) наборов входных данных. Далее следуют описания этих наборов. Описание каждого набора состоит из строки, содержащей числа n и m (1 ≤ n,m ≤ 100), и n строк, каждая из которых содержит по m чисел, разделенных пробелами. j-ое число в i+1-ой строке описания набора входных данных - элемент aij соответствующей таблицы. Гарантируется, что все aij равны либо нулю, либо единице.

**Выходные данные**

Для каждого набора входных данных выведите единственную строку, содержащую слово «YES», если соответствующая таблица является симпатичной, и слово «NO» - в противном случае.

**Пример**

|  |  |  |
| --- | --- | --- |
| **№** | **INPUT.TXT** | **OUTPUT.TXT** |
| 1 | 3 1 1 0 4 4 1 0 1 0 1 1 1 0 0 1 0 1 0 0 0 0 3 3 0 0 1 0 0 1 1 1 1 | YES YES NO |

**Вариант \***

**Задача 1. Уравнение.**

Даны целые числа A, B, C. Рассмотрим все пары целых чисел (x, y) такие, что Ax + By = C, x ≥ 0 и y ≥ 0. Найдите максимум x + y по всем таким примерам или скажите, что таких пар нет.

*Входные данные.*

В первой строке входных данных записано целое число T (1 ≤ T ≤ 104) – количество тестов. Далее следуют описания T тестов, каждое – на одной строке. Описание теста состоит из трех целых чисел A, B, C (1 ≤ A, B, C ≤ 106) – коэффициент уравнения.

*Выходные данные.*

Выведите T строк. В i-й строке выведите одно целое число – ответ на i-й тест: максимум x + y по всем парам (x, y) из условия, если хотя бы одна пара удовлетворяет всем необходимым условиям, и -1 – иначе.

*Примеры.*

|  |  |
| --- | --- |
| **Стандартный ввод** | **Стандартный вывод** |
| 5  1 1 123456  1 2 9  2 2 41  99 82 44353  1 79997 1000000 | 123456  9  -1  527  1000000 |

**Задача 2**

Вам задан неориентированный граф с N вершинами и M ребрами (1 ≤ N ≤ 20000, 1 ≤ M ≤ 200000). В графе отсутствуют петли и кратные ребра. Определите компоненты связности заданного графа.

*Входные данные.*

Граф задан следующим образом: первая строка содержит числа N и M. Каждая из следующих M строк содержит описание ребра – два целых числа из диапазона от 1 до N – номера вершин - концов ребра

*Выходные данные.*

В первой строке выведите число L – количество компонент связности заданного графа. На следующей строке выведите N чисел их диапазона от 1 до L – номера компонент связности, которым принадлежат соответствующие вершины. Компоненты связности следует занумеровать от 1 до L произвольным образом

Примеры.

|  |  |
| --- | --- |
| **Стандартный ввод** | **Стандартный вывод** |
| 4 2  1 2  3 4 | 2  1 1 2 2 |