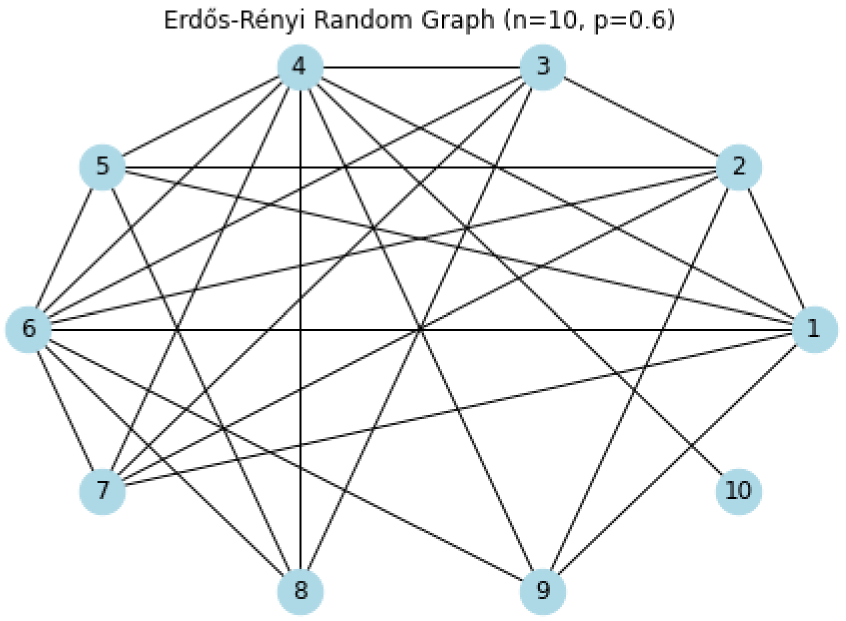
In [40]:

```python
import networkx as nx
import matplotlib.pyplot as plt
import random

print('Enter number of nodes:')
N = int(input())
print('Enter the probability of edge creation:')
P = float(input())

# Create an empty graph
g = nx.Graph()
g.add_nodes_from(range(1, N + 1))  # Add nodes

# Add edges based on probability
for i in g.nodes():
    for j in g.nodes():
        if i < j:  # Avoid self-loops and duplicate edges
            R = random.random()
            if R < P:
                g.add_edge(i, j)
    pos = nx.circular_layout(g)  # Circular layout for better visualiza
    nx.draw(g, pos, with_labels=True, node_size=500, node_color='lightb
    plt.title(f"Erdős-Rényi Random Graph (n={N}, p={P})")
    plt.show()
```



write a program to genrate a random graph and also check the beahvior of giant component as per user input

In [41]:

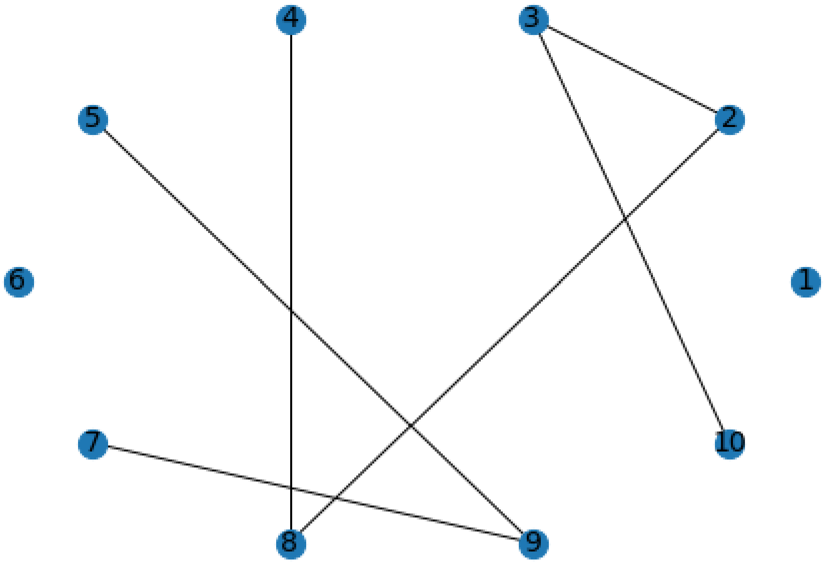
```
#Creation of giant components
#Taking nodes and probability input from the user
print('Enter number of nodes: ')
n = int(input())
print('Enter value for d: ')
d = float(input())
p = d/n
print("Probability: ",p)
```

```
Enter number of nodes:
10
Enter value for d:
1.5
Probability:  0.15
```

In [43]:

```
g = nx.Graph()
g.add_nodes_from(range(1, n+1))

for i in g.nodes():
    for j in g.nodes():
        if i<j:
            r = random.random()
            if r<p:
                g.add_edge(i, j)
pos = nx.circular_layout(g)
nx.draw(g, pos, with_labels = 1, node_size=200,font_size=14)
plt.figure(figsize = (6,4))
plt.show()
```



```
<Figure size 432x288 with 0 Axes>
```

In [44]:

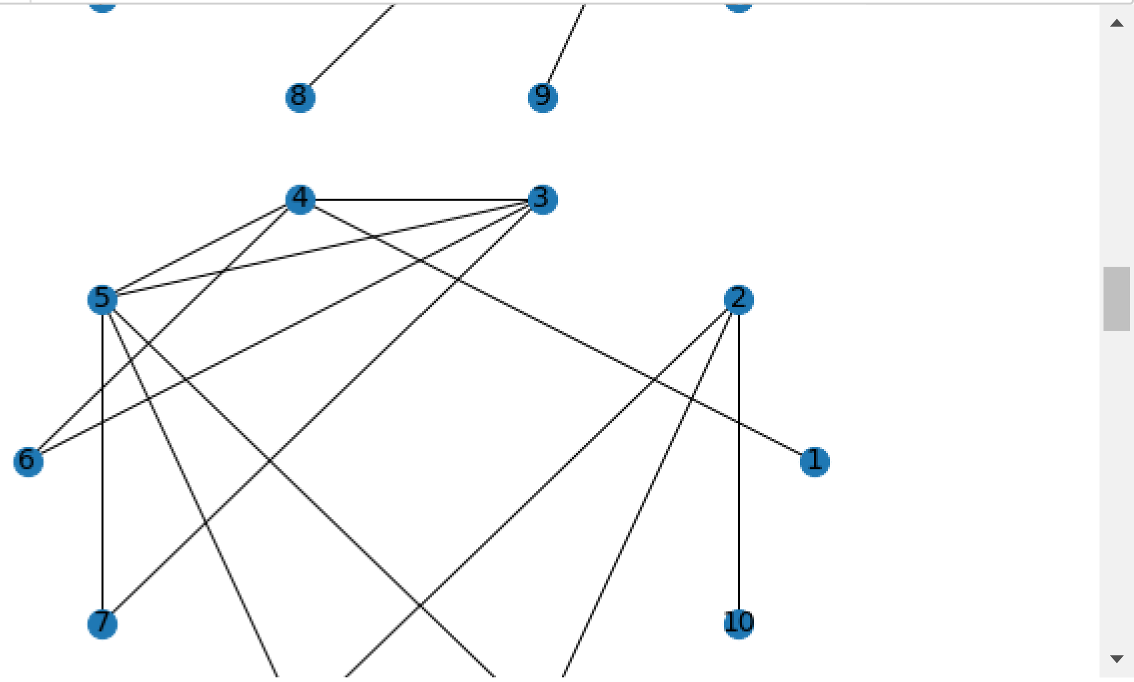
```
#Taking nodes and probability input from the user
#Creating random graph to check for cycle
print('Enter number of nodes: ')
n = int(input())
print('Enter value for d: ')
d = float(input())
p = d/n
```

```
Enter number of nodes:
10
Enter value for d:
1.5
```

In [48]:

```
g = nx.Graph()
g.add_nodes_from(range(1, n+1))

for i in g.nodes():
  for j in g.nodes():
    if i<j:
      r = random.random()
      if r<p:
        g.add_edge(i, j)
  pos = nx.circular_layout(g)
  nx.draw(g, pos, with_labels = 1, node_size=200,font_size=14)
  plt.show()
```



In [*]:

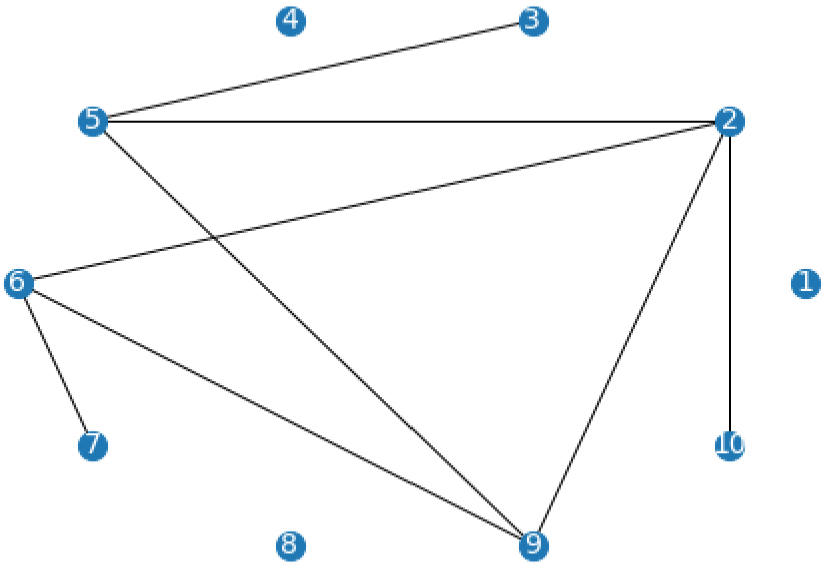
```
#Disappearance of isolated components
#Taking nodes input from the user
import math
print('Enter number of nodes: ')
n = int(input())
p = (math.log(n) / n)
print('Probability : ', p)
```

```
Enter number of nodes:
```

In [47]:

```
#Creating a graph and adding nodes
g = nx.Graph()
g.add_nodes_from(range(1, n+1))

for i in g.nodes():
  for j in g.nodes():
    if i<j:
      r = random.random()
      if r<p:
        g.add_edge(i, j)
pos = nx.circular_layout(g)
nx.draw(g, pos, with_labels = 1, node_size=200,font_size=14, font_color
plt.show()
```



```
<Figure size 432x288 with 0 Axes>
```

In [ ]: