**City group**

Max. Marks: 100

You are living in a town consisting of N cities. Furthurmore, in this town there are K city-groups. You can reach any city from any city in a same city-group instantaneously. you can go from any city in ith city-group to any city in i+1th city-group in 1 second and from any city in i+1th city-group to any city in ith city-group in 1 second for each i between 1 and K-1, you can also go from any city in first city-group to any city in Kth city-group in 1 second and from any city in Kth city-group to any city in first city-group in 1 second.

You have been given Q queries each containing two cities X and Y. For each query, you have to print the minimum time it takes to reach city Y from city X.

Each city-group has a city which does not have a number (i.e. it is not one of the N cities mentioned above). you can visit those cities in the middle of your trip between cities X and Y given in queries.

**INPUT:**  
First line of input will consists of two integers N and K denoting total number of cities and city groups. Next K lines will consists of information regarding city-groups. First integer in these K lines will consists of number of cities Si belonging to that city-group. Next Si integers for each line will consists of cities belonging to ith city-group. Next line will consists of integer Q denoting total number of queries. Next Q lines will consists of two cities X and Y.

**OUTPUT:**   
For each of the query, print minimum time needed to reach city Y from city X.

**CONSTRAINTS:**  
1 ≤ N,K,Q ≤ 105  
0 ≤ Si ≤ N  
Its guaranteed that each city will belong to exactly one city group.

**SAMPLE INPUT**

10 6

0

1 1

2 2 4

0

2 3 5

5 6 8 7 9 10

4

7 10

1 2

1 5

6 2

**SAMPLE OUTPUT**

0

1

3

3

**Explanation**

first query: both cities 7 and 10 are in same city group, so you need 0 time to go from city 7 to city 10

second query: city 1 is in second city-group and city 2 is in third city-group so distance between them is 1

third query: city 1 is in second city-group and city 5 is in 5-th city-group so distance between them is 3

fourth query: city 2 is in third city-group and city 6 is in 6-th city-group so distance between them is 3

**Time Limit:**1.0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded if any testcase passes.

**Allowed Languages:**C, C++, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala 2.11.8, Swift, Visual Basic

<https://www.hackerearth.com/challenge/competitive/may-circuits-17/algorithm/city-group-3/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

class MyClass

{

static void Main(string[] args)

{

// int n = 10, k = 6;

string[] nk = Console.ReadLine().Split(' ');

int n = int.Parse(nk[0]);

int k = int.Parse(nk[1]);

int grupo = 1;

Dictionary<int, int> grupos = new Dictionary<int, int>();

for (int i = 0; i < k; i++)

{

int[] arr = Array.ConvertAll(Console.ReadLine().Split(' '), e => int.Parse(e));

if (arr[0] != 0)

{

for (int j = 1; j < arr.Length; j++)

{

grupos[arr[j]] = grupo;

}

}

grupo++;

}

int q = int.Parse(Console.ReadLine());

while (q-- > 0)

{

string[] xy = Console.ReadLine().Split(' ');

int x = int.Parse(xy[0]);

int y = int.Parse(xy[1]);

int max = Math.Max(grupos[x], grupos[y]);

int min = Math.Min(grupos[x], grupos[y]);

Console.WriteLine(Math.Min( Math.Abs(grupos[x] - grupos[y]), min + (grupo - max) - 1));

}

Console.ReadLine();

}

}