In ABC company, team lead need to discuss about a new project. So that he gathered the team members at a place. One of the team member couldn't understand the theme is willing to reach out his colleague in shortest time period.

Note: The meeting room has been modelled as a directed graph.

Input Format:

First line is a positive integer n, 1<=n<=20 denoting the number of employees in the meeting.
Next consecutive n lines are the unique id's of employees.
Next line consists of a positive integer s, denoting all the ways to reach from one employee to another employee
Next consecutive s lines, contains two employees that are adjacent with some distance.
Next line consists of the id of the employee who got a doubt.
Last line consists of the id of the employee who will clarify doubt.

Output Format:

1; if an employee can reach another employee.
0; if an employee can't reach another employee.

Sample Input:

```
5
2
5
7
6
9
5
2 9
7 2
7 9
6 5
9 5
2
9
```

Sample Output:

```
1
```

Solution:

```
graph = [[] for _ in range(100000)]

def addEdge(S, D):
    graph[S].append(D)

def neighbour(S, V ):
    node=[]
    if (len(graph[S]) > 0):
        for i in range(len(graph[S])):
            node.append(str(graph[S][i]))
        return " ".join(node)
    else:
        return -1
v = int(input())
for i in range(v):
    h=int(input())

e = int(input())

for i in range(e):
    x,y = map(int,input().split())
    addEdge(x,y)
S = int(input())
D=int(input())
l=neighbour(S,v)
if str(D) in l:
    print(1)
else:
    print(0)
```