

```
In [1]: # Import packages
import numpy as np
import scipy as sp
import networkx as nx
import pandas as pd
import datetime
import pickle

# To import Matlab matrices
import scipy.io

# Plotting
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
```

```
Out[2]: '2020-12-30 10:25:55'
```

Fields are:

- index
- problem_village
- graph
- key
- chosen
- guess_rank
- true_rank
- dontknow
- consumption

```
In [3]: # Extract directly from Matlab files
def IndonesiaData():
    """
    Over 600 villages in Indonesia from Alatas Banerjee Chandrasekhar Hanna and Olken 2016 AER

    Return a dictionary of networks.
    """

    result = {}
    dataset_matlab = scipy.io.loadmat('../data/119802-V1/Network Data/Network Data/finalData')
    for i in range(0,633):
        village = str(dataset_matlab['finalData'][['index'][0][0][0][i]])
        graph = nx.from_numpy_matrix(dataset_matlab['finalData'][['graph'][i][0]])
        hhid = [item for sublist in dataset_matlab['finalData'][['key'][i][0]].tolist() for item in sublist]
        mapping = dict(zip(graph, hhid))
        graph = nx.relabel_nodes(graph, mapping)
        num_chosen = len(dataset_matlab['finalData'][['chosen'][i][0]])
        chosen = [dataset_matlab['finalData'][['chosen'][i][0][x][0]] for x in range(num_chosen)]
        guess_rank = {}
        true_rank = {}
        for k in range(num_chosen):
            r_guess = dataset_matlab['finalData'][['guess_rank'][i][0][k]].tolist()
            r_true = dataset_matlab['finalData'][['true_rank'][i][0][k]].tolist()
            guess_rank[chosen[k]] = dict(zip(chosen, r_guess))
            true_rank[chosen[k]] = dict(zip(chosen, r_true))
        consum_list = dataset_matlab['finalData'][['consumption'][i][0]].tolist()
        consum_dict = {}
        for y in consum_list:
            consum_dict[int(y[0])] = y[1]
        result["hamlet_"+str(village)] = {"graph":graph,
                                         "chosen":chosen,
                                         "guess_rank":guess_rank,
                                         "true_rank":true_rank,
                                         "consumption":consum_dict}
    return result
```

```
In [4]: data_dict = IndonesiaData()
```

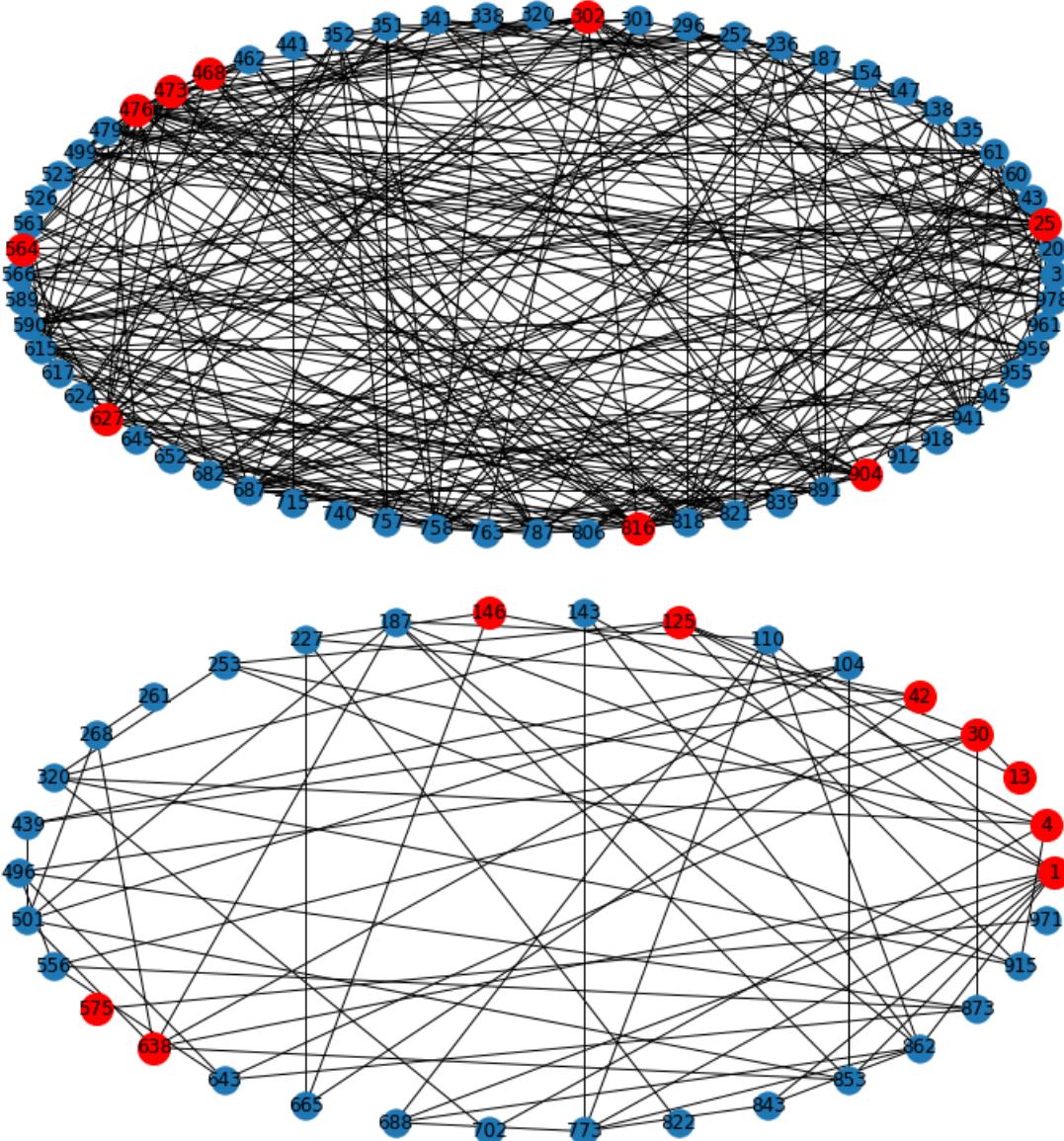
Plot some of the networks

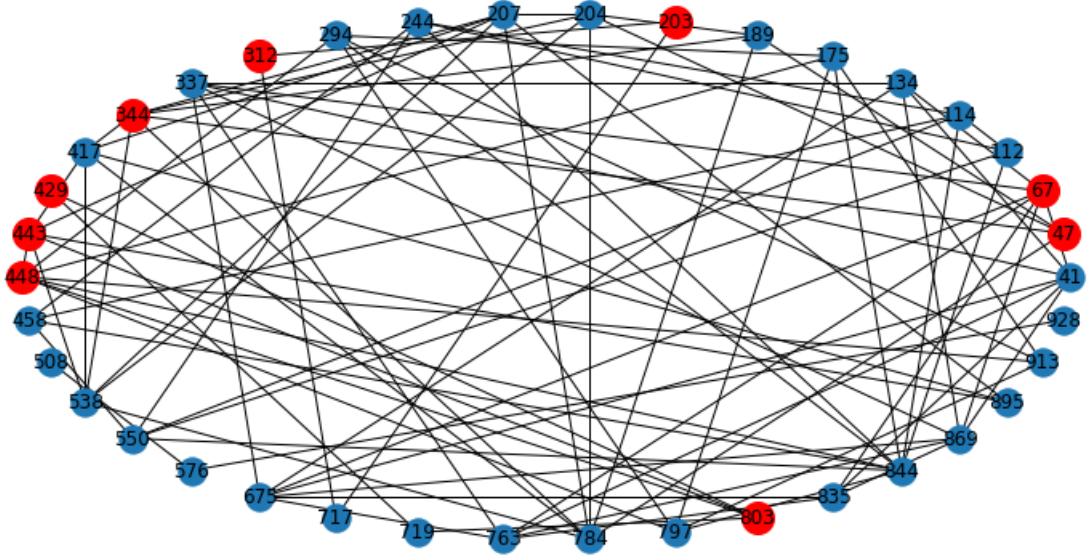
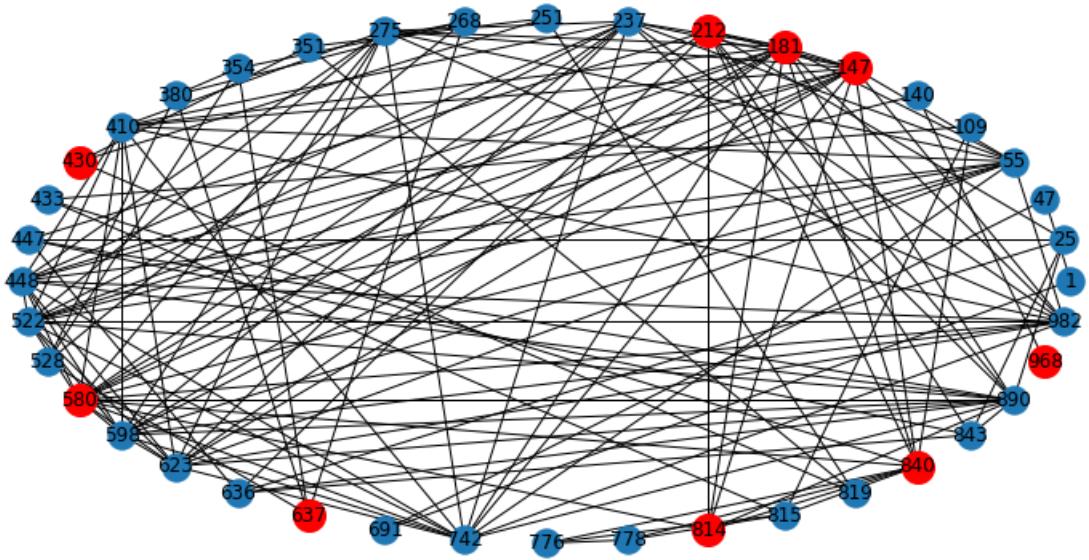
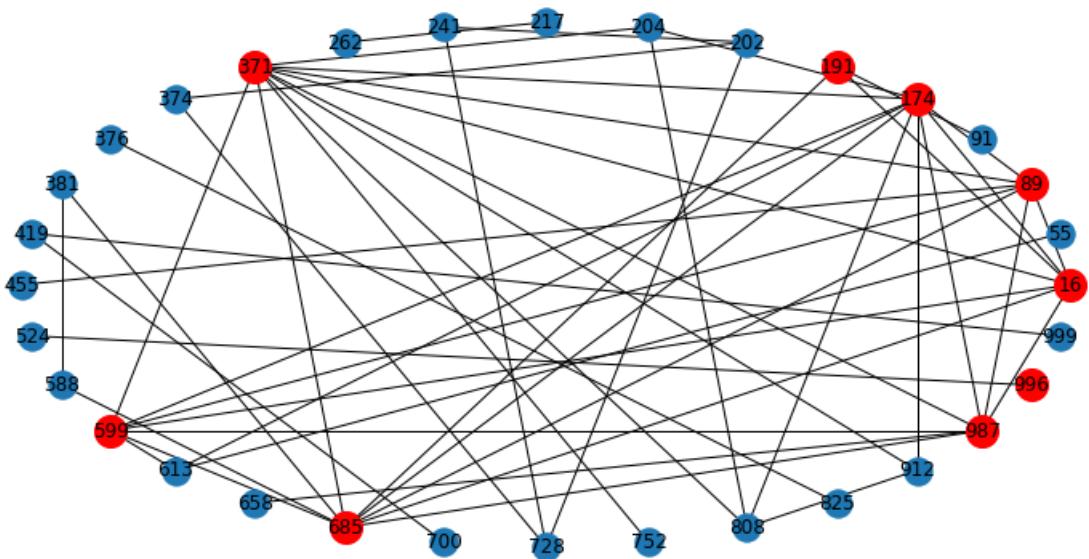
```
In [5]: x = 0
```

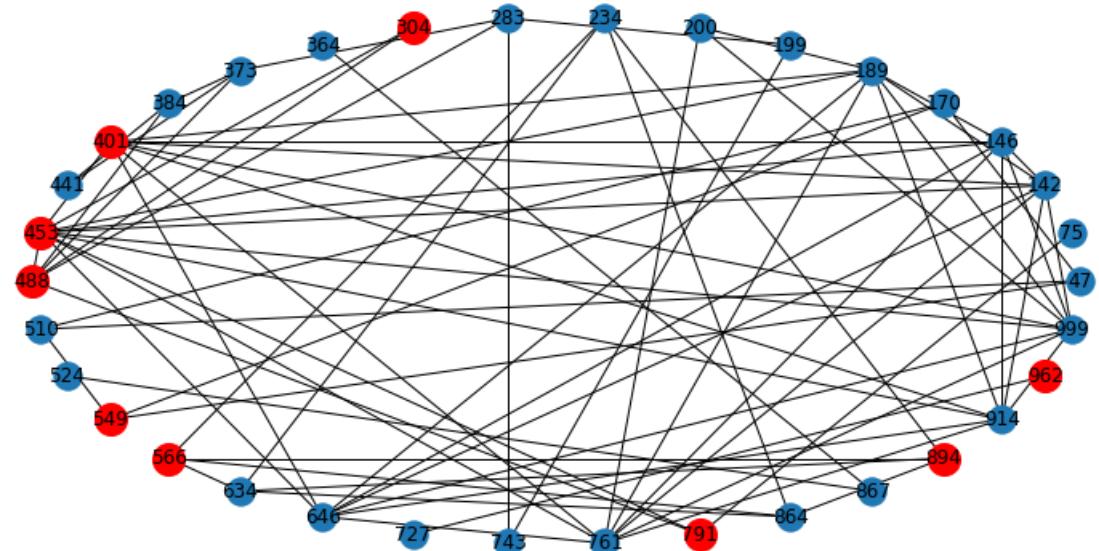
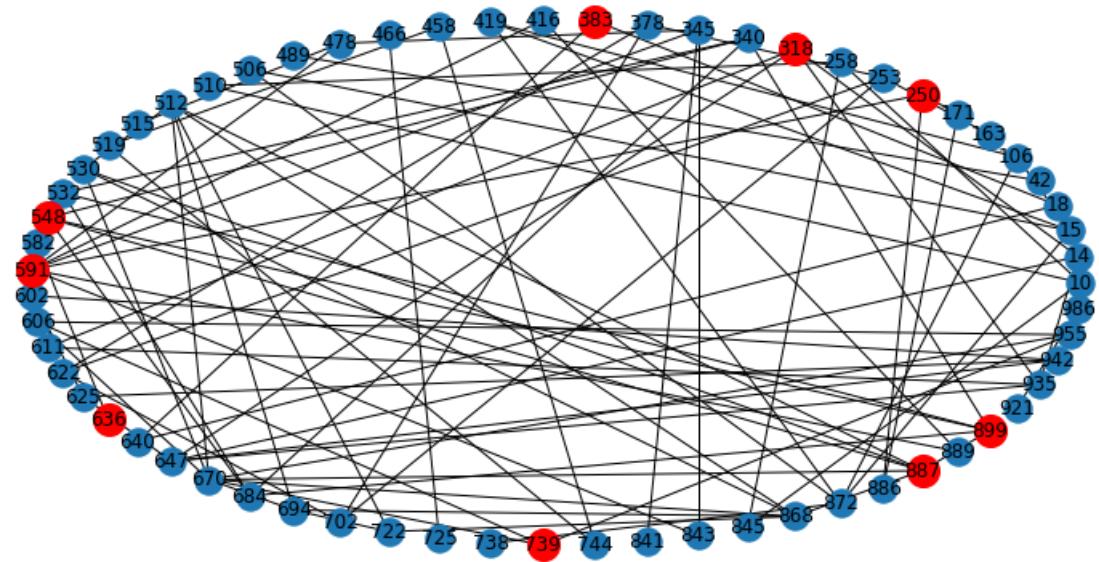
```

for key in data_dict.keys():
    if x <= 50:
        try:
            graph = data_dict[key]["graph"]
            graph.remove_nodes_from(list(nx.isolates(graph)))
            chosen = data_dict[key]["chosen"]
            pos_g = nx.circular_layout(graph)
            plt.figure(figsize=(10,5))
            nx.draw(graph, with_labels=True, pos=pos_g)
            nx.draw_networkx_nodes(graph, pos=pos_g,
                                   nodelist=chosen, node_color='r', node_size=400)
            plt.show()
        except:
            print(key)
            continue
    x+= 1

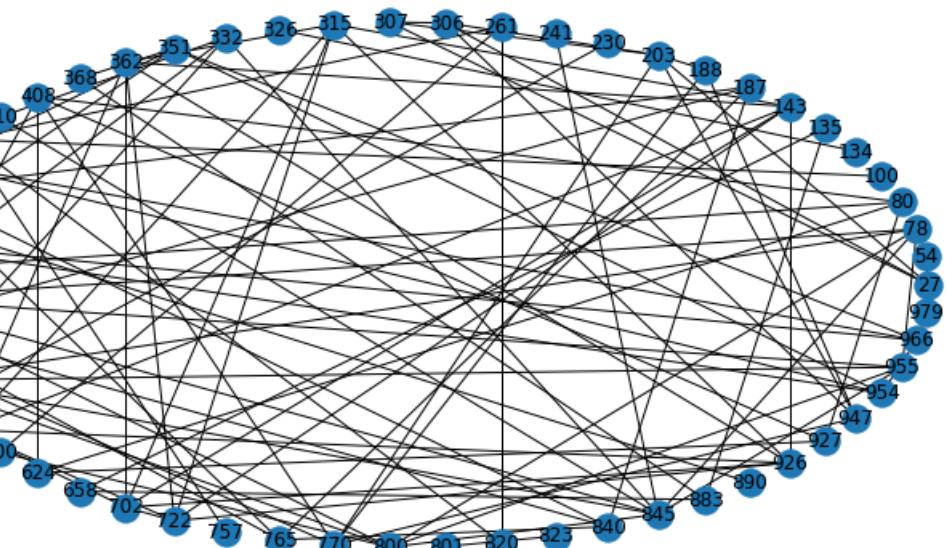
```

