Create an empty graph G.

Add the following nodes: A, B, C, D

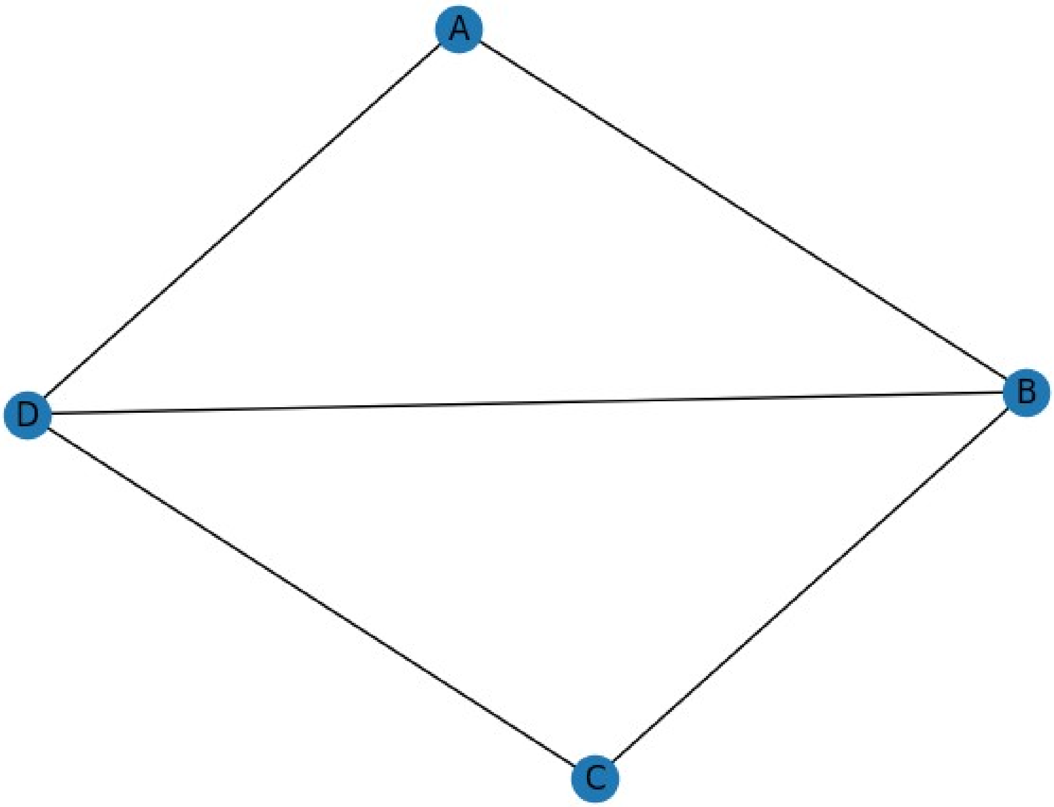
Add the following edges: A-B, B-C, C-D, D-A, B-D

```
import networkx as nx

g = nx.Graph()
g.add_nodes_from(['A', 'B', 'C', 'D'])
g.add_edges_from([('A', 'B'), ('B', 'C'), ('C', 'D'), ('D', 'A'),
('B', 'D')])
```

Visualize the Graph

```
import matplotlib.pyplot as plt
nx.draw(g, with_labels=True)
plt.show()
```



Add an attribute color to node A with value 'red'.

Add a weight attribute to edge (B, D) with value 5.

```
g.nodes['A']['color'] = 'red'
g.edges[('B', 'D')]['weight'] = 5
```

How many nodes are there in the graph?

How many edges are there?

List all the neighbors of node B.

```
print(f"Number of nodes: {g.number_of_nodes()}")
print(f"Number of edges: {g.number_of_edges()}")
print(f"Neighbors of node B: {list(g.neighbors('B'))}")
```

```
Number of nodes: 4
Number of edges: 5
Neighbors of node B: ['A', 'C', 'D']
```

Find the shortest path between node A and node C (assuming unweighted edges).

```
print(f"Shortest path between A and C: {nx.shortest_path(g, 'A', 'C')}")
```

```
Shortest path between A and C: ['A', 'B', 'C']
```