Computer Programming A, B, E
FAST-NU, Lahore, Spring 2017

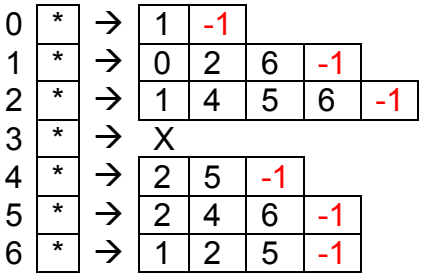
Homework 2

FriendInfo

Due on Tuesday February 14 11:55 P.M. Marked out of 50 points.

In a teeny-tiny social network each user is represented at a given time, simply, by a number. If there are currently n users in the network, they are represented by numbers 0 to n-1. The basic information the network stores is the friendship information: who is friends with whom.

This information is stored in the form of a 2-d array in the heap. Following picture should give you an idea. This structure is entirely on the heap, and stores the information of a network with seven users, numbered 0 to 6. Basically this is an array of pointers, pointing to arrays of integers. In this example, 0 is friends with 1, 1 is friends with 0, 2 and 6 and so on. 3 is not friends with anyone, hence that pointer it set to nullptr. Notice that the size of each array is exactly equal to the number of friends plus one for a -1 in the end.



Your program will read this information from a text file.

For the given example, the file will look as follows (this is self-explanatory: first line contains the total number of users, n, the following lines are comma-separated friend numbers for each of the 0 to n-1 users, where the first number in each line is the user itself).

7

0,1

1,0,2,6

2,1,4,5,6

3

4,2,5

5,5,2,4,6

6,1,2,5

Your program needs to perform the following tasks:

1) When the program starts, it should read FriendInfo from a file friendinfo.txt, allocate the structure, and display the information on the screen. The display should look the same as above. Throughout the program the current, updated info, should be visible.

2) Your program should allow the following queries to the user (print a menu below the friend information):

In the following we assume that the current number of users is n (0 to n-1).

i) Add new user: adds new user, number n, to the network
ii) Make friends: makes two specified users friends with each other
iii) Remove friends: unfriends two specified users
iv) Remove User: removes a specified user, k; unfriends k from all other users, removes k from the array, decrements all numbers above k to keep data consistent.
v) Print Friends: prints all friends of a user
vi) Print Friends of Friends: prints all friends of friends of a user
vii) Print Mutual Friends: prints all mutual friends of two specified users.
viii) Print Likely Friends: for a specified user, x, print 3 users y who are currently not x's friends but have the maximum number of mutual friends with x
ix) Save: save the data onto file (overwrite the previous data)

THE END