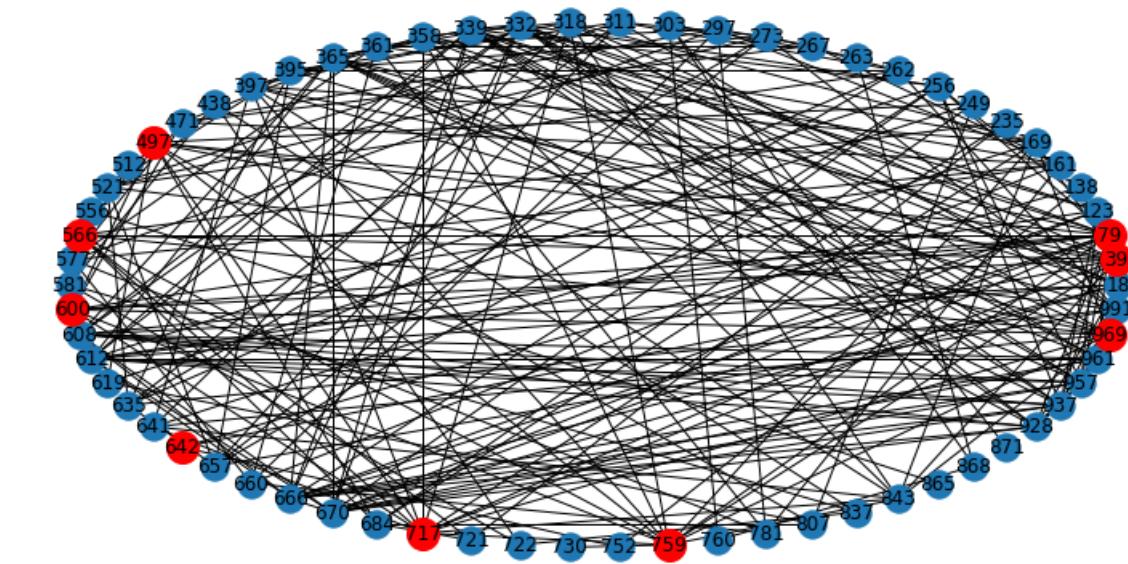




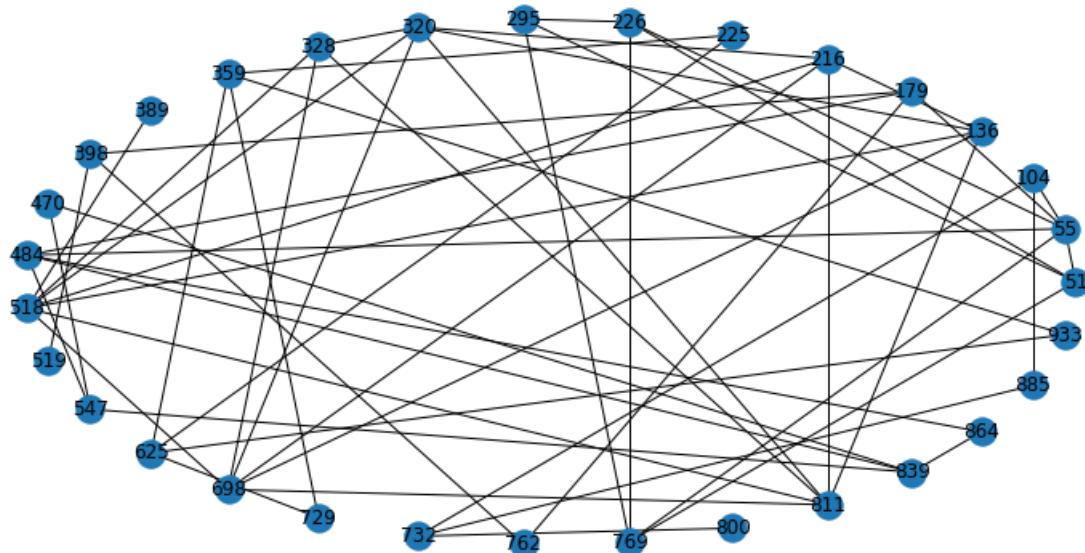
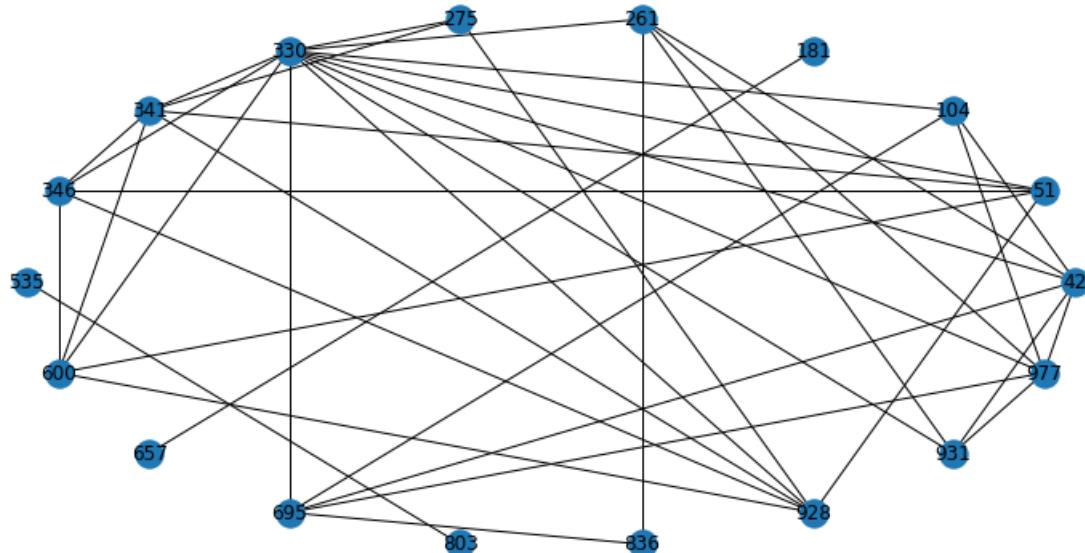


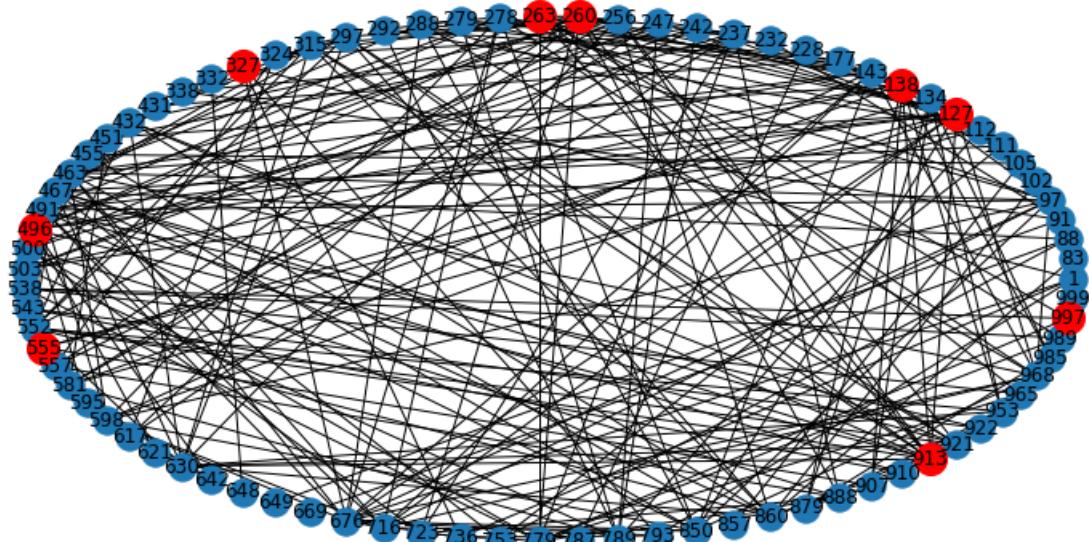
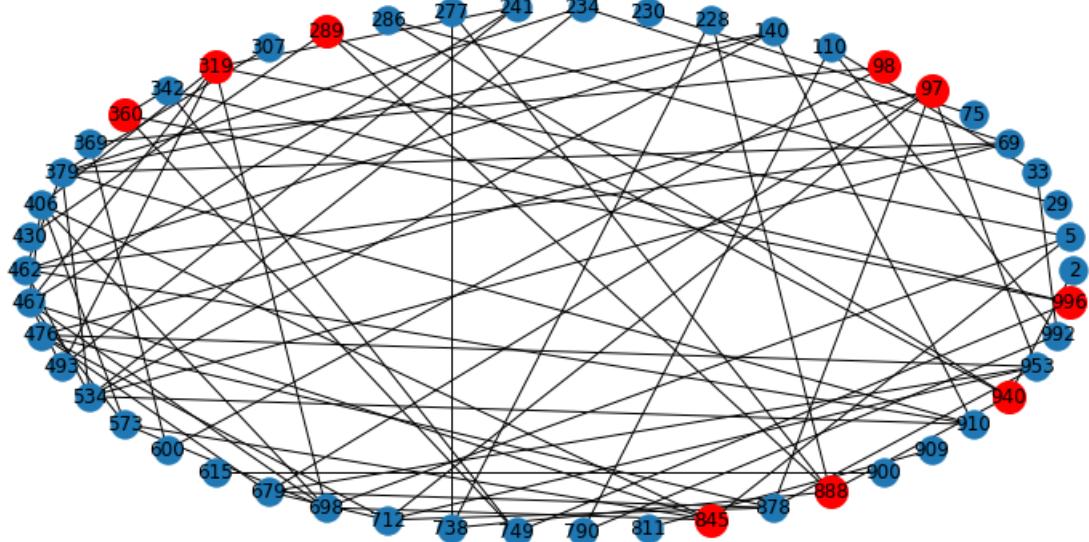
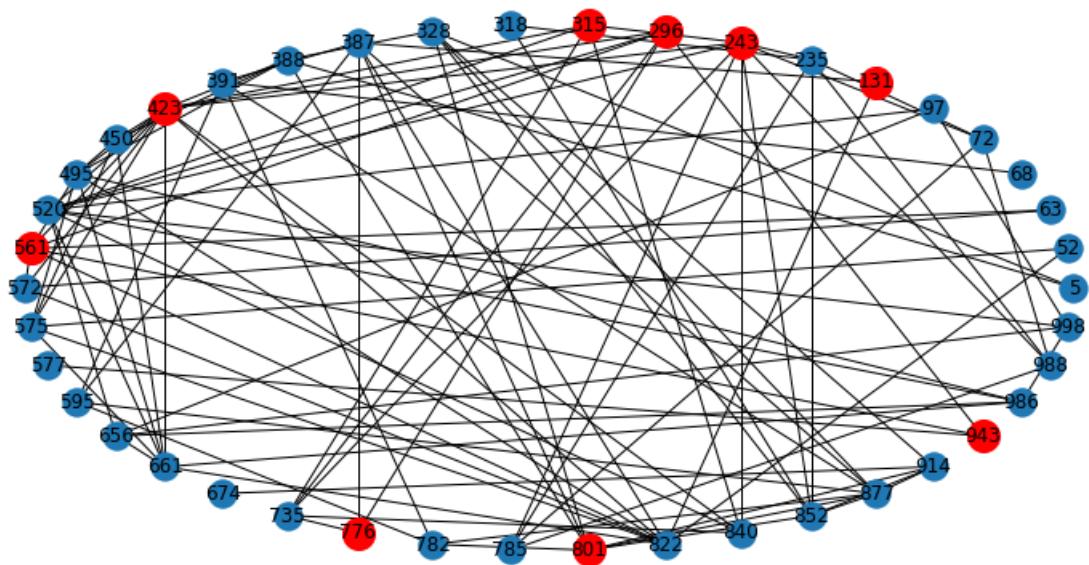
hamlet_8

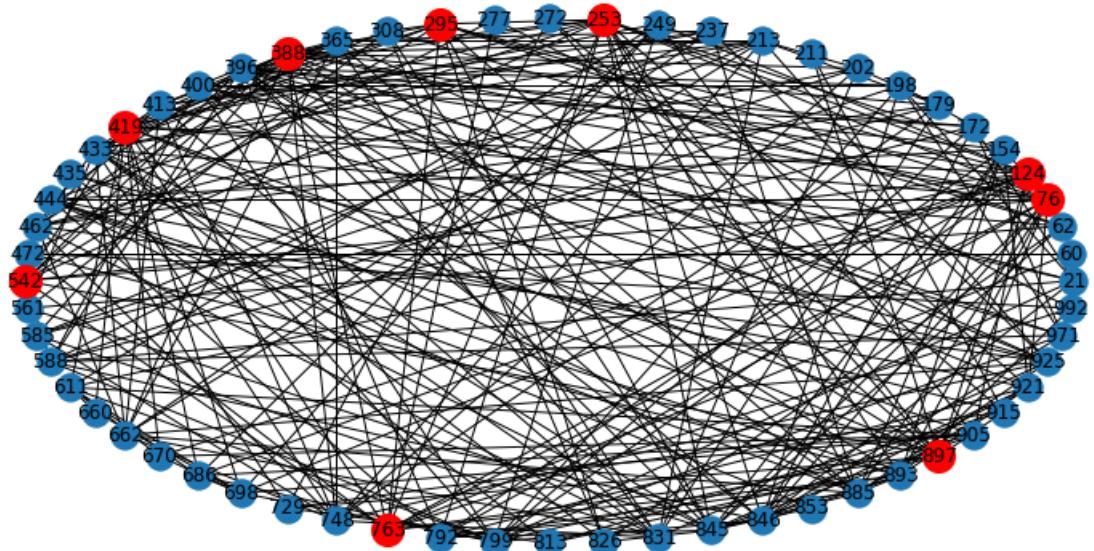
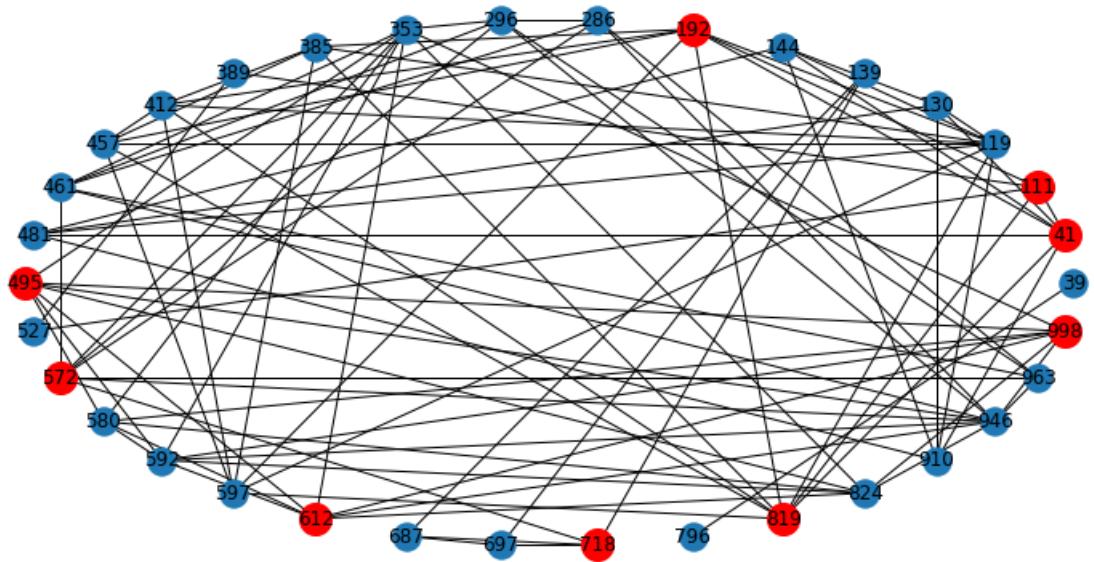




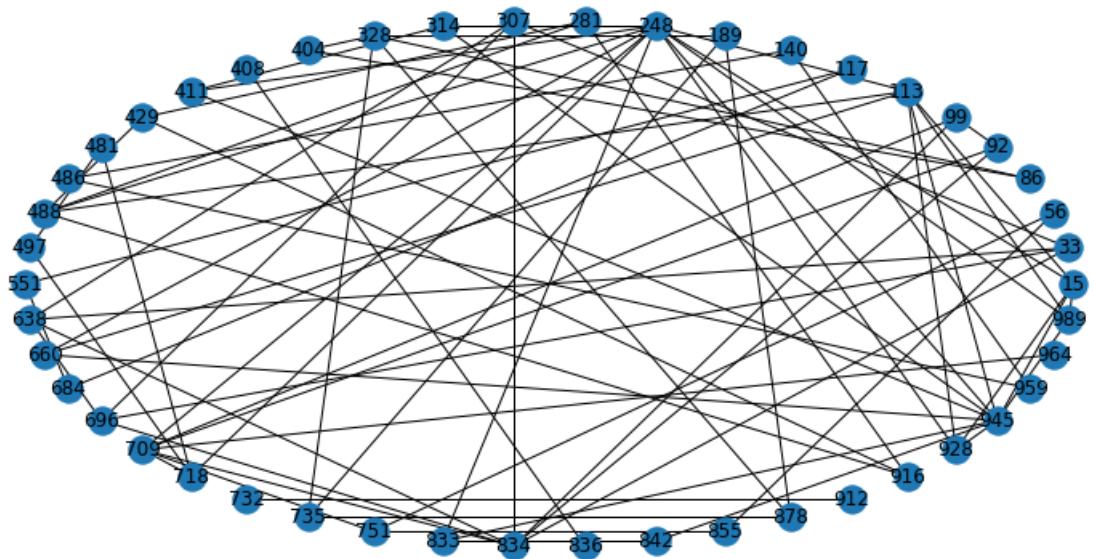
hamlet_10
hamlet_11

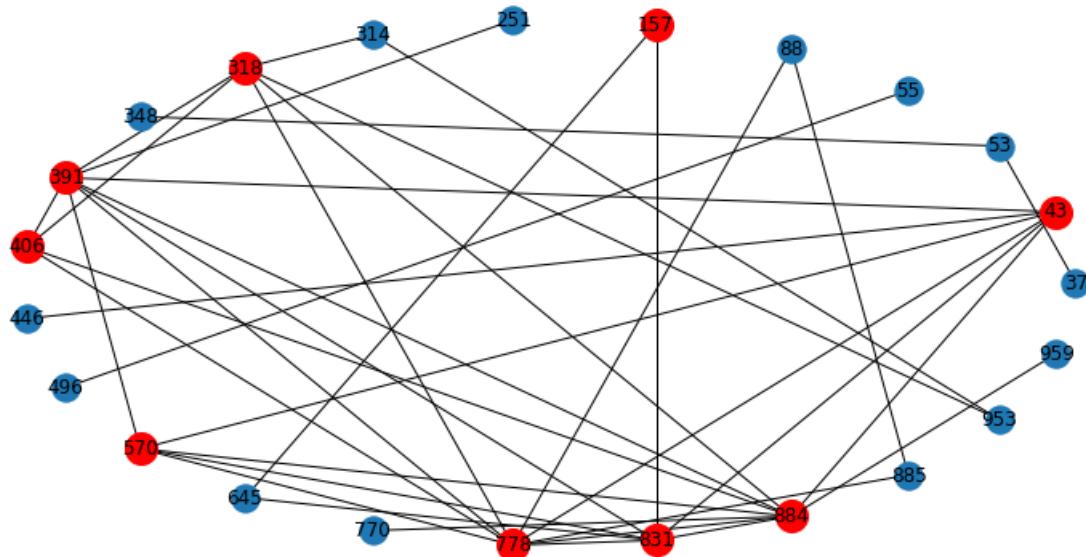
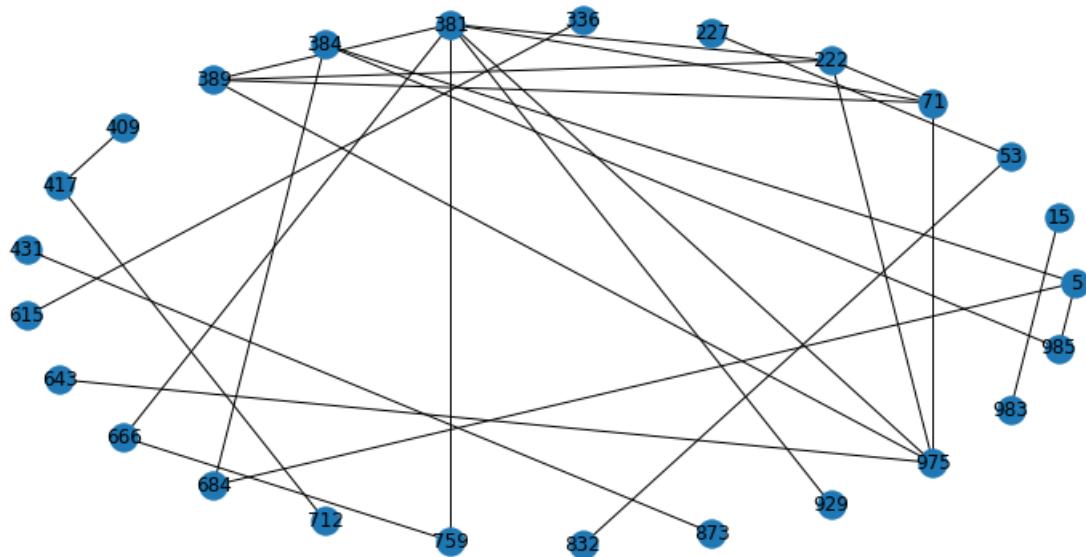




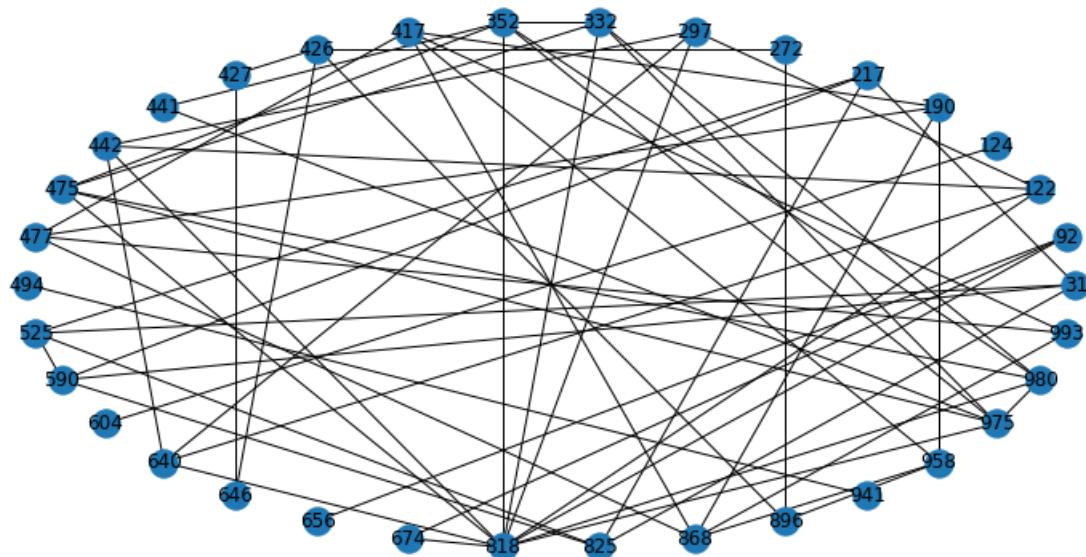


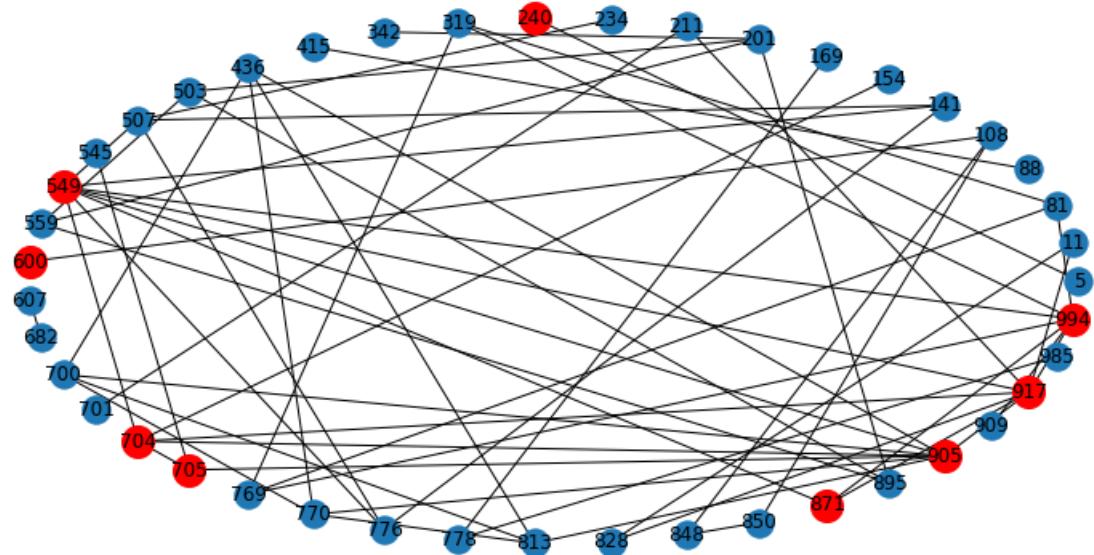
hamlet_17
hamlet_18



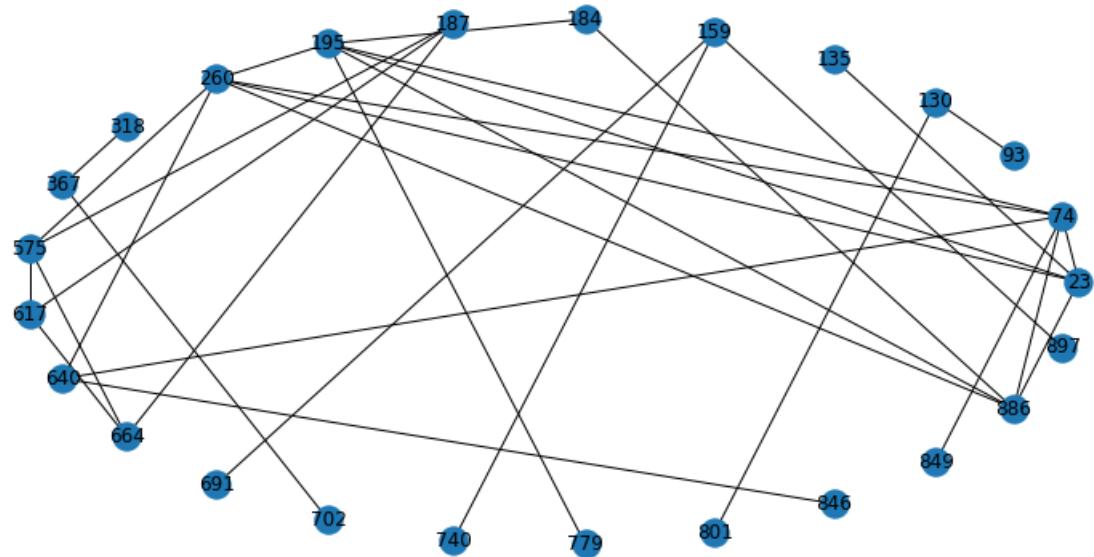
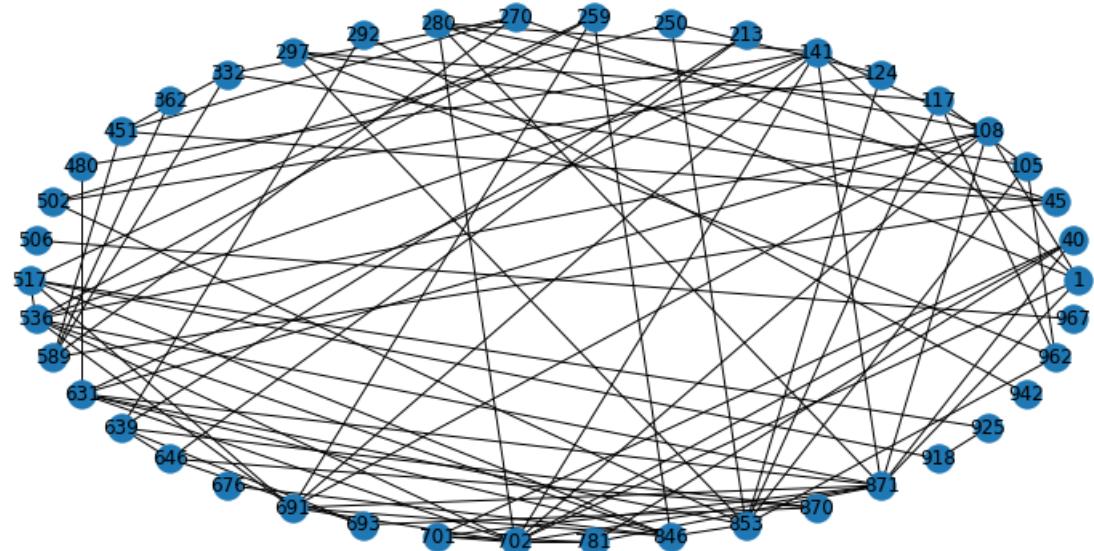


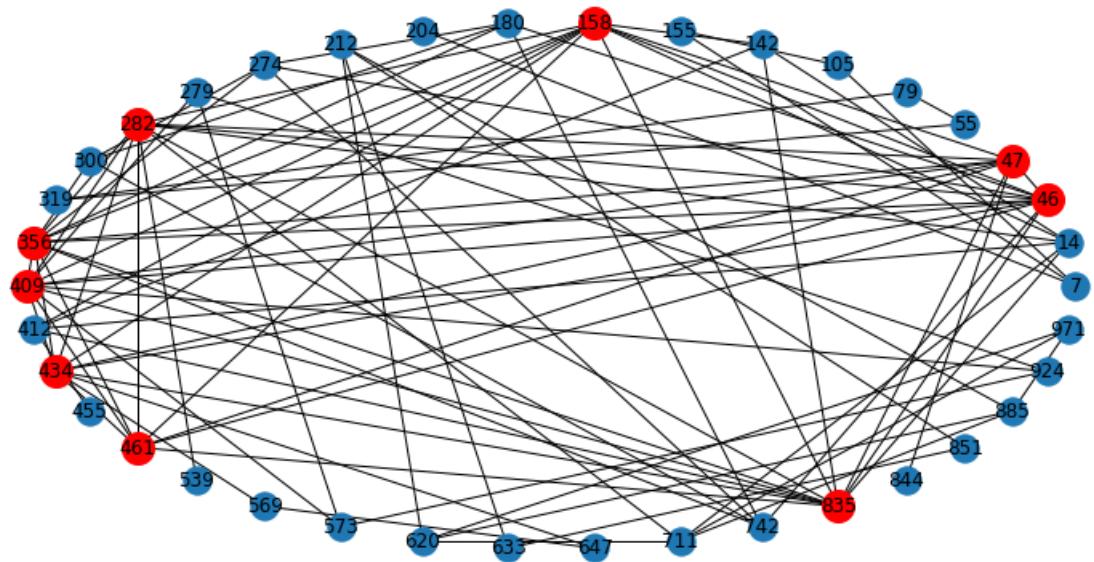
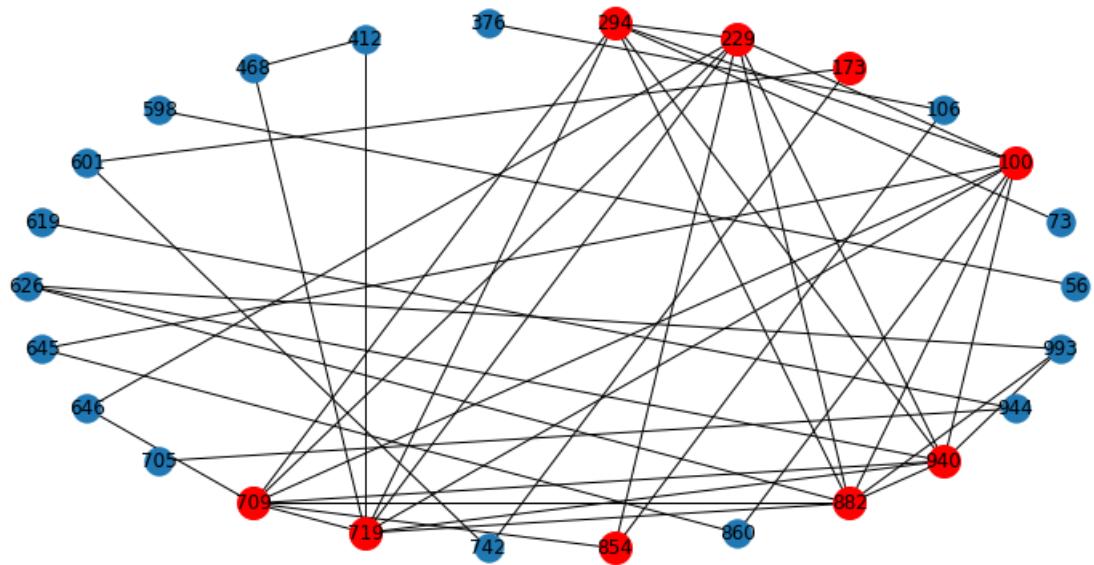
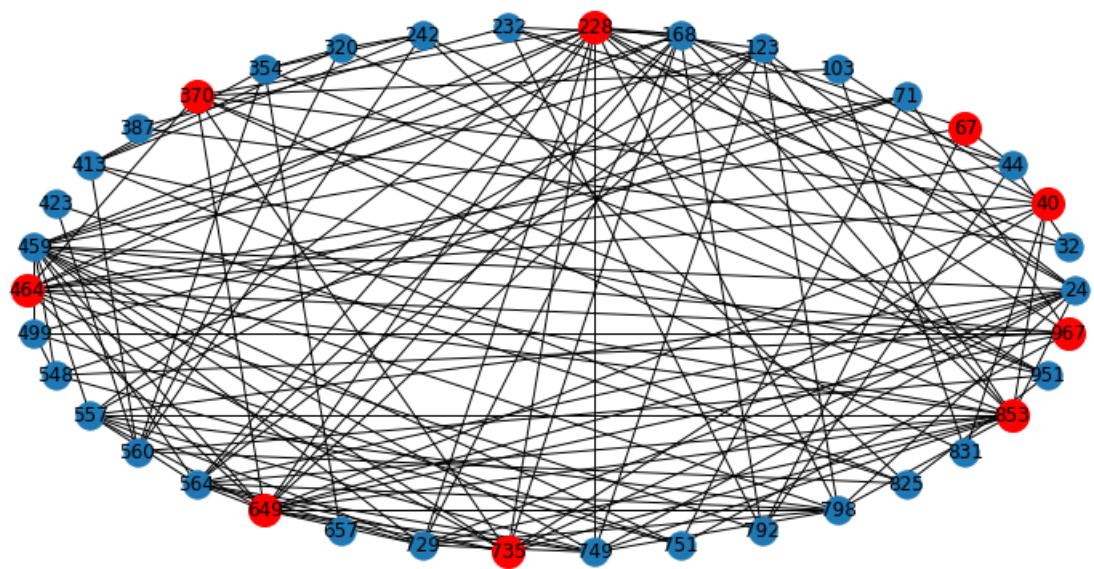
hamlet_20

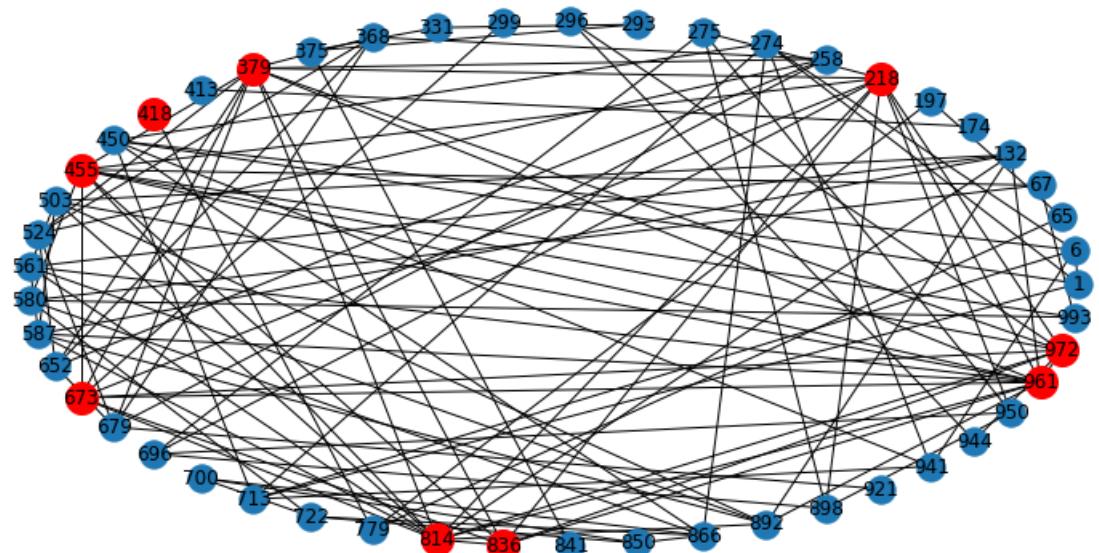
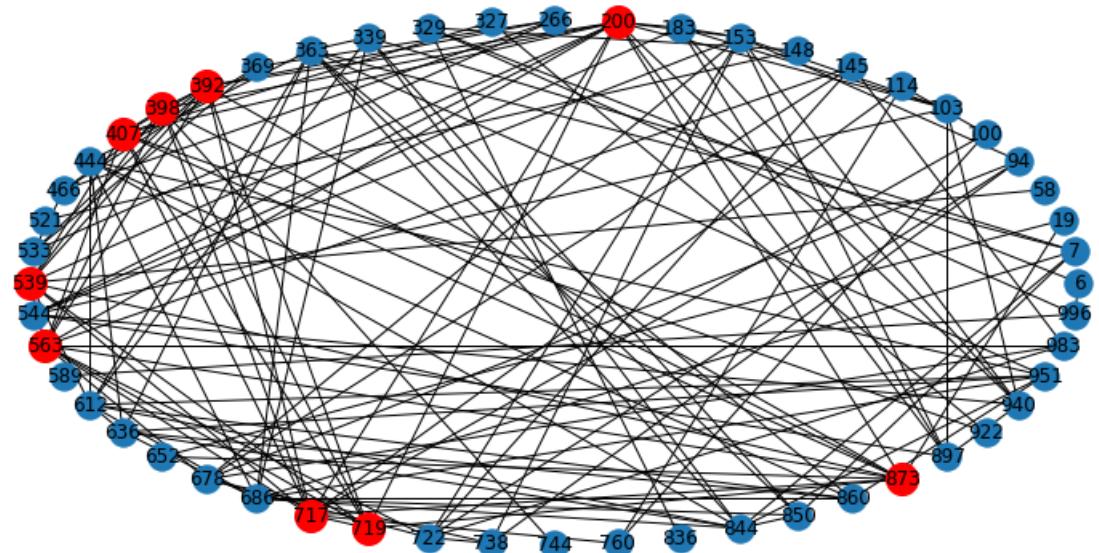
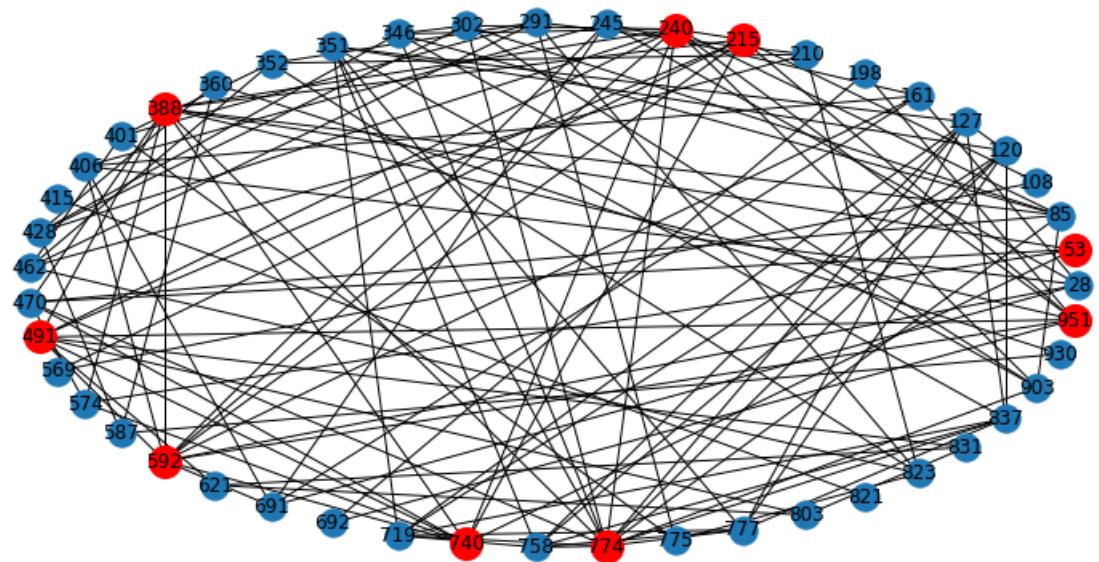


