import numpy as np

import math

file = open('C:\\Users\\User\\Desktop\\number1.txt','r')

line1=file.readline().strip()#gets the first line

vertex\_count=int(line1)#9

#creating adjacency matrix

adj\_matrix= np.zeros((vertex\_count, vertex\_count), dtype='int')

#print(adj\_matrix)

line2=file.readline().strip()#gets number of connections

connections=int(line2)#gets 13

for i in range(connections):#loop runs 13times: 0-12

line=file.readline().strip()

vertices=line.split(' ')

u=int(vertices[0])

v=int(vertices[1])

adj\_matrix[u][v]=1

adj\_matrix[v][u]=1

#print(adj\_matrix)

color=np.empty(vertex\_count, dtype= 'object')

color[:]='WHITE'

#parent=np.empty(vertex\_count,dtype='object')

#parent[:]=np.NaN

d=np.zeros(vertex\_count,dtype='int')

d[:]=math.inf

from queue import Queue

my\_queue=Queue(maxsize=vertex\_count)

def bfs(s):

color[s]='GRAY'

#parent[s]=np.NaN

d[s]=0

my\_queue.put(s)

while not my\_queue.empty():

u=my\_queue.get()

for v in range(vertex\_count):

if adj\_matrix[u][v] == 1:

if color[v] == 'WHITE':

color[v]='GRAY'

d[v]=d[u]+1

#parent[v]=u

my\_queue.put(v)

color[u] = 'BLACK'

return

#running

bfs(0)

lastline=file.readline().strip()

linapos=int(lastline)#linas's position

#print('position of lina is '+str(linapos))

print(str(d[linapos]) )#Minimum number of moves Nora needs to go to ‘x’

#number2------------------------------

file = open('C:\\Users\\User\\Desktop\\number2.txt','r')

line1=file.readline().strip()#gets the first line

vertex\_count=int(line1)#9

#creating adjacency matrix

adj\_matrix= np.zeros((vertex\_count, vertex\_count), dtype='int')

#print(adj\_matrix)

line2=file.readline().strip()#gets number of connections

connections=int(line2)#gets 12

for i in range(connections):#loop runs 12times: 0-11

line=file.readline().strip()

vertices=line.split(' ')

u=int(vertices[0])

v=int(vertices[1])

adj\_matrix[u][v]=1

adj\_matrix[v][u]=1

#print(adj\_matrix)

linapos=int( file.readline().strip() )#lina's position

norapos=int( file.readline().strip() )#nora's position

larapos=int( file.readline().strip() )#lara's position

color=np.empty(vertex\_count, dtype= 'object')

color[:]='WHITE'

d=np.zeros(vertex\_count,dtype='int')

d[:]=math.inf

my\_queue=Queue(maxsize=vertex\_count)

bfs(norapos)

nora\_distance=d[linapos]

#for i in d:

# print(i)

color=np.empty(vertex\_count, dtype= 'object')

color[:]='WHITE'

d=np.zeros(vertex\_count,dtype='int')

d[:]=math.inf

my\_queue=Queue(maxsize=vertex\_count)

bfs(larapos)

lara\_distance=d[linapos]

#for i in d:

#print(i)

if nora\_distance<lara\_distance:

print('Nora')

elif lara\_distance<nora\_distance:

print('Lara')

else:

print('both')

#number3----------------------------

file = open('C:\\Users\\User\\Desktop\\number3.txt','r')

line1=file.readline().strip()#gets the first line

vertex\_count=int(line1)#10

#creating adjacency matrix of reverse graph

adj\_matrix= np.zeros((vertex\_count, vertex\_count), dtype='int')

#print(adj\_matrix)

line2=file.readline().strip()#gets number of connections

connections=int(line2)#gets 14

for i in range(connections):

line=file.readline().strip()

vertices=line.split(' ')

u=int(vertices[0])

v=int(vertices[1])

adj\_matrix[v][u]=1

#print(adj\_matrix)

linapos=int( file.readline().strip() )

color=np.empty(vertex\_count, dtype= 'object')

color[:]='WHITE'

d=np.zeros(vertex\_count,dtype='int')

d[:]=math.inf

my\_queue=Queue(maxsize=vertex\_count)

bfs(linapos)

total\_participants=int( file.readline().strip() )

min\_distance=math.inf

p\_no=-2#participant number

for j in range(total\_participants):#0-4

pos=int( file.readline().strip() )

dist=d[pos]#distance of lina from pos

if dist<min\_distance:

min\_distance=dist

p\_no=j+1

print(min\_distance)