DAY 2
Overview

Author	Bruce	Carl	Carl
Problem	**Power**	**Sliding tiles**	**Area**
Program name	`power`	`tiles`	`area`

Source name	`power.pas`
	`power.java`
	`power.cpp`
		`tiles.pas`
		`tiles.java`
		`tiles.cpp`
			`area.pas`
			`area.java`
			`area.cpp`

Input file	`power.in`	`tiles.in`	`area.in`
Output files(10)	`power.out`	`tiles.out`	`area.out`
Time limit	1 second	3 seconds	10 seconds
Num. of tests	10	10	10
Points per test	10	10	10
Total points	**100**	**100**	**100**

The maximum total score for Day2 is 300 points.

Power

Author

Bruce Merry

Introduction

In order to break into a building, you need to cut the power to it so that the alarms will not go off. The backup power supply will be taken care of, but it is up to you to take out the main electricity supply to the building.

The electricity grid is a network of one-way flows between relay stations. For redundancy, there may be more than one way for electricity to get from one point to another (including loops). You have enough explosive to take out one relay station (unfortunately they have fixed the security hole, so you can't just hack into their systems). However, you wish to avoid attracting attention, so you should blow up the relay station that causes the smallest power loss to the city.

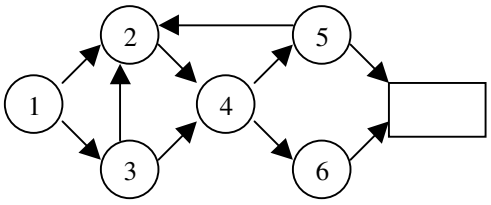
Task

There are *N* relay stations, numbered 1 to *N*. The power station feeds electricity directly to relay 1 (so blowing up relay 1 will black out the entire city). You will be given a list of one-way electricity flows between stations. A relay will receive power only if there is a path from relay 1 to it through (undamaged) relays. The building will receive power if it is connected to a relay that receives power.

You must choose a relay to blow-up, so that as many other relays as possible continue to receive power, while still cutting off power to the building.
In the event of a tie, you may choose freely.

Example

The circles represent relay stations and the rectangle represents the building.



Taking out either relay 1 or relay 4 will cut off the power to the building. However, taking out relay 1 will cut off all the power in the network, while taking out relay 4 will only cut the power to relays 4, 5 and 6.

Input (power.in)

The first line of input contains *N*, the number of relay stations. The second line contains *C*, the number of connections in the network (including connections to the building). The following C lines each contain two integers *X* and *Y*, indicating that relay *X* provides power to relay *Y*. However, if *Y* is 0, then relay *X* provides power to the building.

Sample Input:

```
6
10
1 2
2 4
4 5
5 0
1 3
3 4
4 6
6 0
2 3
5 2
```

Output (power.out)

The output consists only of a single integer. This is the number of the relay station to blow up.

Sample output:

```
4
```

Constraints

1 • *N* • 1000
1 • *C* • 10000
No relay is directly connected to itself, and the connections are all unique. The building initially receives power.

Time limit

1 second.

Scoring

Your score will be the first of these that applies:

- 0% if your output is invalid;
- 0% if you did not cut off the power to the building;
- 100% if you cut the power to as few relays as possible;
- 10% if you took out relay 1;
- 50% otherwise.

Sliding Tiles

Author

Carl Hultquist

Introduction

As a homework assignment, one of your friends has been given what is known as a *sliding tile puzzle* to solve. This is simply a square grid of tiles, with one tile missing. The tiles have the number **1** to $\mathbf{N^2-1}$, and the goal of the puzzle is to use the blank space to move the tiles around into a specific order. In particular, the tiles must be in ascending order starting from the top-left corner, moving from left to right and then continuing at the left side of the next row down. See the figure below for a graphical illustration:

1	**2**	**…**	**N-1**	**N**
N+1	**N+2**	**…**	**2N-1**	**2N**
…	**…**	**…**	**…**	**…**
$\mathbf{N^2-2N+1}$	$\mathbf{N^2-2N+2}$	**…**	$\mathbf{N^2-N-1}$	$\mathbf{N^2-N}$
$\mathbf{N^2-N+1}$	$\mathbf{N^2-N+2}$	**…**	$\mathbf{N^2-1}$	

Task

Given a sliding tile puzzle of size **N** that is in a jumbled state, find the shortest possible series of moves that will slide the tiles into the correct order.