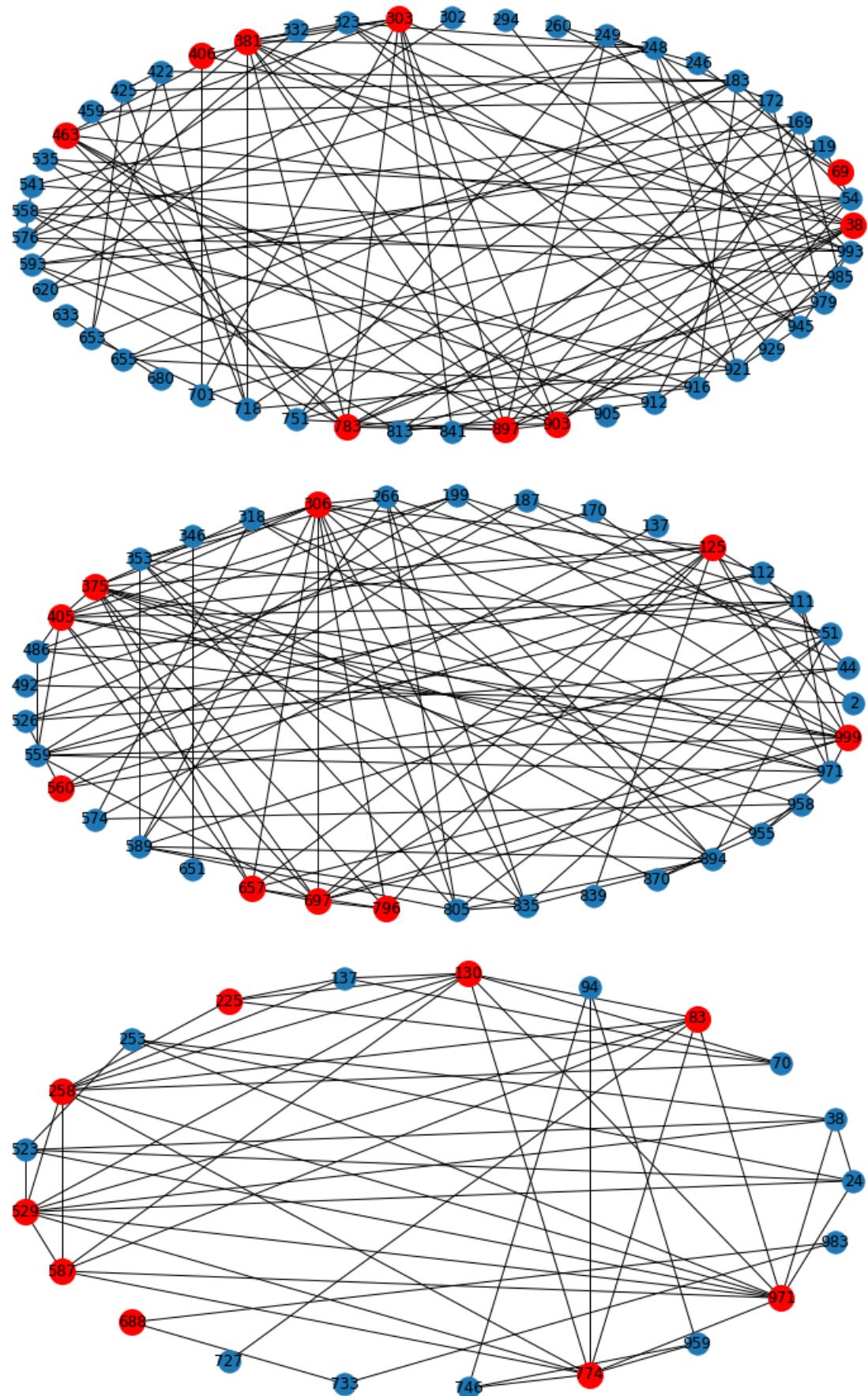


hamlet_22
hamlet_23

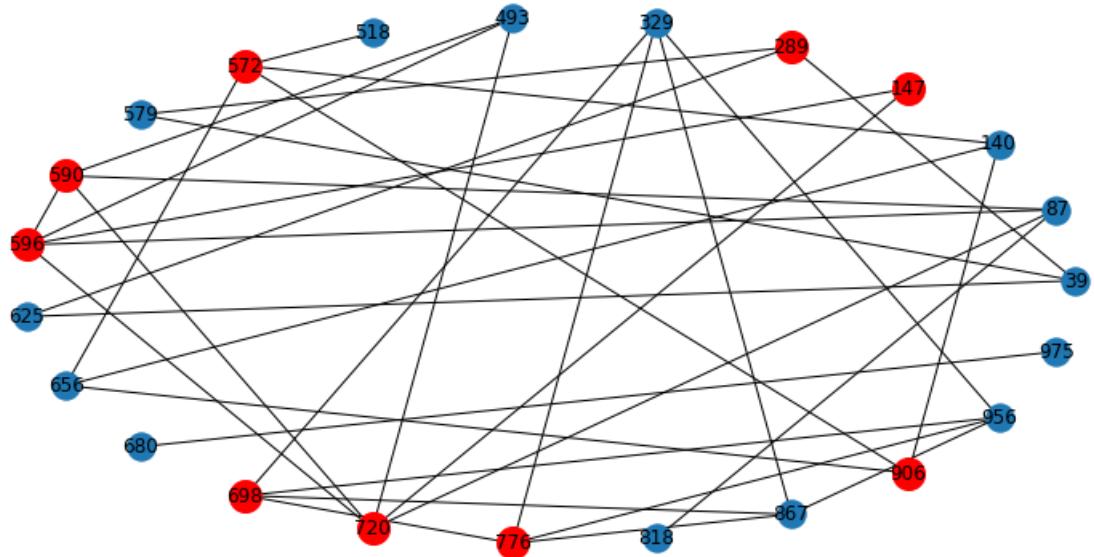
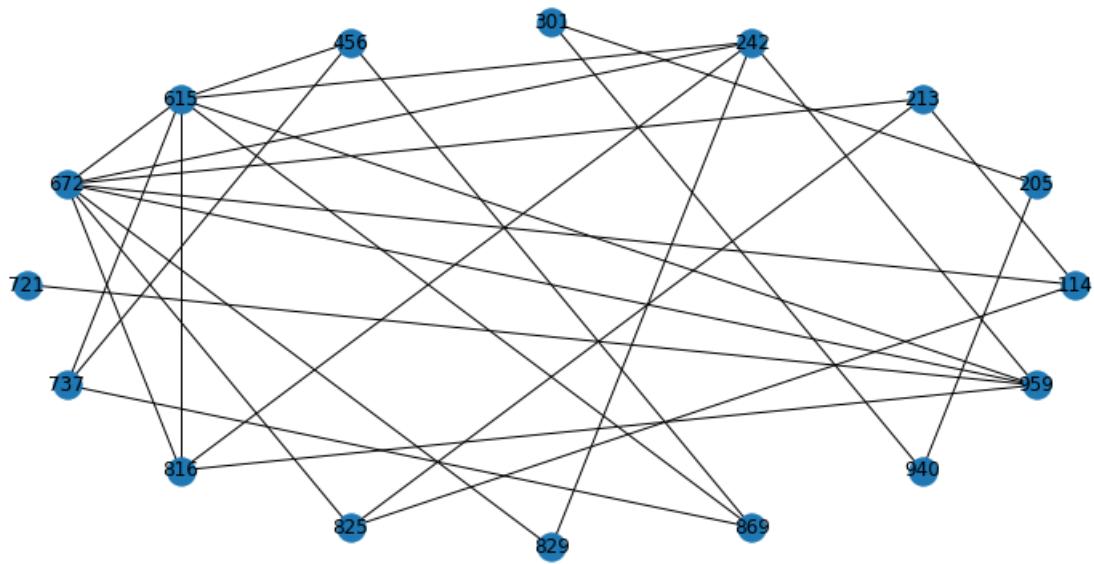




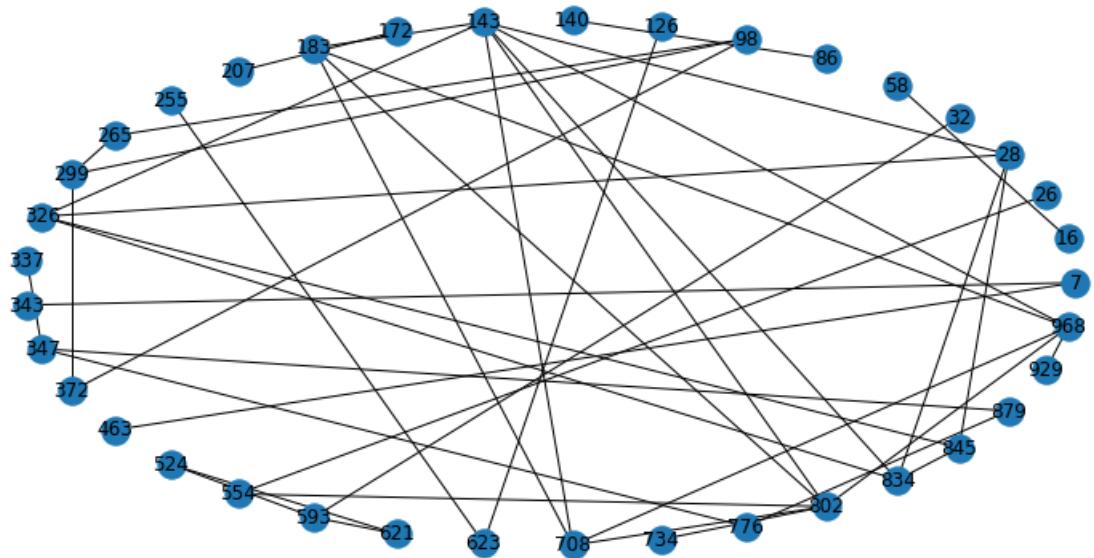


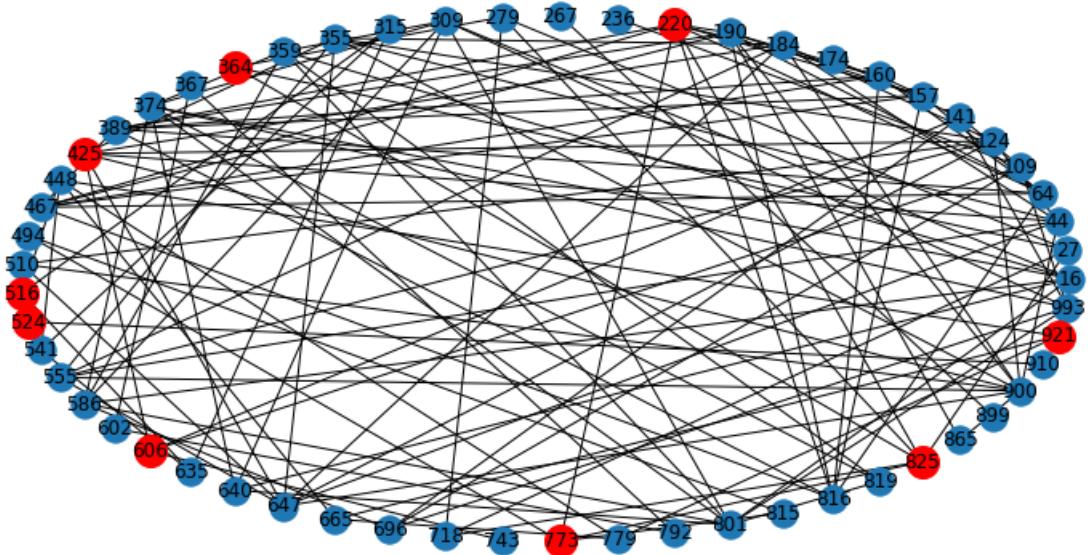
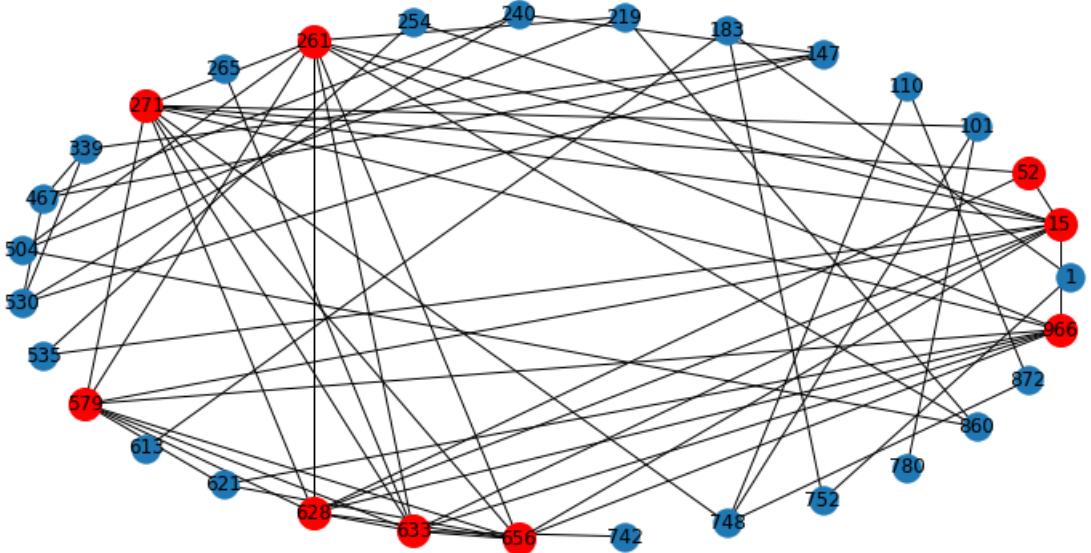
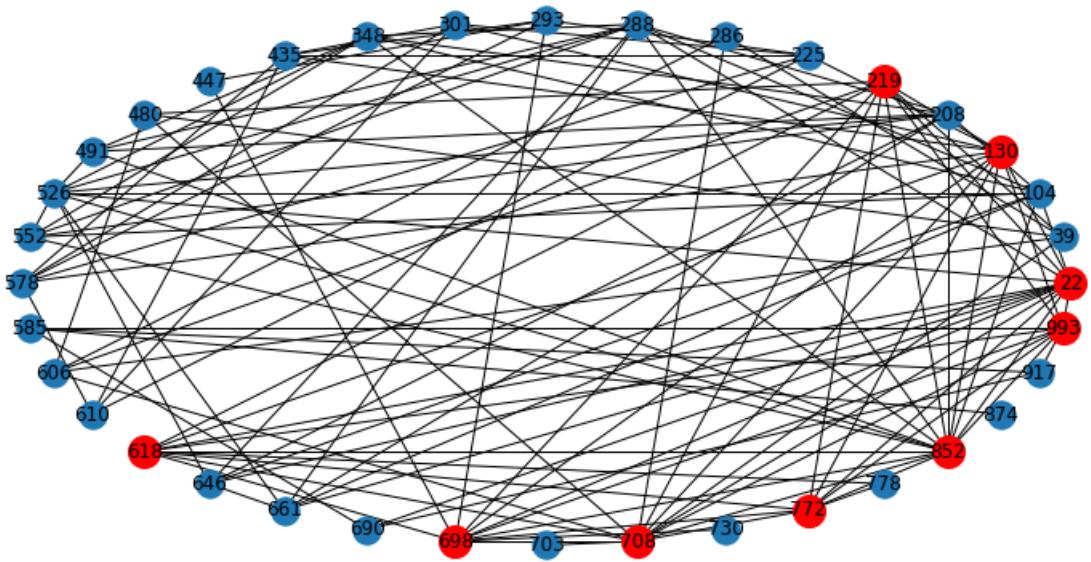


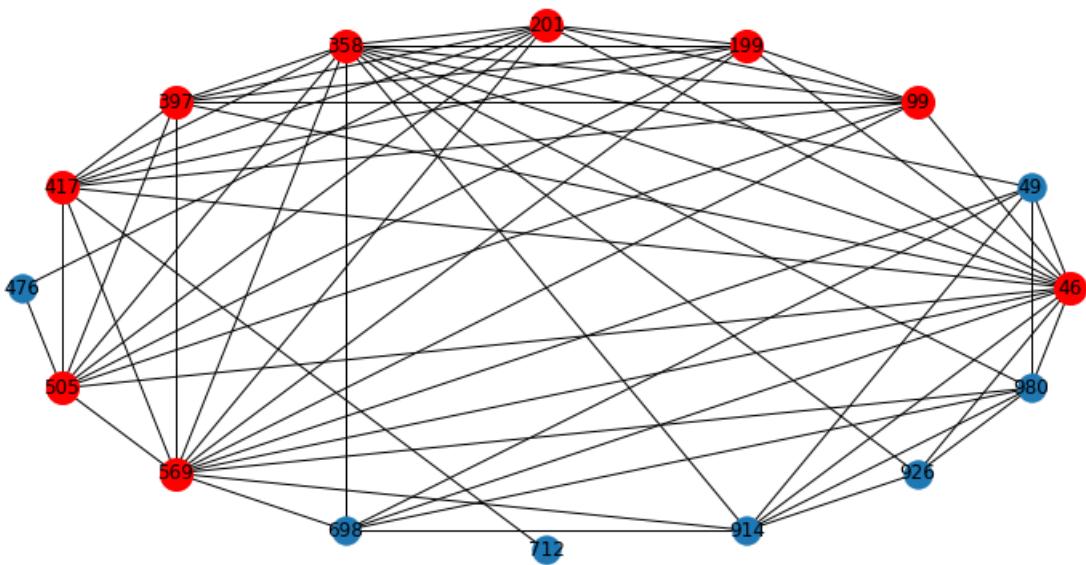
hamlet_33



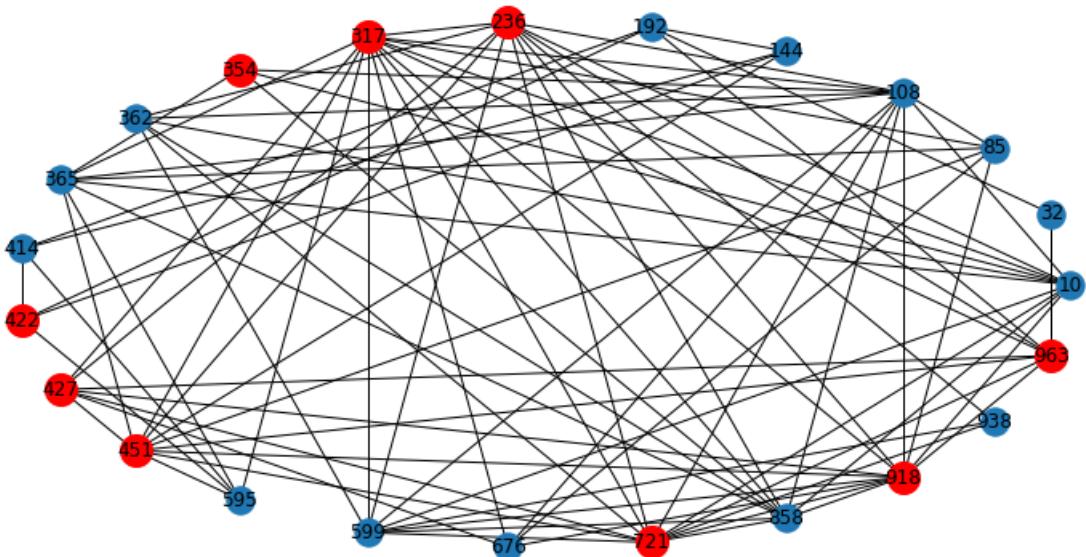
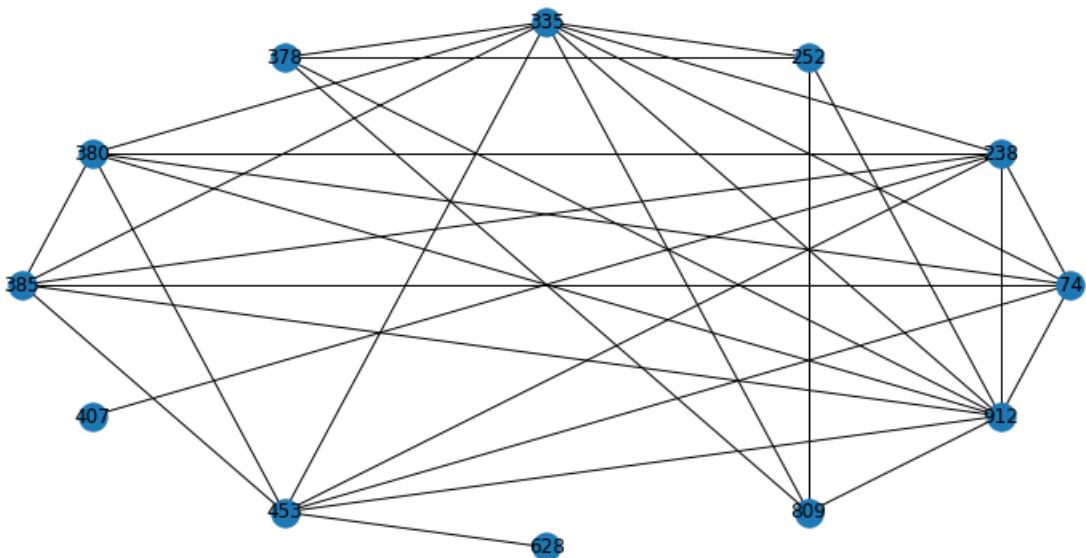
hamlet_35

