Practice Arena

Practice problems aimed to improve your coding skills.

The Post offices of KRville

LAB-PRAC-12_FUN-STRUC

The Post offices of KRville [20 marks]

Problem Statement
Mr C lives in the country of KRville, named after Kernighan and Ritchie, the developers of the C language. Cities in KRville are arranged on a 2D grid with M rows and N columns. Some of the cities have a post office (PO) in them and some do not. All PO have a 6 digit PIN number associated with them. The PIN is always strictly positive and the first digit of the PIN is never 0.

Two towns are said to be neighbours of each other if they lie on adjacent rows as well as adjacent columns. Thus, if M = 5 and N = 5 the town at index (2,2) has 8 neighbours (1,1), (1,2), (1,3), (2,1), (2, 3), (3,1), (3,2), (3,3). As a visual example, the town numbered 5 in the matrix below has neighbors as the towns numbered 1, 2, 3, 4, 6, 7, 8, 9.

1 2 3
4 5 6
7 8 9

1. If a town T1 has its own PO with pin P1 then T1 is serviced by P1.
2. If a town T2 doesn't have a post office of its own and the neighbours of T2 do not have a post office either then T2 is not serviced at all.
3. If a town T3 doesn't have its own PO but one of its neighboring towns has a PO then T3 is serviced by the neighbouring PO. In case multiple neighboring towns have a PO, then the PO with the smallest PIN code will serve T3.

In the first line of the input we will give you M and N which denotes number of rows and columns in the grid. In the second line, you will be given P, which is the number of post offices in the country. In the next P lines, you will be given location of these post-offices and the corresponding PIN numbers as three strictly positive numbers

X Y PIN

X will be the row number of the PO, Y will be the column number of the PO and PIN will be its PIN number. Note that the row numbers and column numbers we give you will start from one i.e. they will not be like C array indices which start from 0.

In the first line of the output, in the format given below. print the number of towns which are not serviced by any PO. In the next line, you have to print the PIN code of the post-office which serves the maximum number of towns. If there are multiple such POs, print the PIN of the post-office with smallest PIN.

Caution

1. Follow the output format closely. Do not make spelling, space, capitalization errors.
2. The row and the column numbers will be given to you in human readable form i.e. the first row will have row number 1. Note that these are not C indices that start from zero. These are human readable location numbers.

3. If the town is a corner town then it has only 3 neighbours. If a town is an edge town, it has 5 neighbours. Otherwise, each town has eight neighbours.
4. We will never give you input where two different PO have the same pin number. Different PO have different pins. All pins are non-negative integers.
5. Do not wrap around the matrix in search for a post office. For example, if M = 3 and N = 5, then the post office at index (1,0) cannot serve the town at index (1,4). Similarly the post office at index (2,2) cannot serve the town at index (0,2). Only postoffices that are physically adjacent to a town can serve a town in case that town does not have a post office of its own.
6. Be careful about extra/missing lines and extra/missing spaces in your output.

HINTS: You may solve this problem using structures (although it is not compulsory). Create a structure

```
struct town{
    int POInTown;
    int POServingTown;
    int numTownsServedbyPO;
};
```

where POInTown denotes the PIN code of the post office if one is located in the town itself. If no post office is there in the town, then this variable can be -1. POServingTown tells us which post office serves this town. If there is a post office in town then obviously POServingTown = POInTown. numTownsServedbyPO tells us how many towns does the post office in this town serve. If this town does not have a PO then this number can be -1.

Create a 2D matrix of these structures
struct town[M][N];
and work with it to solve the problem.

EXAMPLE:

```
INPUT
3 3
2
1 1 208016
3 3 200001

OUTPUT:
NO PO: 2
BUSIEST PO: 200001
```

Explanation: The country looks like this (a * indicates no PO in that town)

```
208016 * *
   *     * *
   *    * 200001
```

There are two towns without any PO access. 20001 is the busiest PO serving 4 towns. 208016 serves only 3 towns.

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There are two lines in your output. Printing each line correctly, in the correct order, carries 50% weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

Start Solving! (/editor/practice/6234)