Homework 1

Homework 1-1: Line Segment Intersection Detection (30 points)

Create a python program that accepts user inputs of four x-y coordinates. These coordinates represent the endpoints of two line segments in a two-dimensional space.
The program should analyze whether these two line segments intersect.
Display the output as “1” if there is an intersection between the line segments, and “0” if there is no intersection.
Implement your solution without using any pre-existing Python modules specifically designed for this purpose.

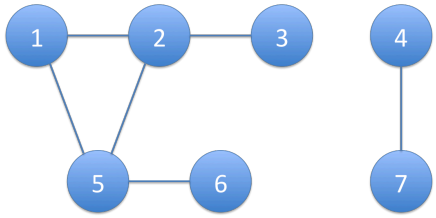
Input: The number of test cases followed by sets of four x-y coordinates. Each set of coordinates represents two endpoints for the first line segment and two endpoints for the second line segment.
Output: A single digit, either “0” or “1”, indicating the absence or presence of an intersection between the two line segments.
Example:

Input File: **input.txt**
4
7 8 5 12 10 10 3 10
5 10 5 2 2 1 8 7
10 10 3 10 6 3 6 6
5 4 8 6 10 6 5 1

Output File: **output.txt**
1
1
0
0

Homework 1-2: Computer Virus (35 points)

Create a Python program to calculate the total number of computers infected by a virus in a network. The virus propagates across the network, infecting any computer connected to an already infected machine. Consider a network of seven computers connected as illustrated in the provided diagram. If computer #1 gets infected, computers #2, #3, #5, and #6 also become infected as a result of their connections. Computers #4 and #7 remain unaffected. Your task is to develop an algorithm in Python that determines the total count of infected computers in such scenarios.



Input:

In the first line, the number of computers is given and it is less than 100.
The second line is the number of edges that represent a connection of two computers.
After the second line input, each line has two numbers that represent two computers that are connected to each other.
In the last line, the computer number is given as the first infected computer.

Output:

Total number of computers that will be infected through the network by the first infected computer.

Example
Input File: **input.txt**

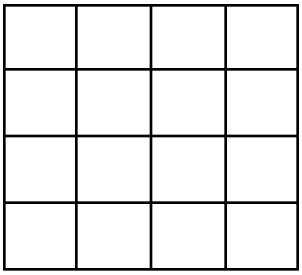
```
7
6
1 2
2 3
1 5
5 2
5 6
4 7
1
```

Output File: **output.txt**

```
4
```

Homework 1-3: Civilization (35 points)

Think of the world as a two-dimensional space where there are `NxN` squares. For example, if `N=4`, it is like below.

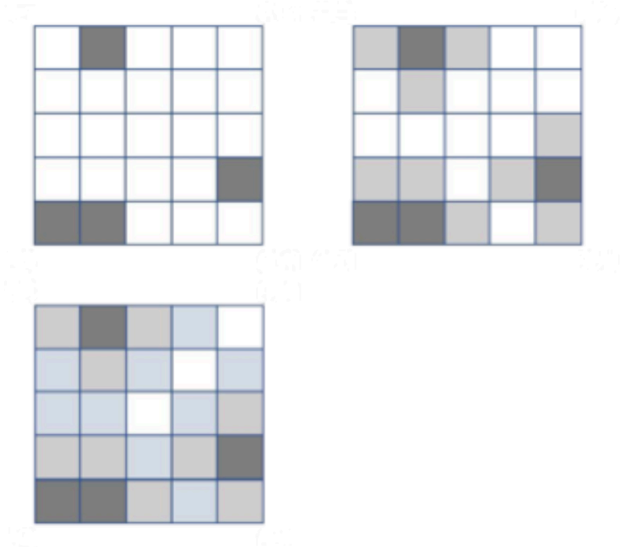


The square at the top left is in the (`0, 0`) position. The square at the bottom right is in the (`N-1, N-1`) position. Two squares (`a, b`) and (`a ', b'`) are adjacent to each other when both of the following two conditions are satisfied.

Condition 1: `|a-a'| = 1 and b=b'`
Condition 2: `|b-b'| = 1 and a=a'`

The first birthplace of civilization is all in different `K` squares. The space of each square is civilized or undeveloped. The first birthplace of civilization is a civilization area. Every year, civilization spreads civilization to its neighbor (adjacent squares). If the square (`a, b`) is a civilized area, civilization will spread to the square next to this square, unless the boundary of the world is exceeded in the next year.

This is an example.



Write a program to estimate years for all civilizations to be connected through adjacent areas (squares).

Input: In the first line, there are two numbers, `N` and `K` as world size and number of civilizations. In the next `K` lines, there are two numbers `x` and `y`, one on each line, indicating the position of the square corresponding to the birthplace of the civilization (`1 ≤ x, y ≤ N`).
Output: years to connect all civilizations.

Example 1
input.txt

```
10 3
0 2
0 7
0 9
```

output.txt

```
2
```

Example 2
input.txt

```
5 4
0 4
1 4
1 0
4 3
```

output.txt

```
2
```