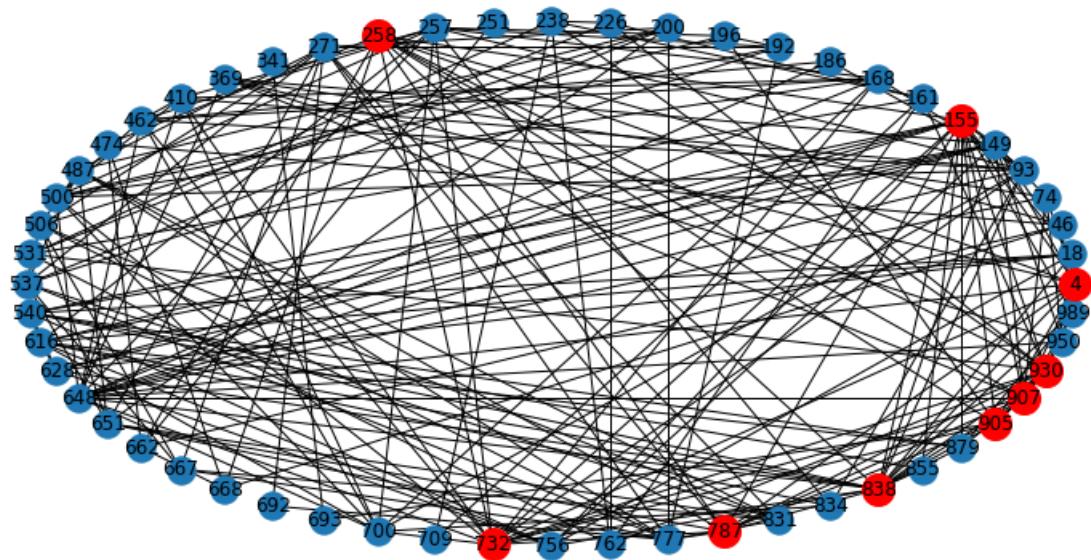
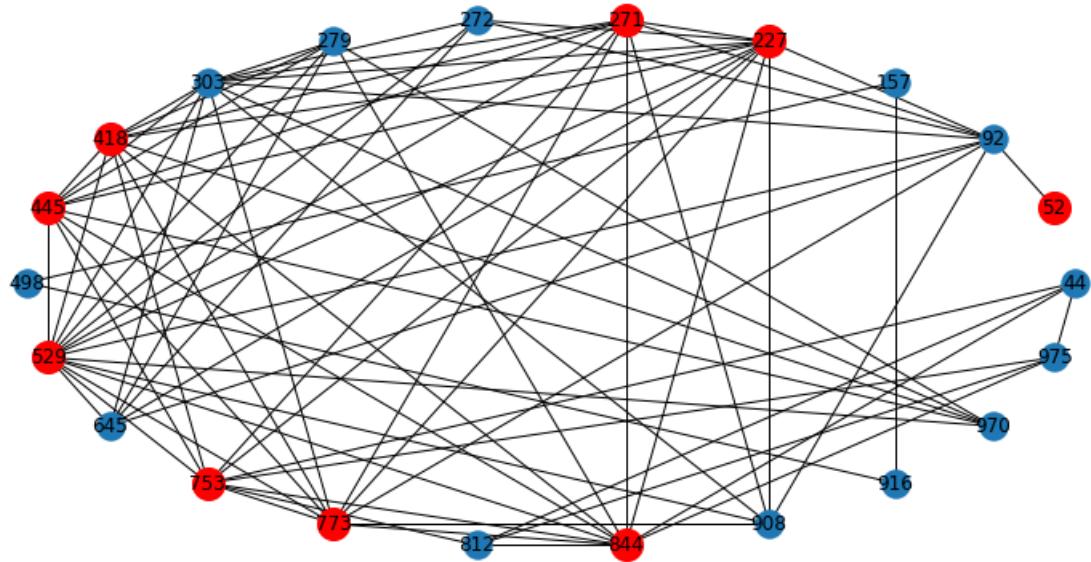
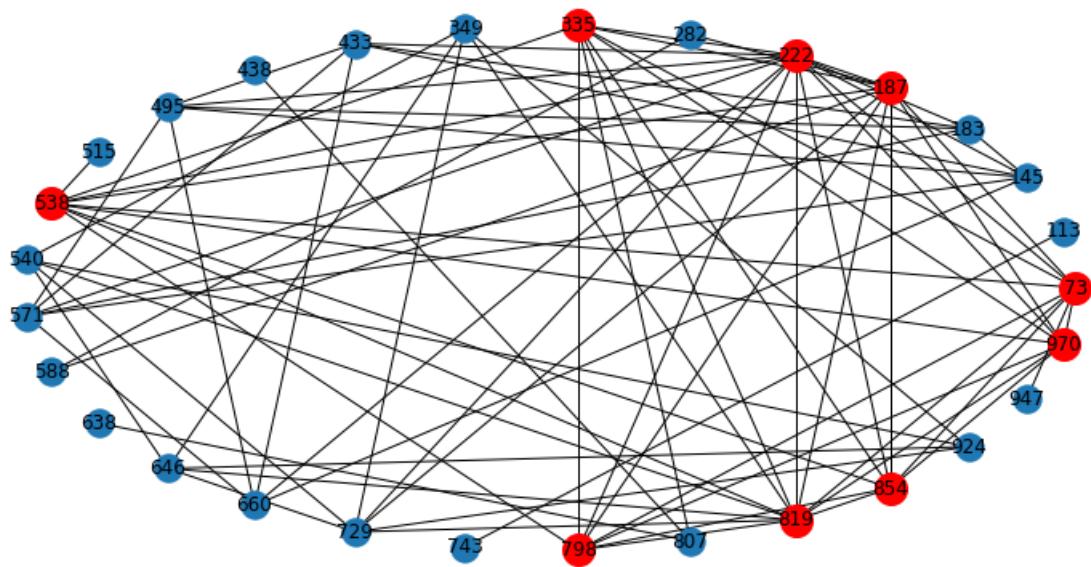


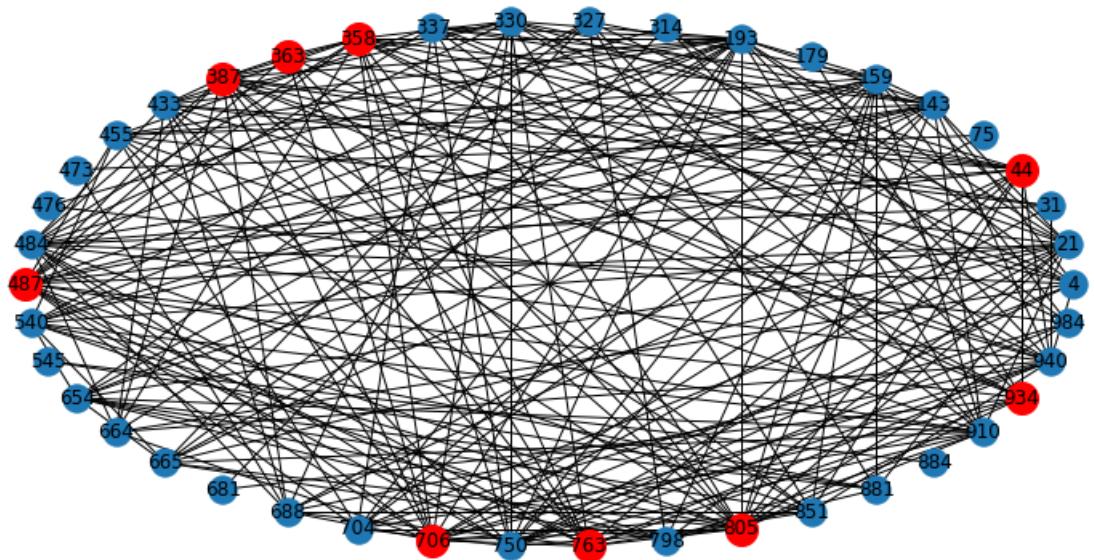
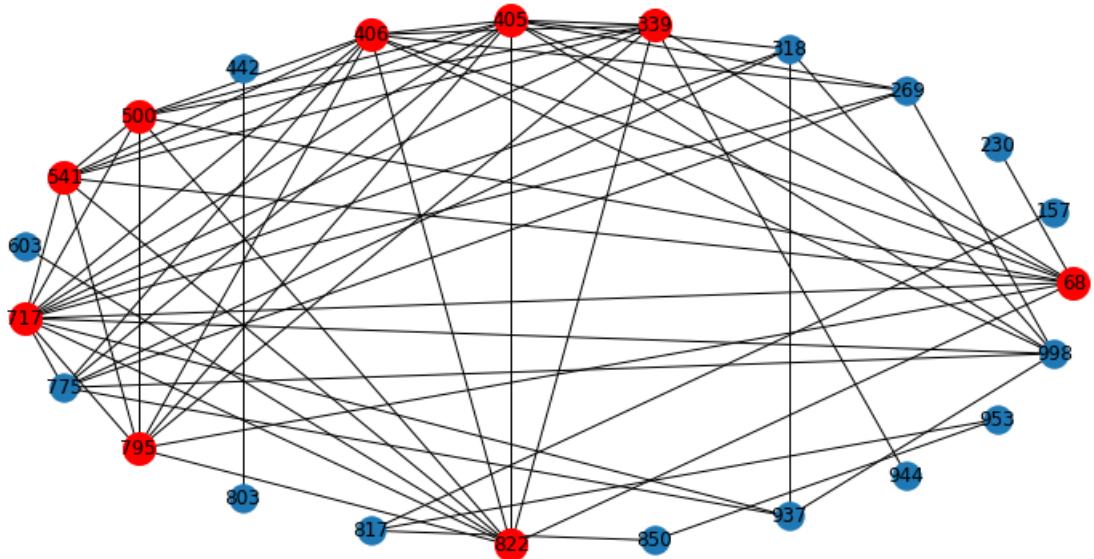
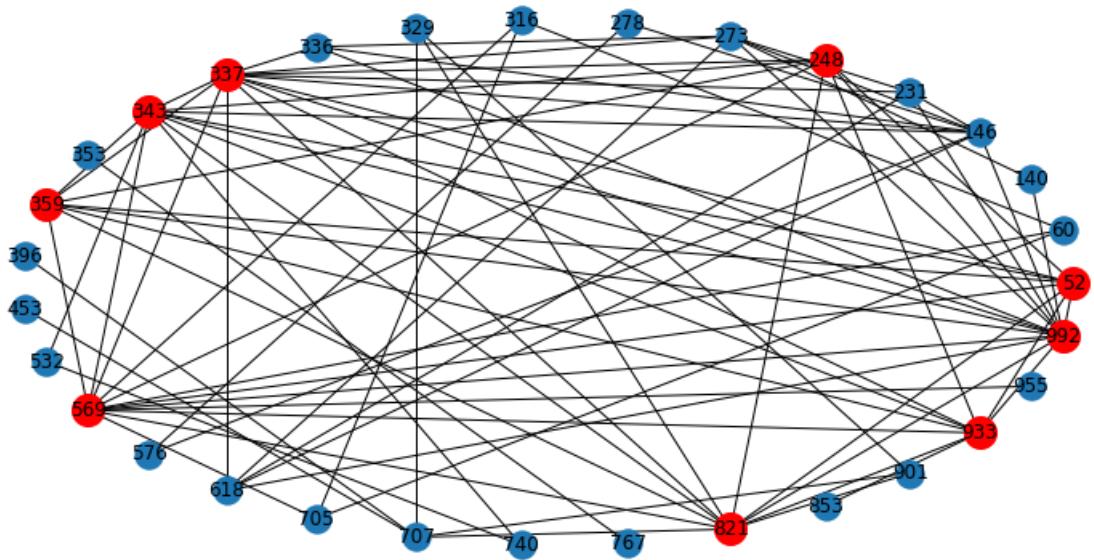


