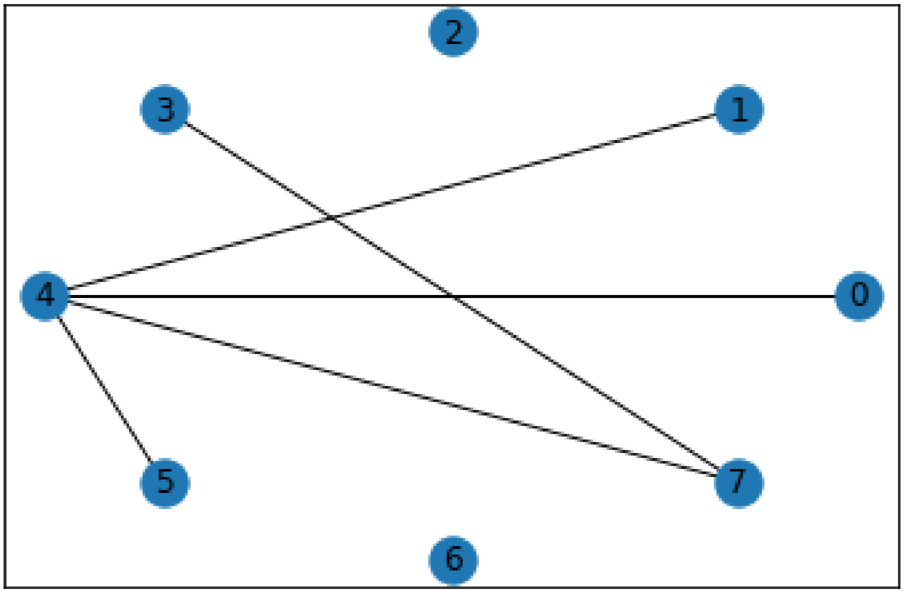
EulerCharacteristicandErdosReyniGraphs

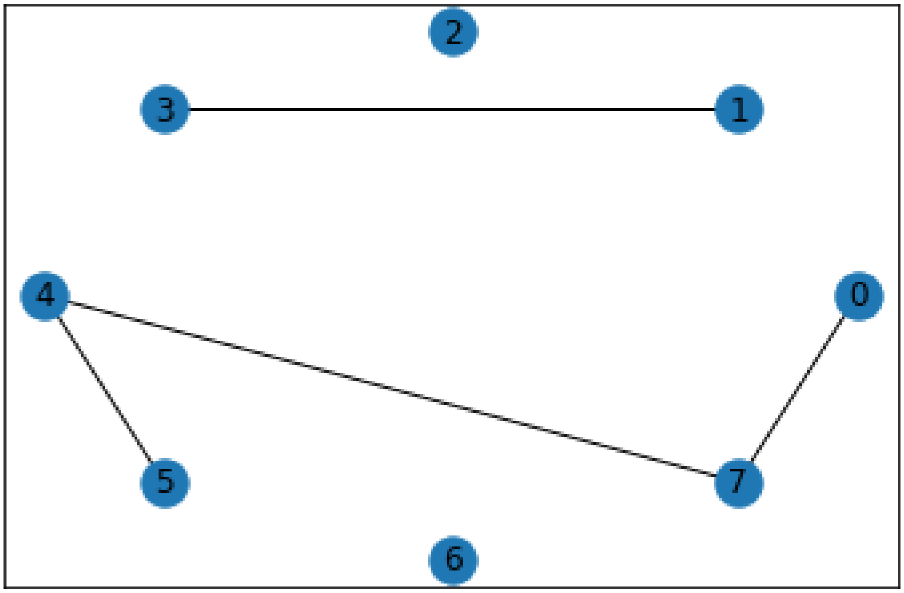
April 4, 2022

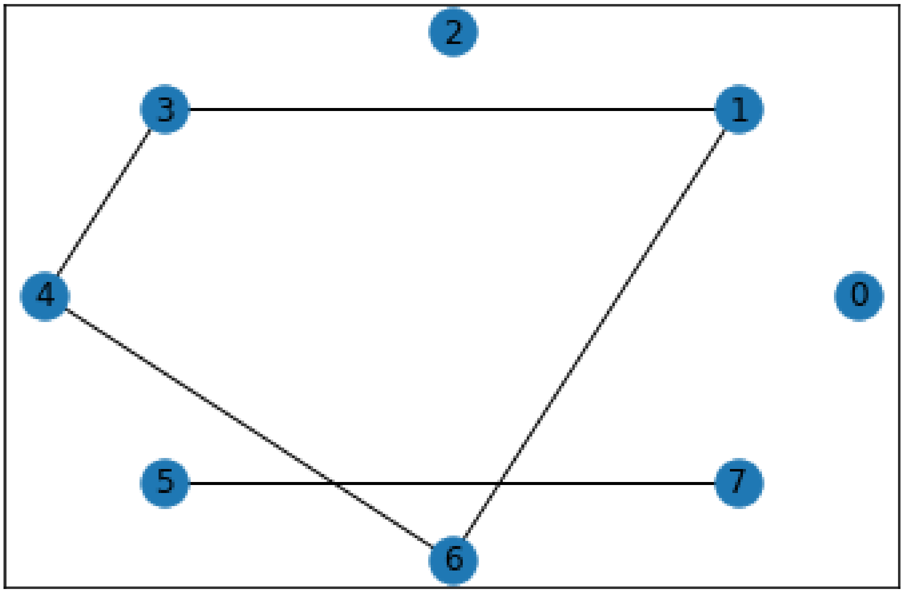
[633]:

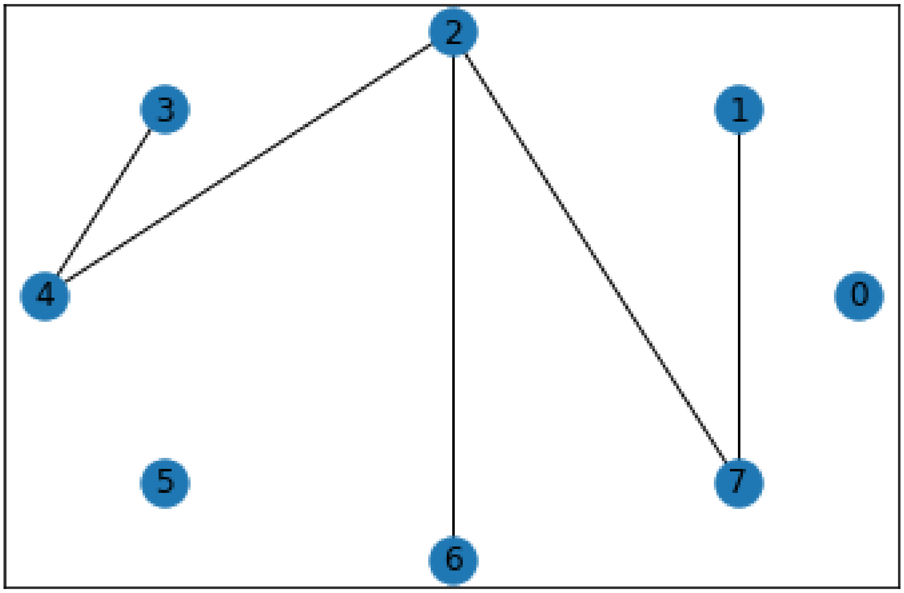
```
#Here we use netorkx's inbuilt function to generate some erdos-reyni graphs

#Import necessary libraries
import networkx as nx
import matplotlib.pyplot as plt
import numpy as np
from itertools import combinations

def expected_edge(v, p):
    "Returns the expected number of edges in an erdos reyni graph with |v|␣
 ↪vertices and probabibilty parameter p"
    return p*v*(v-1)/2


print(f"Expected value of number of edges: {expected_edge(8, .2)}")
for i in range(6):
    G = nx.erdos_renyi_graph(8, .2, seed=None, directed=False)
    print(f"Number of edges in Erdos-Reyni Graph: {G.number_of_edges()}")
    plt.figure()
    nx.draw_networkx(G, pos=nx.circular_layout(G))
```

