Programming Assignment 4
Computing properties of PPI networks (100 points)

A Protein-Protein Interaction (PPI) network is an undirected graph where nodes represent proteins and an edge between two nodes represents interacting proteins (either physically or functionally).
You will write a program that computes local properties in a PPI network and in random graphs.
A detailed description of such properties, already presented in class, follows:

Degree distributions
Recall that the degree of a node in a network with *n* nodes is the number of edges incident on (i.e., connected to) that node. We define p_k to be the number of nodes in the network that have degree k. The degree distribution of a network is a histogram or table of the frequencies p_k for all k.

Cliques of size 3.
A clique of size 3 in an undirected graph is a set of 3 nodes that are pairwise adjacent. For instance, the nodes *a, b*, and *c* form a clique if the following 3 edges are present in the graph: ab, bc, and ac. In other words *a, b*, and *c* form a triangle.

Neighborhoods of a node.
The neighborhood of node *a* at distance 1 is the set T1 of all nodes adjacent to *a*. The neighborhood of node *a* at distance 2 is the set T2 of all nodes that are adjacent to the nodes of T1. T2 is a set and as such does not contain repeated nodes.

Random graphs

In a random Erdos and Renyi graph each edge is present with equal probability p. If you choose $p = 1/2$ then each edge occurs with probability ½, and you can decide about the presence/absence of an edge between two nodes by tossing an unbiased coin. You simulate that by using a pseudo-random number generator. In python this is encoded by the functions numpy.random, random.

Your program

Takes in input a PPI network and:

a) Computes and prints the degree distribution of the network
b) Computes and prints all cliques of size 3
c) Computes the neighborhood T2 of all nodes of the network. It prints the node whose neighborhood has the largest size among all nodes of the graph.

Generates a random graph in which each edge occurs with probability 0.5. The graph has the same number of nodes of the input PPI interaction network. As *ids* of the nodes you can use the same labels of the input graph or integers. It does not matter.

Repeats for the random graph tasks a), b) and c).

Generates a random graph in which each edge is present with probability $p = 0.1$ and another graph with probability $p = 0.8$.

Repeats for the random graph tasks a), b) and c).

INPUT DATA
In this assignment you analyze the Protein-Protein Interaction (PPI) graph of the herpes Kaposi virus. The file kshv.sif contains such a graph in sif format. Each line of the file represents an edge and looks like the one below:

```
kshv_ORF53 1.0   kshv_ORF45
```

In the above example, the edge connects the two proteins `kshv_ORF53 and kshv_ORF45.` The intermediate value 1.0 on the same line is the weight of the edge. In your file all weights are 1.

OUTPUT
Your program prints the following results for both the PPI network and for the random graphs:

1) a table (histogram) of the degree frequencies, where each line consists of a value of k and the corresponding p_k, for instance:
   k=0 $p_0 = 4$ (if in the network there are 4 isolated nodes)
   k=1 $p_1 = 7$ (if in the network there are 7 nodes of degree 1)
   and so on.
2) all cliques of size 3 as follows:
   a,b,c; x,y,z; ….
3) The neighborhood(s) of largest size over all nodes of the graph.
   For instance is the size T2 of the node a is 15 and this is the maximum of the sizes of T2 over all nodes of the graph, then you print something like:

   Node a has the largest neighborhood T2 and the size of T2 is 15.

Print the same results for the random graph with $p = 0.5, p = 0.1$ *and* $p = 0.8$.

IMPLEMENTATION
Use an *adjacency matrix* representation of the graph, i.e. a $n \times n$ *matrix* ADJ with ADJ[u,v]=1 if there is an edge between nodes u and v, 0 otherwise. Write a function to **Create-Adjacency-Matrix** that process the .sif representation of the graph and generates ADJ.

SUBMISSION
Electronically submit the following:

a) a pdf file of the cytoscape drawing of the random graph you have generated with p=1/2.
b) a pdf file of the cytoscape drawing of the random graph you have generated with p=0.1 and p=0.8
c) an iPython notebook file (*.ipynb) with your implementation.