# PROGRAMMING TASK

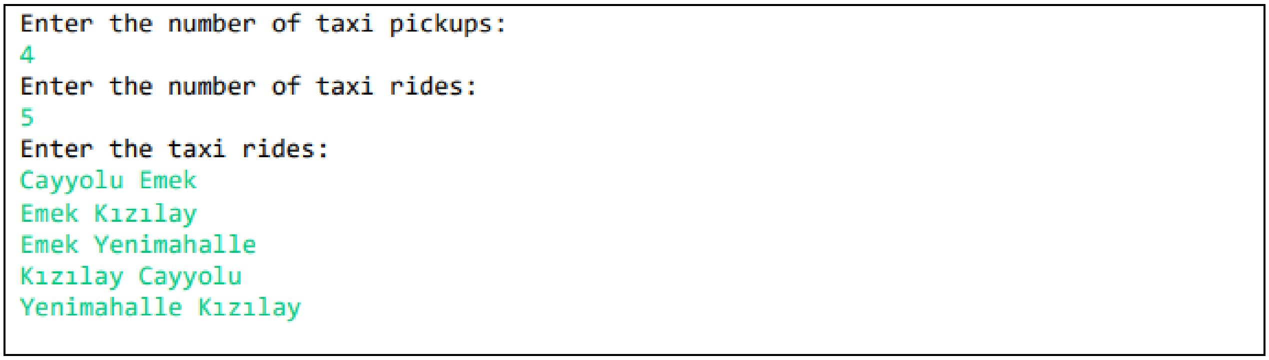
**In this part, you must implement your own graph data structure by taking inspiration from your textbook and using it to help to solve the problem. You are not allowed to use any external library or .jar file. Any solutions without using graph data structure are not evaluated!**

### Question 1(25 points):

You started as a software developer in the Uber company. In this company, a special ride network is created every day. And your first task is to prove whether you can keep this company's ride network in a tree structure. But the developer who left the company before you has already designed this network using a directed graph. Your task is to measure whether this directed graph can be a tree. After some research, you found that for a directed graph to be a tree, it must have one less number of edges than the number of vertices, it must be connected, and there must be no cycles in the graph. Now prove to your team leader that you can solve this problem.

**Note:** Here you must use the symbol chart data structure when retrieving taxi pickups from the user.

**Sample Input:**



```
Enter the number of taxi pickups:
4
Enter the number of taxi rides:
5
Enter the taxi rides:
Cayyolu Emek
Emek Kızılay
Emek Yenimahalle
Kızılay Cayyolu
Yenimahalle Kızılay
```

The person who will use your program must first enter the number of taxi pickups in the ride network. Then it should enter how many taxi rides there are connecting these taxi pickups. Finally, it should specify line by line from which taxi pickup to which one can go. For example, in the first line, it is stated that there is a one-way departure from Çayyolu to Emek.

**Sample Output:**

```
Cayyolu:  Emek
Emek:  Yenimahalle Kızılay
Kızılay:  Cayyolu
Yenimahalle:  Kızılay
This ride network cannot be kept in a tree structure.
```

In the program output, show the user to which taxi pickups a taxi in this ride network can go from its location. For example, in line 1, it is stated that there are only departures from Çayyolu to Emek. And in the last line, print the information on whether this ride mesh can be kept in a tree structure.

**Question 2 (25 points):**

All farmland in a town will be harvested by a company. The company first needs to calculate the minimum cost of navigating between the farms. Since these farmlands are all next to each other, information is collected from the landowners in the town and kept in a grid structure with N rows and M columns. You are asked to write a program for this company. This program uses the **MST (Minimum Spanning Tree)** algorithm to output a single integer expressing the minimum cost of good trip in a grid structure.

Let your program first get the number of test cases from the user. For each test case, ask for the number of rows and columns in the grid. Then calculate the cost of an edge e connecting any two cells with values a and b as follows:

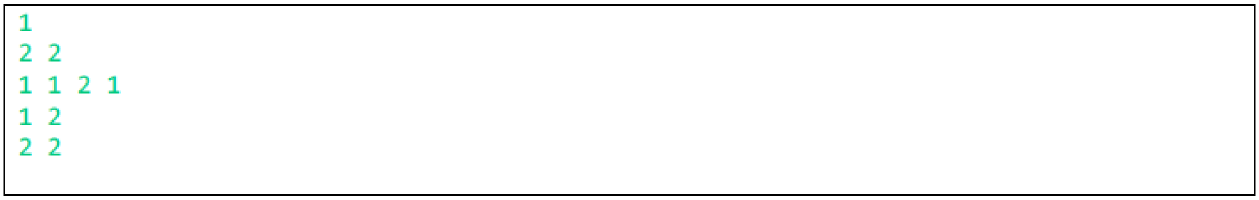
$E(e)=a \oplus b$ *(where $\oplus$ denotes the bitwise xor operation of integers a and b)*

We define good trip between two cells (u1,v1) and (u2,v2) as a trip starting at cell (u1,v1) and ending at cell (u2,v2) while visiting every cell of the grid at least once. For a given edge e, if you visit this edge T(e) times then the cost of the trip is:

$\sum_{(\forall e)} \; ( E(e) * ( T(e) / 2 )$ *(here, T(e) / 2 is the ceiling of the result of the division of T(e) by 2)*

Now please find the minimum path cost of harvesting all farmland for the given starting cell (u1,v1) and ending cell (u2,v2).

**Sample Input:**



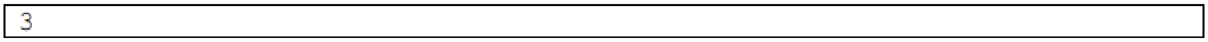
```
1
2 2
1 1 2 1
1 2
2 2
```

Here, the first line is the number of test cases and the next line is the grid size for that test case (N X M).

The line expressed by line 3 for this sample case is will consist of four integers, r1,c1,r2,c2, denoting the coordinates (r1,c1) of the starting cell and (r2,c2) of the ending cell.

Finally, N lines will follow, each containing M integers, denoting the values in the grid, such that the j.value in the  i.row will denote the number in the cell (i, j) of the grid.

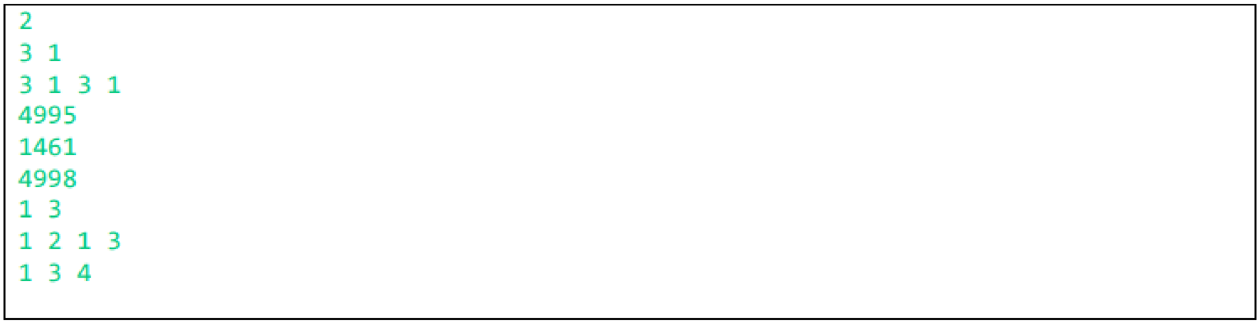
**Sample Output:**



```
3
```

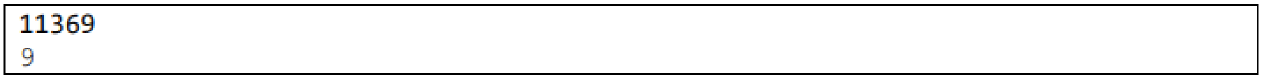
It outputs a single integer indicating the minimum cost of trip for each of the test cases, one after the other.

**Sample_2 Input:**



```
2
3 1
3 1 3 1
4995
1461
4998
1 3
1 2 1 3
1 3 4
```

**Sample_2 Output:**



```
11369
9
```