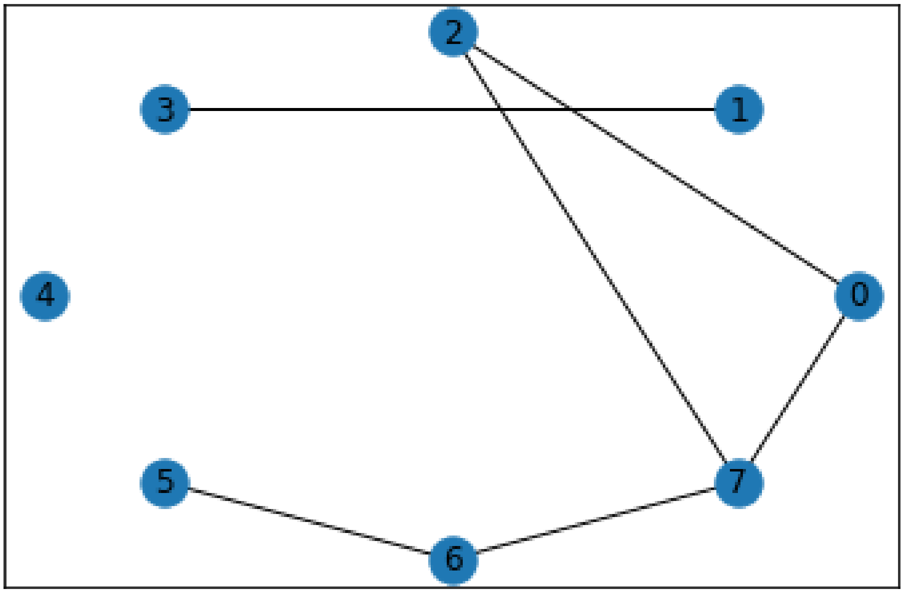
```
Expected value of number of edges: 5.6000000000000005
Number of edges in Erdos-Reyni Graph: 5
Number of edges in Erdos-Reyni Graph: 4
Number of edges in Erdos-Reyni Graph: 5
Number of edges in Erdos-Reyni Graph: 5
Number of edges in Erdos-Reyni Graph: 6
Number of edges in Erdos-Reyni Graph: 6
```

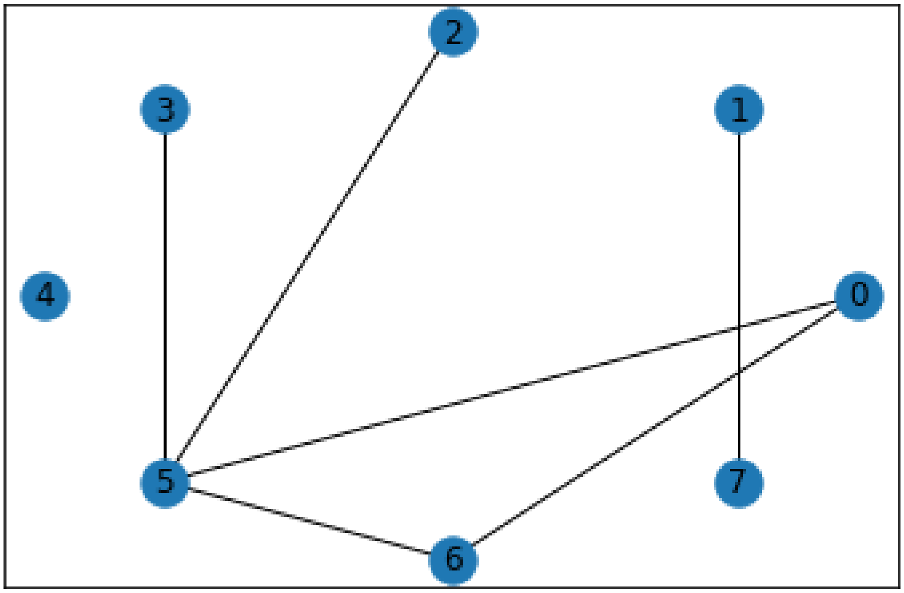

2
3
1
4
0
5
7
6


2
3
1
4
0
5
7
6


2
3
1
4
0
5
7
6


2
3
1
4
0
5
7
6





[634]:

