Due: - Written Exercises due Friday, May 6th, at 4 pm in Homework box in 2131 Kemper;
- Program due electronically Friday, May 6th, at 11:50 pm.

Written Exercises (25 pts): The written exercises should be typed and each page should have at the top your name and ID#, section, and hw#. Handwritten answers will not be graded.

J&K, 6.6.8, 6.8.6, 6.8.10, 7.1.2, 7.1.8, 7.1.14, 7.2.2, 7.2.6, 7.3.6 d & e, 7.4.8.

Program (75 pts): Handin a makefile and source code files, and have your makefile produce the executable file **`netcensus`** (use **`all`** … at the top). The third line in the source code files must contain the author of the file, ID, and section #. Use the *handin* program for electronic submission, described in the UNIX tutorial. For this homework use:

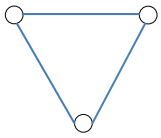
`handin cs30 hw5 Makefile <your file1> <your file2>` …

The date and time your files are created in the cs30 directory will be counted as your submit times. If those times are later than 11:50 pm on the due date your submissions will be considered late.

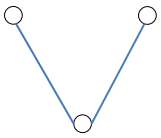
Social Networks

We all are connected to others in many ways, through friendship, collaboration, common interests, etc. Social Networks are abstract structures made of nodes, e.g. people, and the links between them. **Social Network Analysis** is an area concerned with developing quantitative measures for studying social networks. In SNA we study important topological and dynamical features of abstract networks, like hubs, paths, connectivity, percolation, etc. and relate them to real life situations, like forming alliances in congress, or infections spreading in populations, and others. Here, we will take a look at social networks, and compute some measures of relevance to social science.

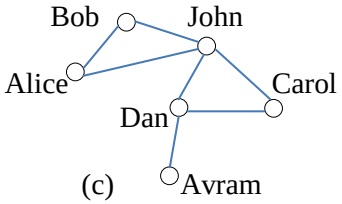
A **triad** in a network is a connected, undirected chain of three nodes, say A, B, and C. Sociologically, we distinguish two types of triads: transitive and intransitive. A **transitive** triad, see Fig. a) below, is when all three nodes are linked pairwise, i.e. A-B, B-C, and C-A exist in the network (e.g. the friend of my friend is my friend too). An **intransitive** triad, Fig. b) below, is one where if A-B and B-C exist in the network, then C-A doesn't (e.g. the friend of my friend is not my friend).



(a)



(b)



(c)

Write a program that asks the user for the name of a file which contains a social network. Assume the nodes in the network are strings, of length at most 30 characters, and no white spaces. The first line of the input file specifies **`n`**, the number of nodes <= 200, and the next **`m`** lines specify **`m`** links (edges). The program should then read in the network from this file and write to a file **`census.out`** the network file name in the first line, and the following network census in the next 5 lines: (1) the number of nodes, (2) number of links, (3) most connected node, (4) the number of transitive triads, and (5) the number of intransitive triads in the network. For the network in Fig. c) above, the respective counts are:
6, 7, John, 2, 6.

Hint: Use an adjacency matrix **`int A[200][200]`** to store the network, where **`A[i][j]=1`**, if there is a link between nodes **`i`** and **`j`**, and **`A[i][j]=0`** otherwise. Use an array of strings **`char names[200][30]`** to store the names.

Make sure the output of your program matches exactly the output below of my executable located at **`/home/cs30/public/hw5/netcensus`** on the csif machines. In the same directory you will find a file **`net.txt`** which contains an example network.

```
[cs30@pc50 hw5]$ less net.txt

6
Bob Alice
Bob John
Alice John
Dan Avram
John Karol
Karol Dan
John Dan

[cs30@pc50 hw5]$ ./netcensus
Name of network file: net.txt

[cs30@pc50 hw5]$ less census.out

net.txt
Number of nodes: 6
Number of links: 7
Most connected node: John
Transitive triads: 2
Intransitive triads: 6
```