

```

# NAME: Mantej Lamba
# ID: 2259394598
# DATE: 2022-10-09
# DESCRIPTION: wrote a Python program to parse a file and give recommendations similar to a Facebook network

from typing import IO, List

def open_file() -> IO:
    # Add comments
    file_name = input("Enter a filename: ") # asking for user file name
    file_pointer = None # creating a file pointer
    while file_pointer is None: # keeps looping until proper file name is given
        try:
            file_pointer = open(file_name, "r") # opening the file given
        except IOError:
            print("Error in filename.") # reasking for file name is the file name is not valid
            file_name = input("Enter a filename: ")

    return file_pointer # returning file pointer

def read_file(fp: IO) -> List[List[int]]:
    # Add comments
    n = fp.readline() # reading the file line
    n = int(n) # first value is the number of users in the social network
    network = [] # creating list
    for i in range(n):
        network.append([]) # adding the users to a list

    line = fp.readline() # reading the next line
    while line is not None and len(line) >= 3: # checking the line for specific parameters
        split_line = line.strip().split(" ") # splitting the line
        network[int(split_line[0])].append(int(split_line[1])) # get the user id values to hold the friends for each values
        network[int(split_line[1])].append(int(split_line[0]))
        line = fp.readline() # reading the next line

    return network # returning list

def init_matrix(n: int) -> List[List[int]]:
    # Add comments
    matrix = [] # initializing empty list
    for row in range(n): # iterating through n
        matrix.append([]) # appending empty list
        for column in range(n):
            matrix[row].append(0) # intializing with 0
    return matrix # returning matrix

def num_in_common_between_lists(list1: List[int], list2: List[int]) -> int:
    common = 0 # intializing common variable
    for i in list1: # looping through list 1
        if i in list2: # checking if item in list 1 is in list 2
            common+=1 # if it is, then incrementing common

    return common #returning common

def calc_similarity_scores(network: List[List[int]]) -> List[List[int]]:
    length = len(network) # getting the length of network
    similarity_matrix = init_matrix(length) # creating an empty matrix
    for i in range(length): # looping through the network to access each value
        for j in range(length):
            similarity_matrix[i][j] = num_in_common_between_lists(network[i],network[j]) # finding the number in common
    return similarity_matrix

def recommend(user_id: int, network: List[List[int]], similarity_matrix: List[List[int]]) -> int:
    similarity_matrix = calc_similarity_scores(network) #creating similarity matrix
    listScores = similarity_matrix[user_id] # making the appropriate list
    score = 0 # intializing variables
    id = -1

    # enumerating through items to find location of what we are looking for
    for i,update in enumerate(listScores):
        if i in network[user_id] or i == user_id: # skips itself and the people that are already its friends
            continue
        if update > score:
            id = i
            score = update # refreshes the score if a higher one was found

```

```

    return id

def main():
    print("Facebook Friend Recommendation Program")
    # setting up to run the code
    file = open_file()
    network = read_file(file)
    similarity_matrix = calc_similarity_scores(network)
    n = len(network)

    # runs as long as conditionals are true
    while True:
        try:
            user = input("Enter a number in the range of 0 to {:d}:".format(n-1)) # user input
            if int(user) < 0: # checking if user input is invalid or not
                print("Error: input must be an int between 0 and {:d}:".format(n-1))
                user = input("Enter a number in the range of 0 to {:d}:".format(n-1))
                user = int(user)
            # if not, then we call recommend function
            friend = recommend(int(user), network, similarity_matrix)
            print("The suggested friend for " + str(user) + " is " + str(friend))
            # asking for continuation
            continueQ = input("Do you want to continue (yes/no)?")
            # checking if it's yes or no
            # lower to take in values of no, NO, nO, No
            if continueQ == "yes":
                continue
            if continueQ.lower() == "no":
                break
        except ValueError: # checking if input is outside of range
            print("Error: input must be an int between 0 and {:d}:".format(n-1))
        except IndexError: # checking if input is an integer
            print("Error: input must be an int between 0 to {:d}:".format(n-1))

if __name__ == "__main__":
    main()

```