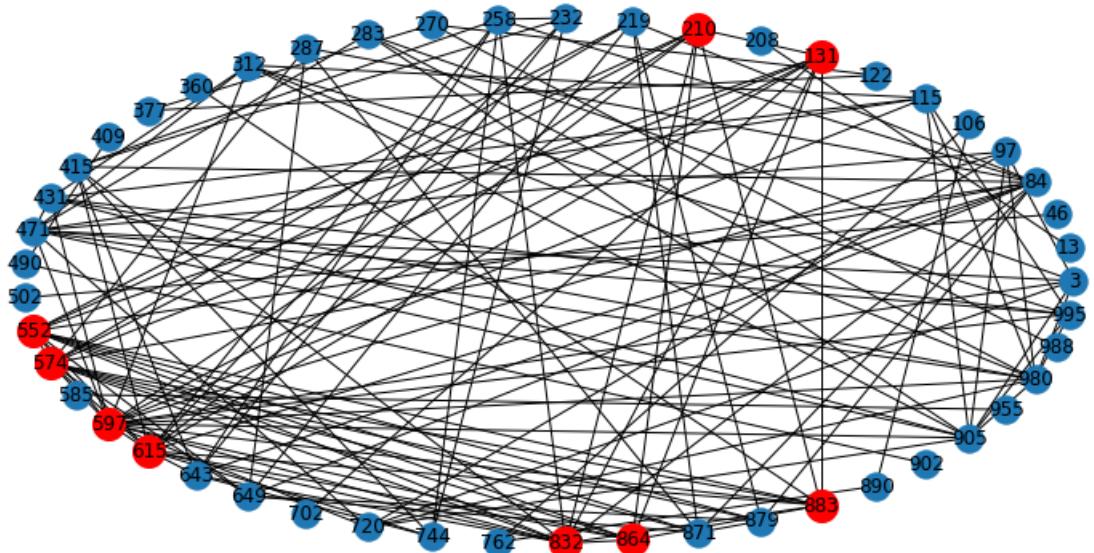
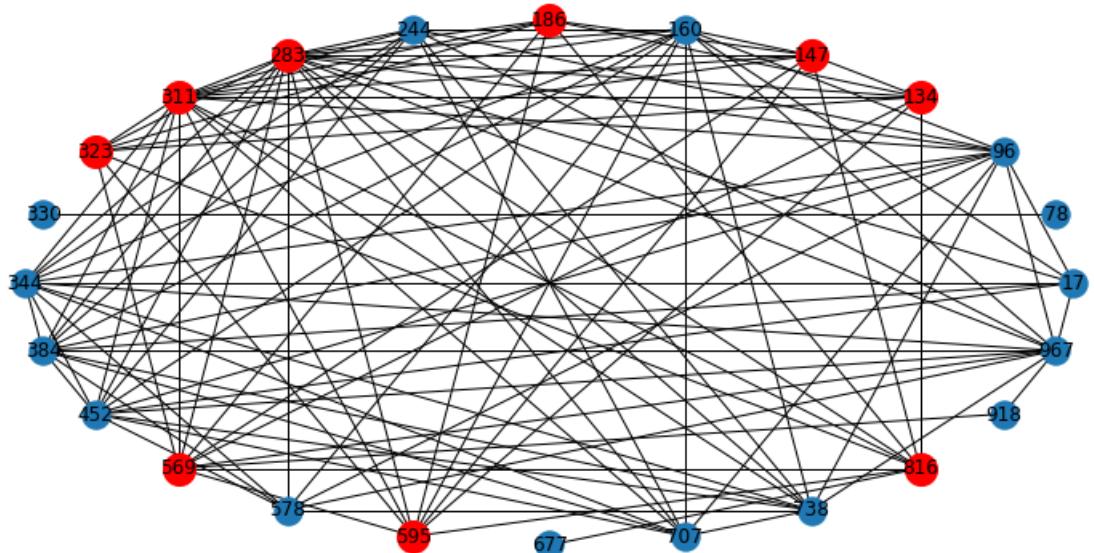
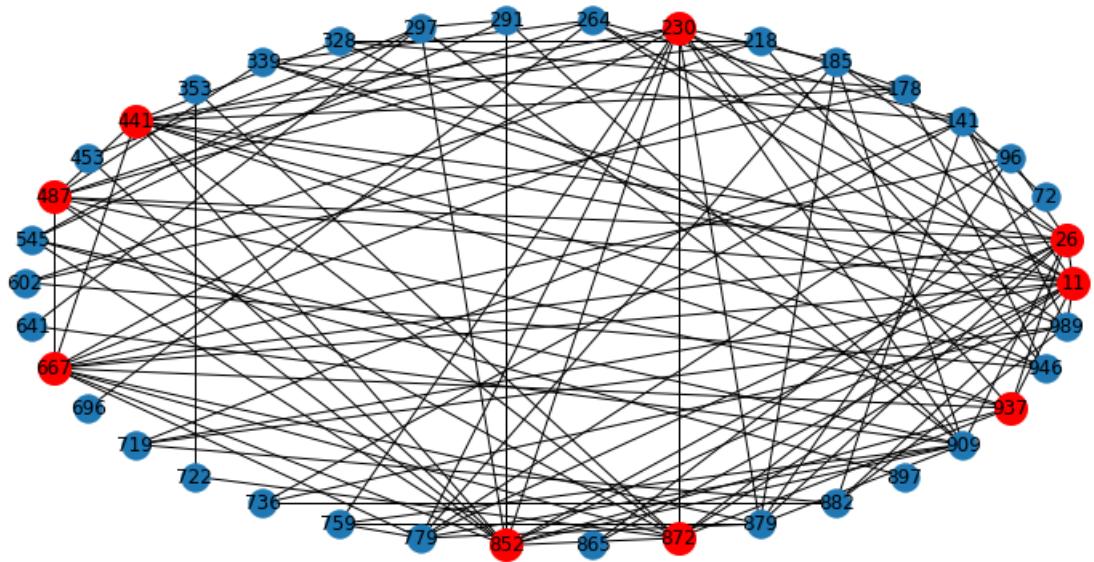


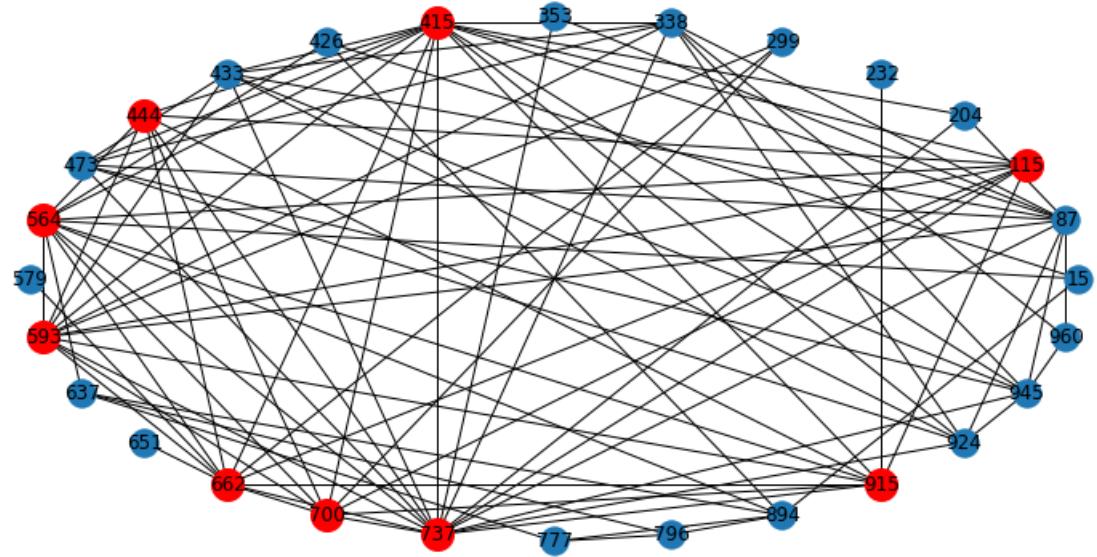
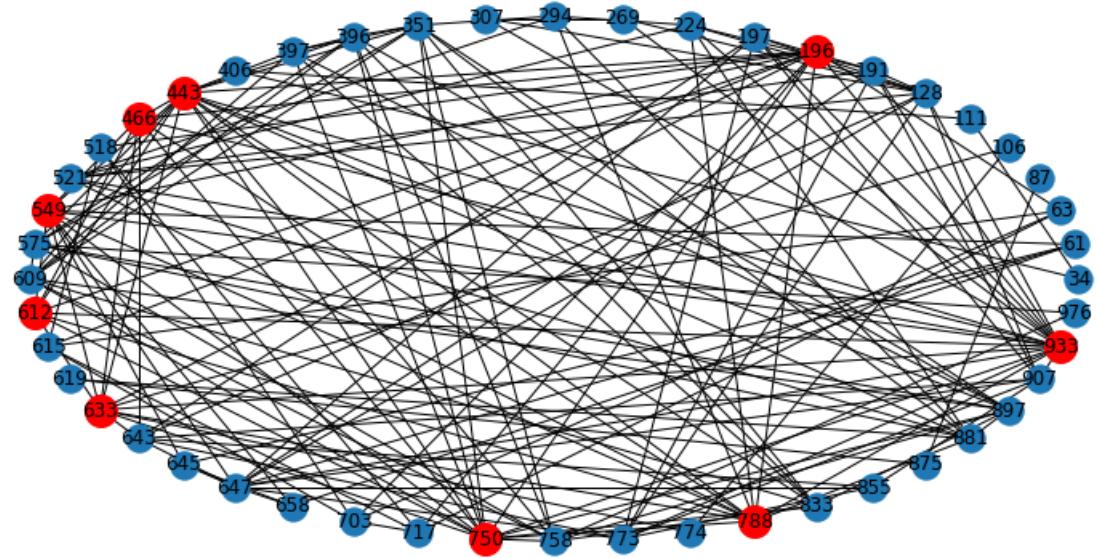
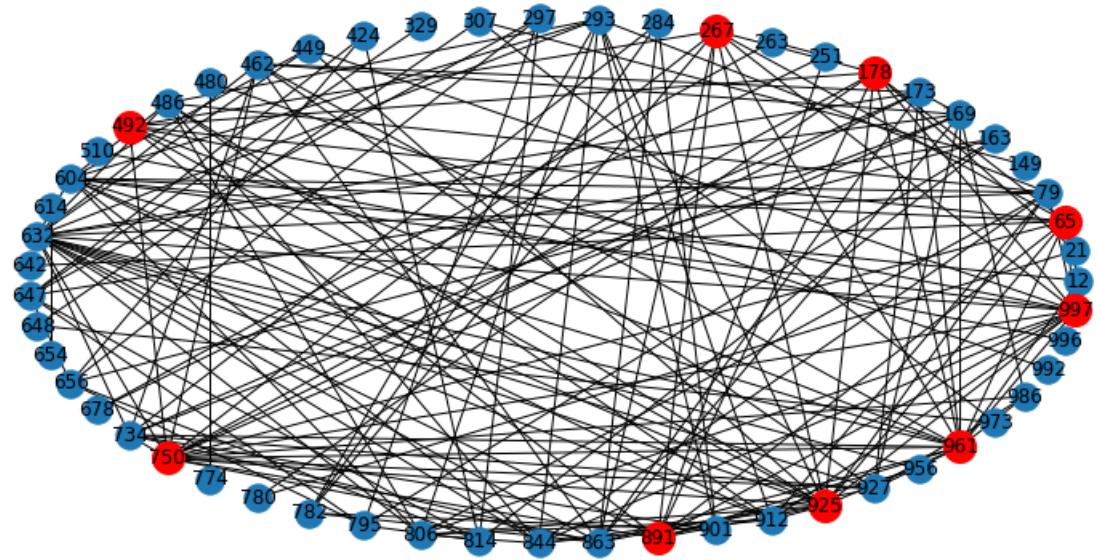
hamlet_40

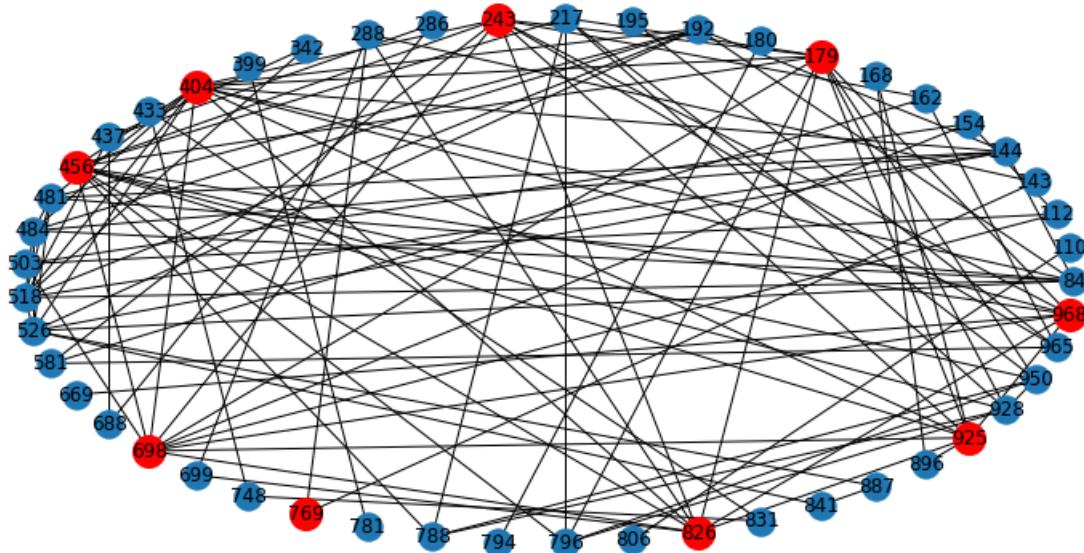
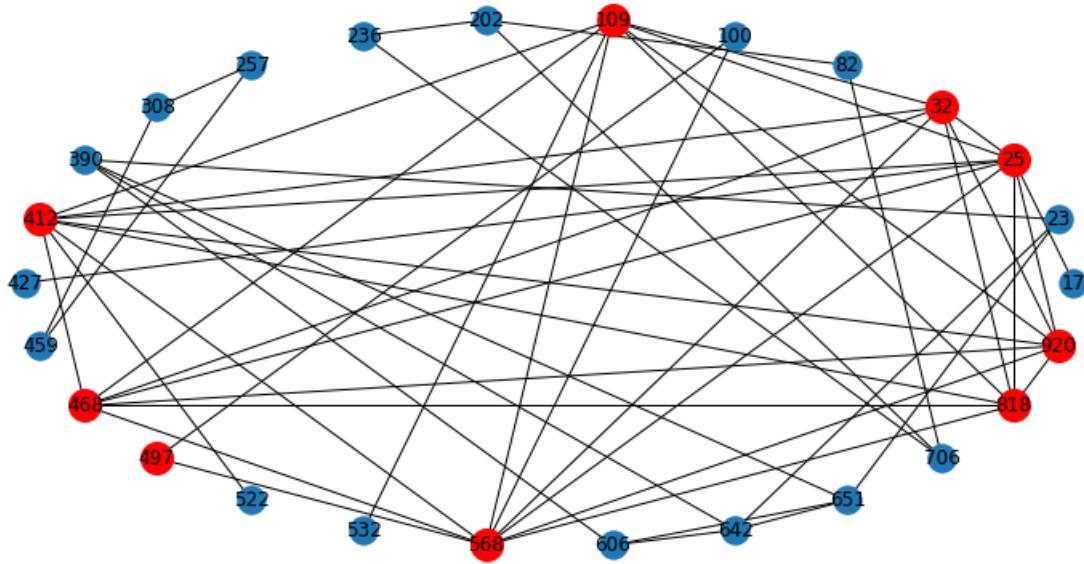