Example

Given the following puzzle:

1	**2**	**3**
	4	**6**
7	**5**	**8**

the tiles could be placed in the correct order in 3 moves, by first moving tile **4**, then tile **5**, and finally tile **8**.

Input (tiles.in)

The first line of **tiles.in** will contain a single integer, **N**, indicating the size of the puzzle. The following **N** lines will each contain **N** integers describing the initial state of the puzzle. The integer with value $\mathbf{N^2}$ will indicate where the blank square of the puzzle is.

Sample Input:

```
3
1 2 3
9 4 6
7 5 8
```

Output (tiles.out)

The first line of **tiles.out** must contain a single integer, **L**, indicating the length of the solution that you have found. The following **L** lines of **tiles.out** must each contain a single integer, indicating the tile that is to be moved.

Sample output:

```
3
4
5
8
```

Constraints

- 3 • **N** • 25
- There is always guaranteed to be a solution for any of the test cases used.

Time limit

3 seconds

Scoring

If your output differs in any way from the above description, for any test case, then your score will be 0 points for that test case. If, after performing all the moves specified by your output for a test case, the tiles are not in the correct order, then you score 0 points for that test case.

Suppose the evaluation program finds a solution with **M** moves for a test case.

- If **L** • **M**, then you score 10 points for the test case.
- Otherwise, if **M** × 1.05 • **L** (that is, your solution is within 5% of the evaluation program' s solution), then you score 8 points for the test case.
- Otherwise, if **M** × 1.075 • **L** (that is, your solution is within 7.5% of the evaluation program' s solution), then you score 5 points for the test case.
- Otherwise, if **M** × 1.1 • **L** (that is, your solution is within 10% of the evaluation program' s solution), then you score 2 points for the test case.
- Otherwise, you score 1 point for the test case.

Area

Author

Carl Hultquist

Introduction

Jack desperately wants to marry his school sweetheart, Jill. But before Jill will consent to marriage, she sets a hard puzzle for Jack so that he can show how serious he is about marriage.

Jill takes all the rectangular love letters that Jack has ever sent her, and scatters them on the floor, with some letters overlapping others. For Jack to pass the test, he needs to work out what area of the floor is covered by letters.

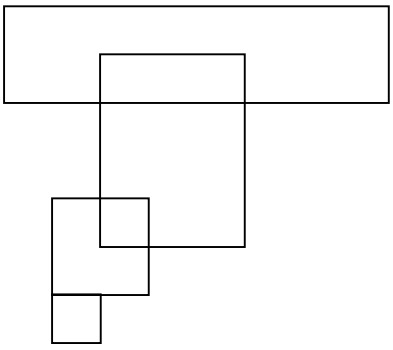
Task

Given the positions of all the rectangular letters on the floor, find the total area of floor covered. Note that letters may overlap, meaning that some areas of the floor might be covered by more than one letter.

Example

Given four letters with the following sets of top-left and bottom-right co-ordinates – in **(x; y)** format:

(1; 2) to **(3; 4)**
(1; 1) to **(2; 2)**
(0; 6) to **(7; 8)**
(2; 3) to **(5; 7)**



The total area of the floor covered by letters is 27. This occurs since the sum of the areas of the letters is 31, but 4 of these units are comprised of overlapping letters.

Input (area.in)

The first line of **area.in** will contain a single integer, **N**, which specifies the number of rectangles that Jill has thrown onto the plane. The next **N** lines will each contain 4 integers, $\mathbf{x_1}$, $\mathbf{y_1}$, $\mathbf{x_2}$ and $\mathbf{y_2}$, which specify the bottom left and top right corners of each rectangle. So ($\mathbf{x_1}$; $\mathbf{y_1}$) specifies the bottom left corner and ($\mathbf{x_2}$; $\mathbf{y_2}$) specifies the top right corner.

Sample Input:

```
4
1 2 3 4
1 1 2 2
0 6 7 8
2 3 5 7
```

Output (area.out)

area.out should contain a single integer which specifies the area occupied by the **N** rectangles in the input.

Sample output:

```
27
```

Constraints

- $0 <= \mathbf{x_i}$; $\mathbf{y_i} <= 10000$
- $2 <= \mathbf{N} <= 10000$

Time limit

6 seconds

Scoring

If your program produces the correct answer for a test case, then you will score 10 points for that test case. Otherwise, you will score 0 points for that test case.