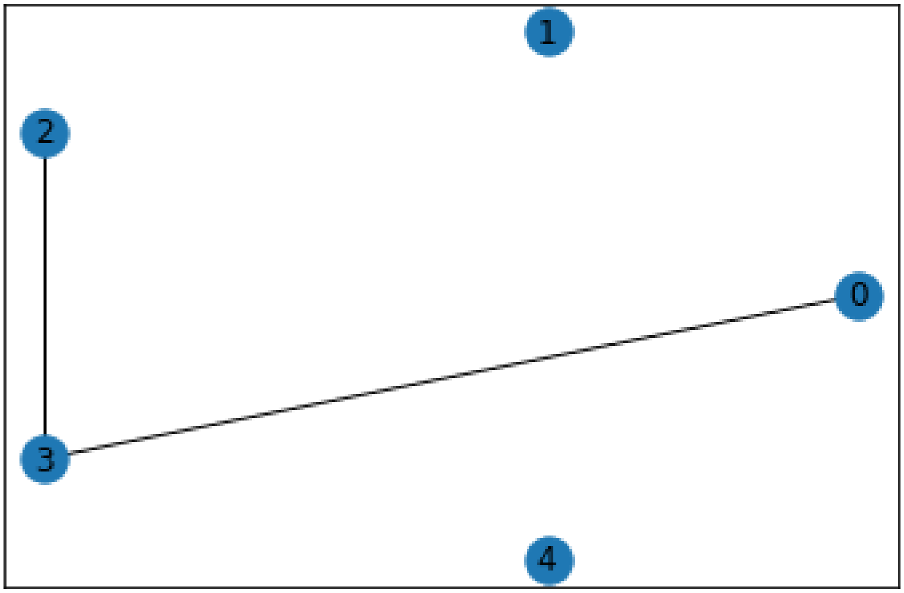
```
#Finding number of k-cliques in a Graph
#
#Found this code online at the following link: https://iq.opengenus.org/
 ↪algorithm-to-find-cliques-of-a-given-size-k/

from itertools import combinations
import networkx as nx
```

```python
def k_cliques(graph):
    # 2-cliques
    cliques = [{i, j} for i, j in graph.edges() if i != j]
    k = 2

    while cliques:
        # result
        yield k, cliques

        # merge k-cliques into (k+1)-cliques
        cliques_1 = set()
        for u, v in combinations(cliques, 2):
            w = u ^ v
            if len(w) == 2 and graph.has_edge(*w):
                cliques_1.add(tuple(u | w))

        # remove duplicates
        cliques = list(map(set, cliques_1))
        k += 1


def print_cliques(graph, size_k):
    for k, cliques in k_cliques(graph):
        if k == size_k:
            return len(cliques)
```

[635]:

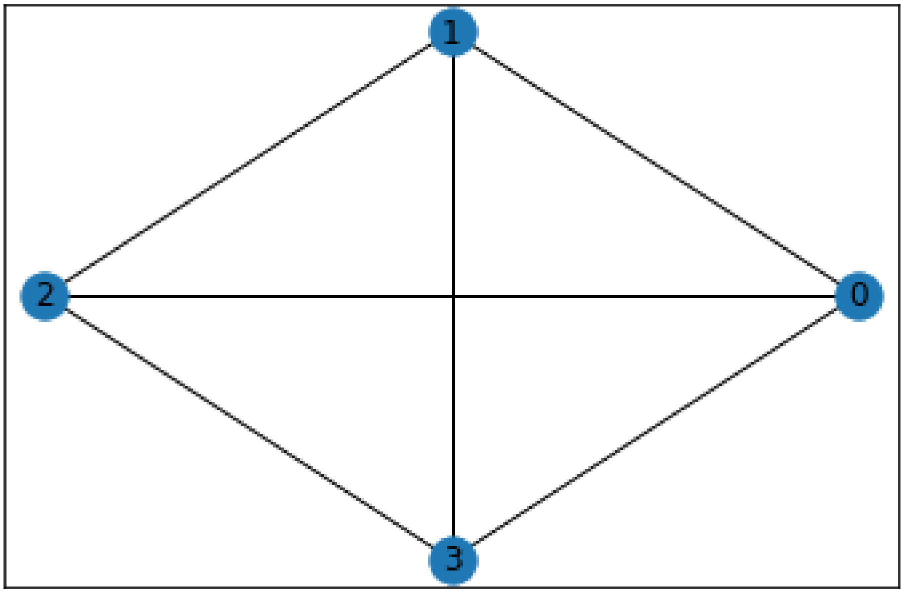
```python
#Generate graph with some k-cliques

G = nx.erdos_renyi_graph(5, .5, seed=None, directed=False)

size_k = 1

print(type(print_cliques(G, size_k)))

print(f"Number of {size_k} cliques in graph: {print_cliques(G, size_k)}")

nx.draw_networkx(G, pos=nx.circular_layout(G))
```

```
<class 'NoneType'>
Number of 1 cliques in graph: None
```