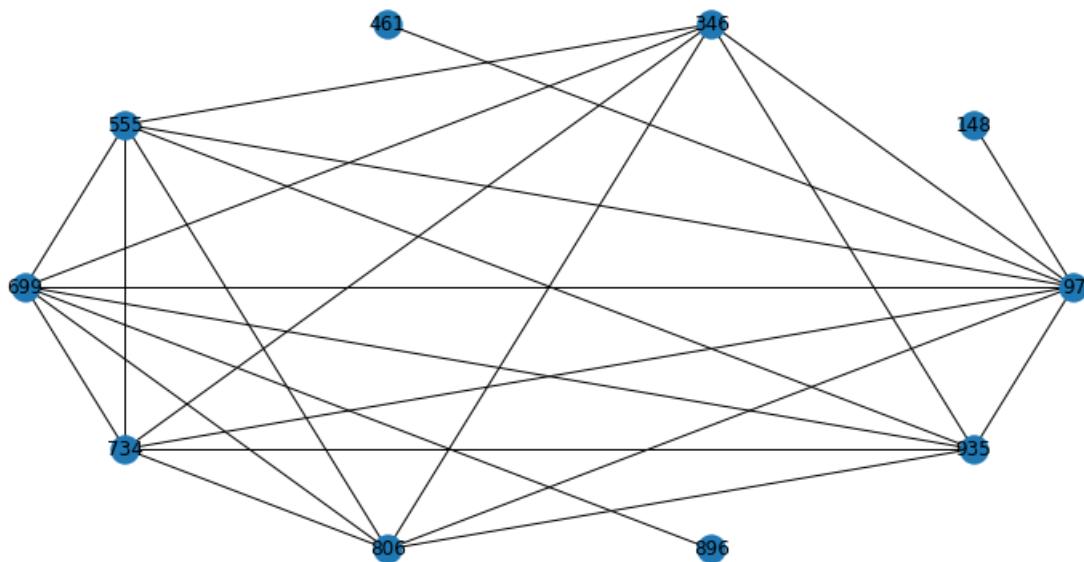


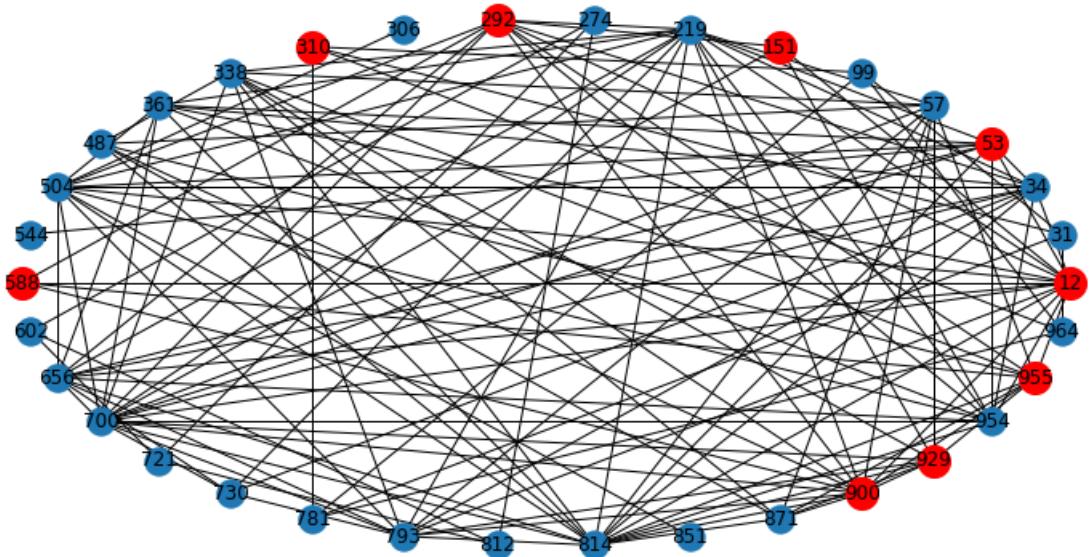
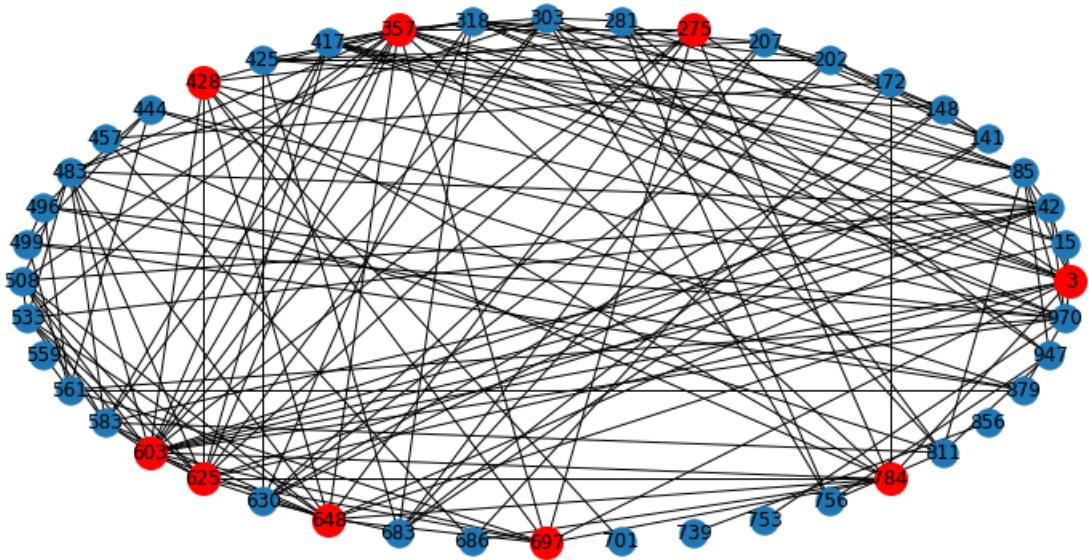




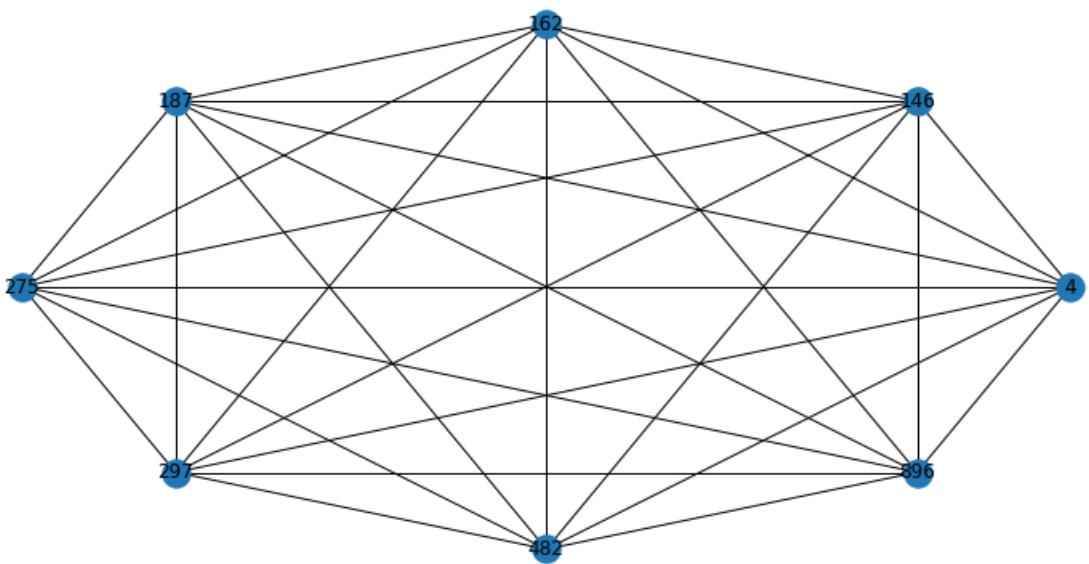


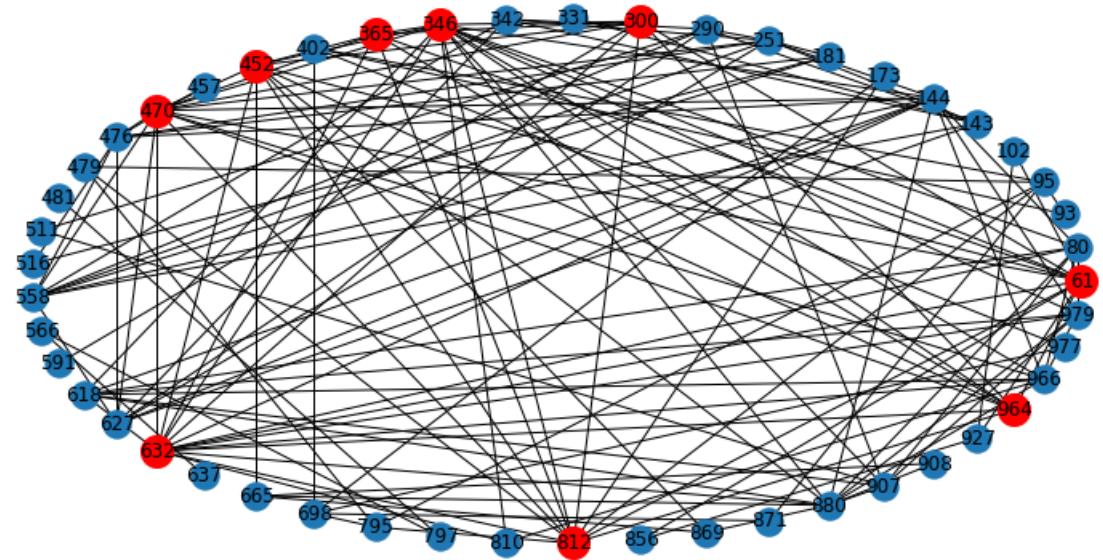
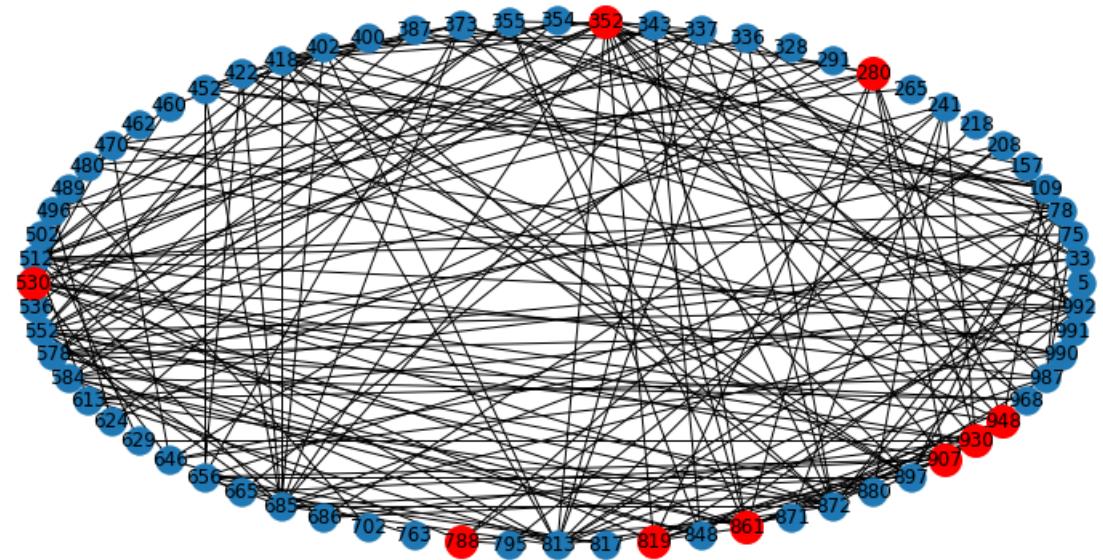
hamlet_56



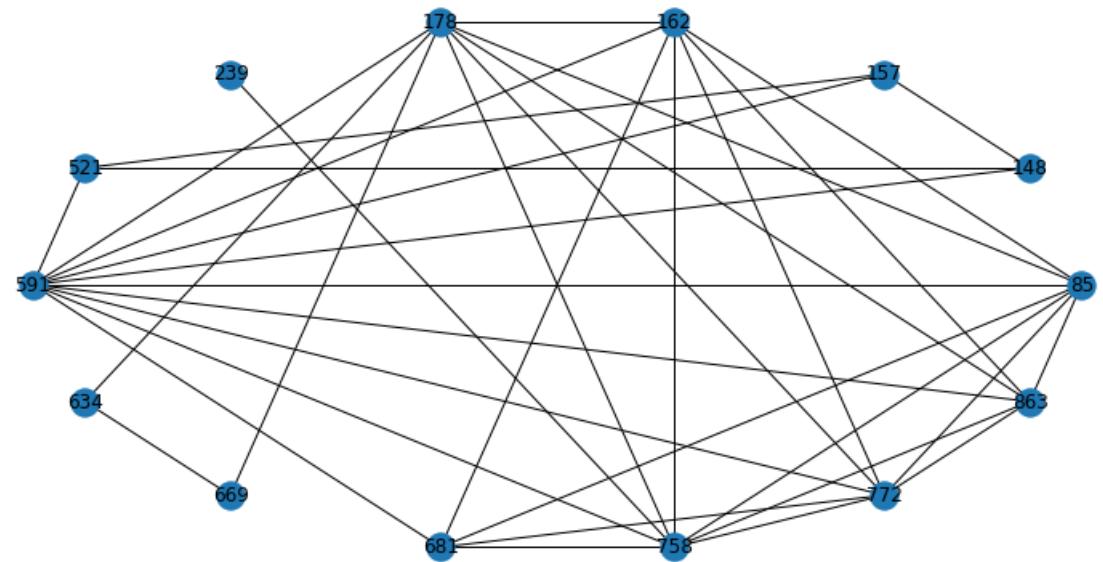


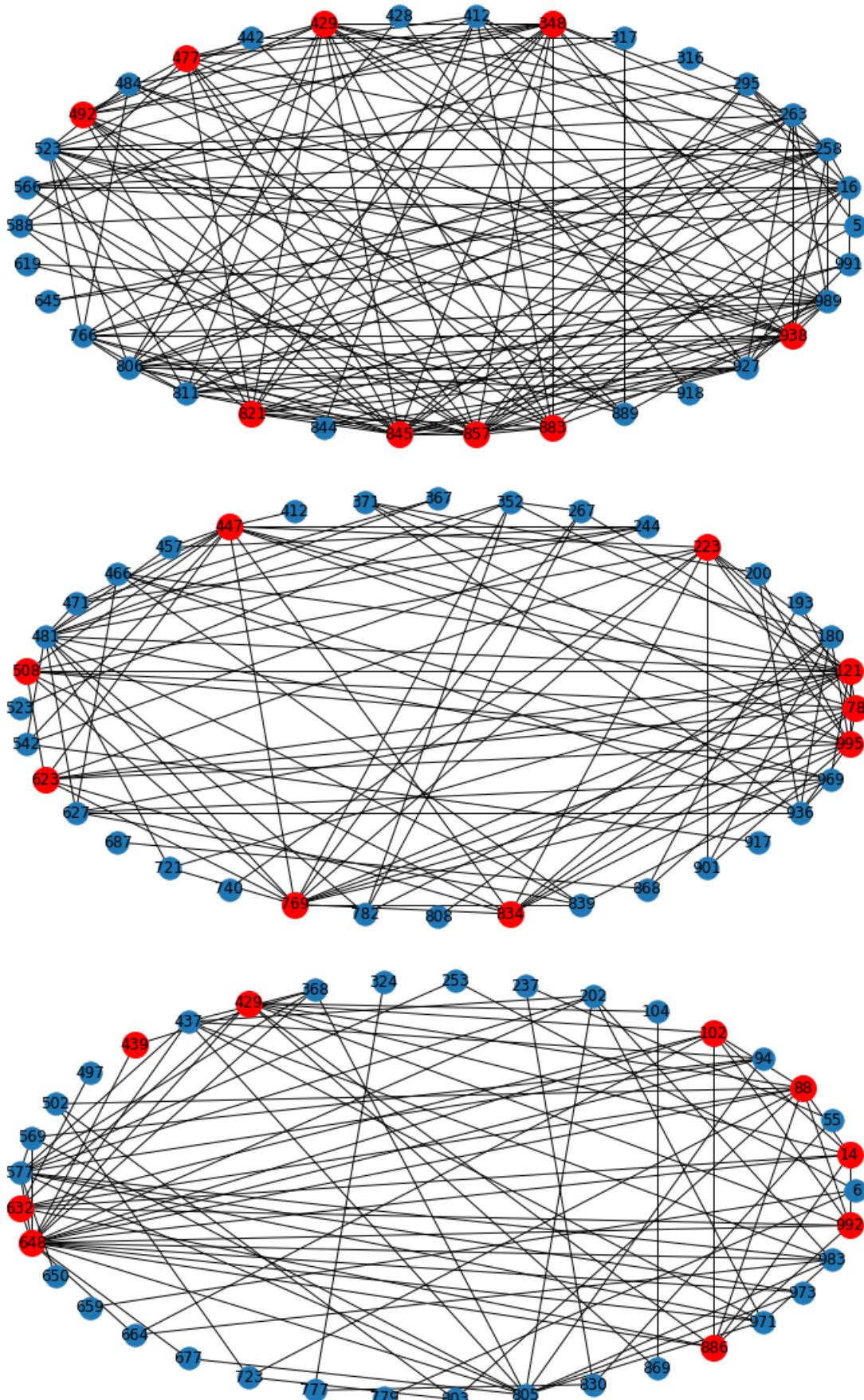
hamlet_59





hamlet_62





Export the data as a pickle file

```
In [6]: with open('../data/data_dict.pickle', 'wb') as handle:  
    pickle.dump(data_dict, handle)
```