Question 1:

There are given lines, find out which lines are parallel to each other.

USE CASE: At the runtime we need to query out the lines of a particular slope.

Use any data structure or algorithm of your choice to handle the use case in most effective

manner.

Use lines\_question1.json

Question 2:

Given a list of line\_segments your task is to create a graph and a directed graph in the

networkx from the line\_segments (HINT: treat them as edges of the graph.)

NOTE: There is a node which does not have any inedge, this node will become the head

of the graph. Use this to construct the digraph.

Return the output of the traversal along with the code.

Use edges.json

Question 3:

From the graph you constructed in question 2 perform the following operations:

- Undirected - graph:

1. Count the number of nodes with degree 2.

2. Out of those nodes found in the step 1, give the property of:

a. color: red

b. And to all the other nodes with degrees greater than 2 give color : blue.

c. Give the head node the color : green.

3. Now find out the count of largest consecutives blue colored nodes.

- Directed - graph:

1. Mark the nodes with out-degree 2 and in-degree as color : blue.

2. Those edges which are making an angle of 90 degree with the blue colored

nodes, give those edges the property of ‘label’ : ‘perpendicular’.

1. Also show these labeled and colored edges on the graph.

Question 4:

- Given a Matrix consisting of 0s and 1s. Find the number of islands of connected 1s

present in the matrix.

- Note: A 1 is said to be connected if it has another 1 around it (either of the 8

directions).

- **Input:**

- The first line of input will be the number of testcases **T** , then T test cases follow.

The first line of each testcase contains two space separated integers N and M.

Then in the next line are the NxM inputs of the matrix A separated by space .

- **Output:**

- For each testcase in a new line, print the number of islands present.

- **Your Task:**

- You don't need to read input or print anything. Your task is to complete the

function **findIslands()** which takes the matrix A and its dimensions N and M as

inputs and returns the number of islands of connected 1s present in the matrix. A 1

is said to be connected if it has another 1 around it (either of the 8 directions).

**Example(To be used only for expected output) :**

- **Input**

- 2

- 3 3

- 1 1 0 0 0 1 1 0 1

- 4 4

- 1 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0

- **Output**

- 2

- 2

- **Explanation** :

- **Testcase 1:** The graph will look like

- 1 1 0

- 0 0 1

- 1 0 1

- Here, two islands will be formed

- First island will be formed by elements {A[0][0] , A[0][1], A[1][2], A[2][2]}

- Second island will be formed by {A[2][0]} **.**

- **Testcase 2:** The graph will look like

- 1 1 0 0

- 0 0 1 0

- 0 0 0 1

- 0 1 0 0

- Here, two islands will be formed

- First island will be formed by elements {A[0][0] , A[0][1], A[1][2], A[2][3]}

- Second island will be formed by {A[3][1]} **.**