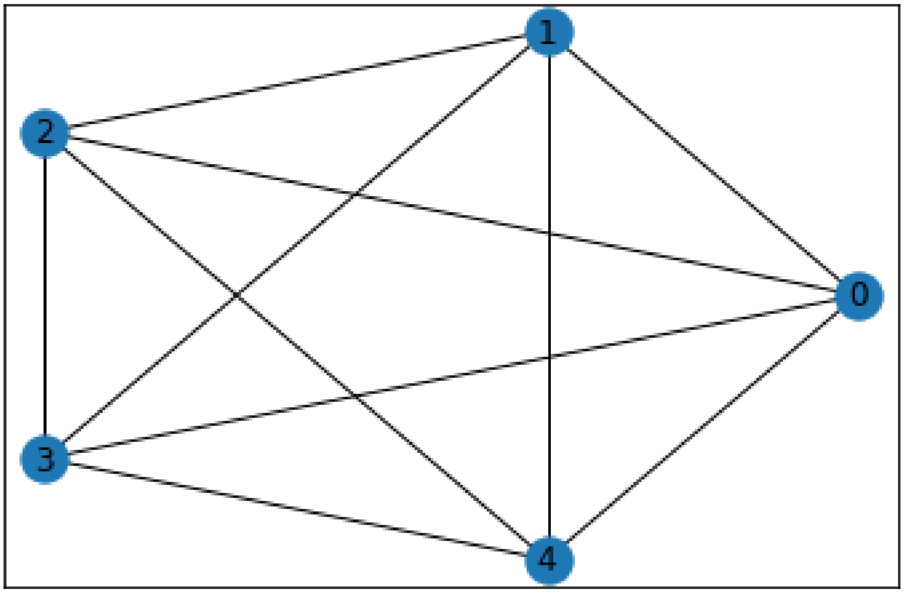
[646]:
```
#NEW AND IMPROVED EULER CHARACTERISTIC

def Euler_Char(G):
    chi = 0
    chi = chi + G.number_of_nodes()
    #print(print_cliques(G, 3))
    for i in range(2, G.number_of_nodes()+1):
        if type(print_cliques(G, i)) == type(1):
            chi = chi + (-1)**(i+1) * print_cliques(G, i)
        else:
            break
    return chi

G = nx.erdos_renyi_graph(4, .9, seed=None, directed=False)

print(f"Euler Characteristic of G: {Euler_Char(G)}")
nx.draw_networkx(G, pos=nx.circular_layout(G))
```

```
Euler Characteristic of G: 1
```



```
[647]: G = nx.erdos_renyi_graph(5, .9, seed=None, directed=False)


       print(f"Euler Characteristic of G: {Euler_Char(G)}")
       nx.draw_networkx(G, pos=nx.circular_layout(G))
```

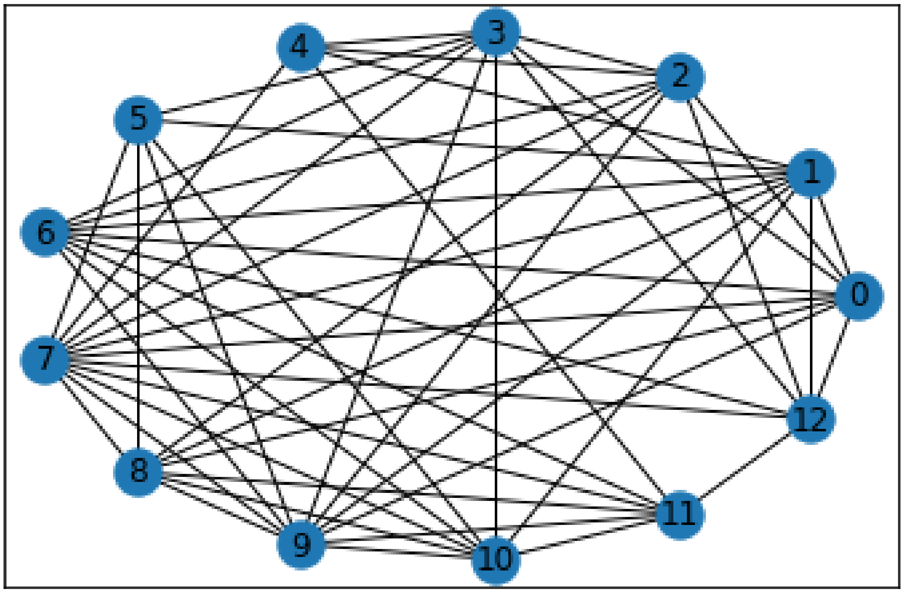
```
Euler Characteristic of G: 1
```



[652]:

```
#EULER CHARACTERISTIC CONTINUED

G = nx.erdos_renyi_graph(13, .6, seed=None, directed=False)


print(f"Euler Characteristic of G: {Euler_Char(G)}")
nx.draw_networkx(G, pos=nx.circular_layout(G))
```

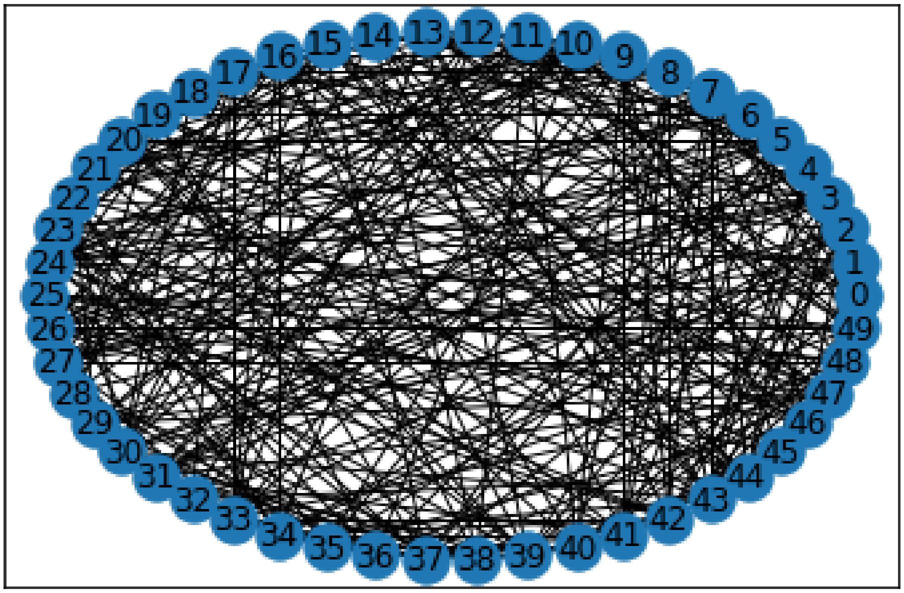
```
Euler Characteristic of G: 6
```



[539]:

```
#EULER CHARACTERISTIC CONTINUED

G = nx.erdos_renyi_graph(50, .25, seed=None, directed=False)


print(f"Euler Characteristic of G: {Euler_Char(G)}")
nx.draw_networkx(G, pos=nx.circular_layout(G))
```

```
Euler Characteristic of G: 0
```


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49