SOCIAL NETWORK ANALYSIS
ASSIGNMENT 3

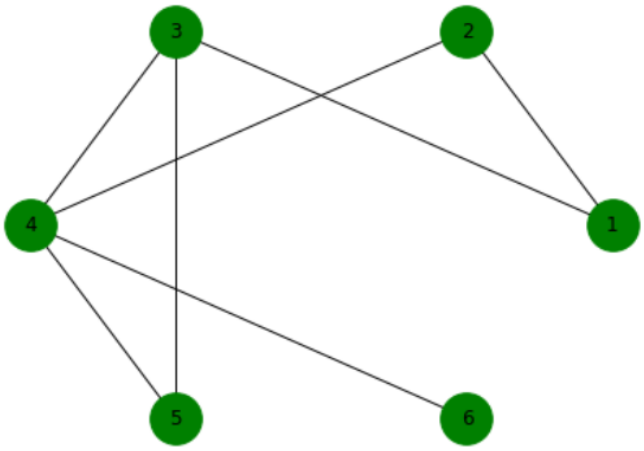
U20CS005
BANSI MARAKANA

Generate different types of graphs shown below using NetworkX and plot them. Also display the adjacency matrix of these graphs.

1) Undirected Graph

```
import networkx as nx
import matplotlib.pyplot as plt
g = nx.Graph()
g.add_edges_from([(1,2),(1,3),(2,4),(3,4),(3,5),(4,5),(4,6)])
nx.draw_circular(g, with_labels = True, node_color ='green',node_size = 1000)
A = nx.adjacency_matrix(g)
A.setdiag(A.diagonal() * 2)
print(A.todense())
```

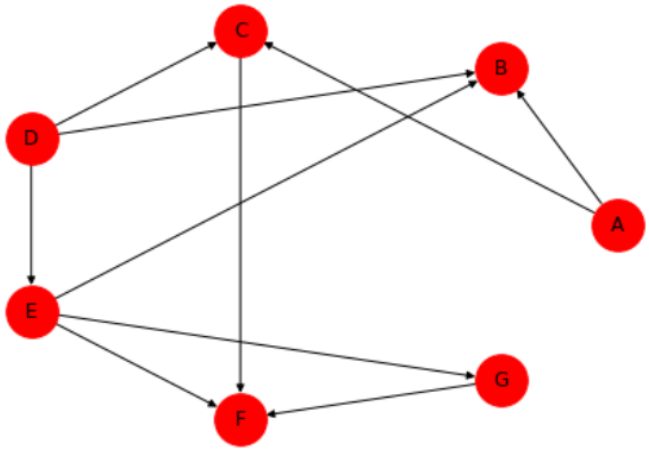
```
[[0 1 1 0 0 0]
 [1 0 0 1 0 0]
 [1 0 0 1 1 0]
 [0 1 1 0 1 1]
 [0 0 1 1 0 0]
 [0 0 0 1 0 0]]
```



2) Directed Graph

```
import networkx as nx
import matplotlib.pyplot as plt
g = nx.DiGraph()
g.add_edges_from([("A","B"),("A","C"),("D","B"),("E","B"),("D","E"),("E","F"),("E","G"),("G","F"),("D","C"),("C","F")])
nx.draw_circular(g, with_labels = True, node_color ='red',node_size = 1000)
A = nx.adjacency_matrix(g)
print(A.todense())
```

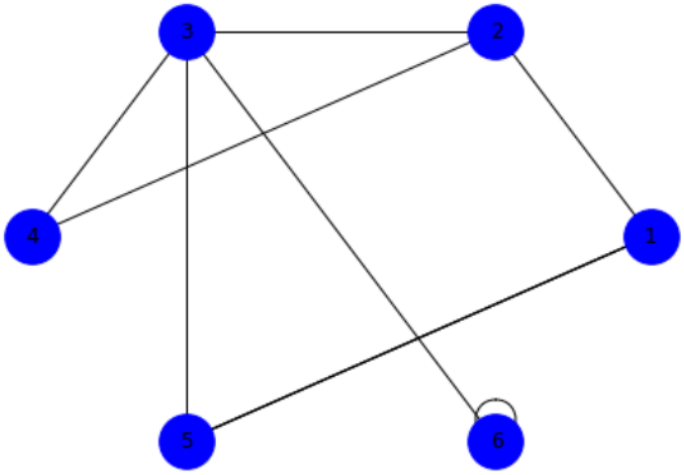
```
[[0 1 1 0 0 0 0]
 [0 0 0 0 0 0 0]
 [0 0 0 0 0 1 0]
 [0 1 1 0 1 0 0]
 [0 1 0 0 0 1 1]
 [0 0 0 0 0 0 0]
 [0 0 0 0 0 1 0]]
```



3) Multigraph

```
import networkx as nx
import matplotlib.pyplot as plt
g = nx.MultiGraph()
g.add_nodes_from([1,2,3,4,5])
g.add_edges_from([(1,5),(1,5),(1,5),(1,2),(5,3),(2,3),(2,3),(2,4),(3,4),(3,6),(6,6)])
pos = nx.circular_layout(g)
nx.draw(g,pos, with_labels = True, node_color ='blue',node_size = 1000)
A = nx.adjacency_matrix(g)
A.setdiag(A.diagonal() * 2)
print(A.todense())
```

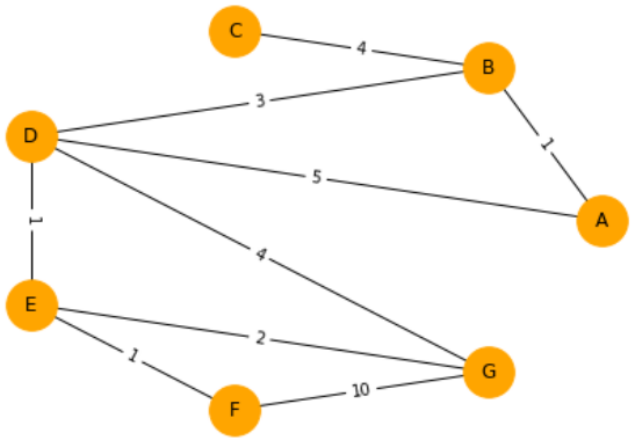
```
[[0 1 0 0 3 0]
 [1 0 2 1 0 0]
 [0 2 0 1 1 1]
 [0 1 1 0 0 0]
 [3 0 1 0 0 0]
 [0 0 1 0 0 2]]
```



4) Weighted Graph

```
import networkx as nx
import matplotlib.pyplot as plt
g = nx.Graph()
g.add_nodes_from(['A','B','C','D','E','F','G'])
g.add_edge('A','B',weight=1)
g.add_edge('B','D',weight=3)
g.add_edge('C','B',weight=4)
g.add_edge('A','D',weight=5)
g.add_edge('D','E',weight=1)
g.add_edge('D','G',weight=4)
g.add_edge('E','G',weight=2)
g.add_edge('G','F',weight=10)
g.add_edge('E','F',weight=1)
weight_labels = nx.get_edge_attributes(g,'weight')
pos = nx.circular_layout(g)
nx.draw(g,pos, with_labels = True, node_color ='orange',node_size = 1000)
nx.draw_networkx_edge_labels(g,pos,edge_labels=weight_labels)
A = nx.adjacency_matrix(g)
A.setdiag(A.diagonal() * 2)
print(A.todense())
```

```
[[ 0  1  0  5  0  0  0]
 [ 1  0  4  3  0  0  0]
 [ 0  4  0  0  0  0  0]
 [ 5  3  0  0  1  0  4]
 [ 0  0  0  1  0  1  2]
 [ 0  0  0  0  1  0 10]
 [ 0  0  0  4  2 10  0]]
```



5) Complete Graph

```
import networkx as nx
import matplotlib.pyplot as plt
g = nx.complete_graph(5)
mapping = {0: "A", 1: "B", 2: "C", 3: "D", 4: "E"}
g = nx.relabel_nodes(g, mapping, copy=False)
nx.draw(g, node_color = 'purple',node_size = 1000,with_labels = True)
print(A.todense())
```

```
[[ 0  1  0  5  0  0  0]
 [ 1  0  4  3  0  0  0]
 [ 0  4  0  0  0  0  0]
 [ 5  3  0  0  1  0  4]
 [ 0  0  0  1  0  1  2]
 [ 0  0  0  0  1  0 10]
 [ 0  0  0  4  2 10  0]]
```