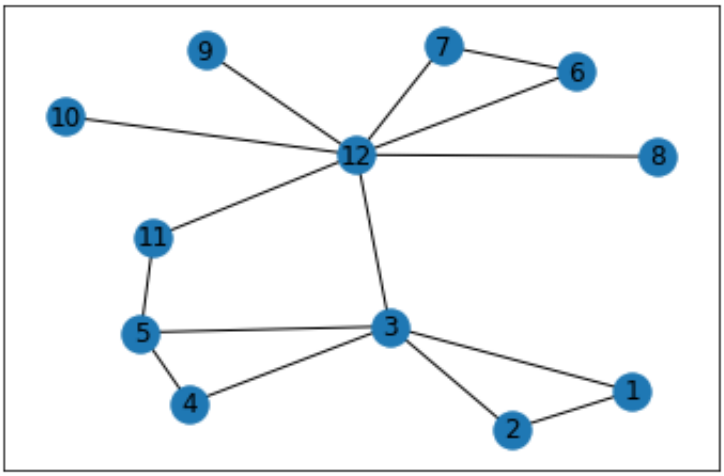
In [3]:
```
import numpy as np
import matplotlib.pyplot as plt

import networkx as nx

G = nx.Graph()
G.add_edge('1','2')
G.add_edge('1','3')
G.add_edge('2','3')
G.add_edge('3','4')
G.add_edge('4','5')
G.add_edge('3','5')
G.add_edge('3','12')
G.add_edge('12','6')
G.add_edge('12','11')
G.add_edge('12','10')
G.add_edge('12','9')
G.add_edge('12','8')
G.add_edge('12','7')
G.add_edge('6','7')
G.add_edge('11','5')
nx.draw_networkx(G)
```



In [4]:
```
A = nx.adjacency_matrix(G)
A.todense()
```

Out[4]:
```
matrix([[0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0],
        [0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0],
        [0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0],
        [0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1],
        [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1],
        [0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0]])
```

In [5]:
```
deg_view = nx.degree(G)
deg_vals = dict(deg_view).values()

#PROBLEM 5    ANSWER: 3 -> .49 12->.73
nx.betweenness_centrality(G)
```

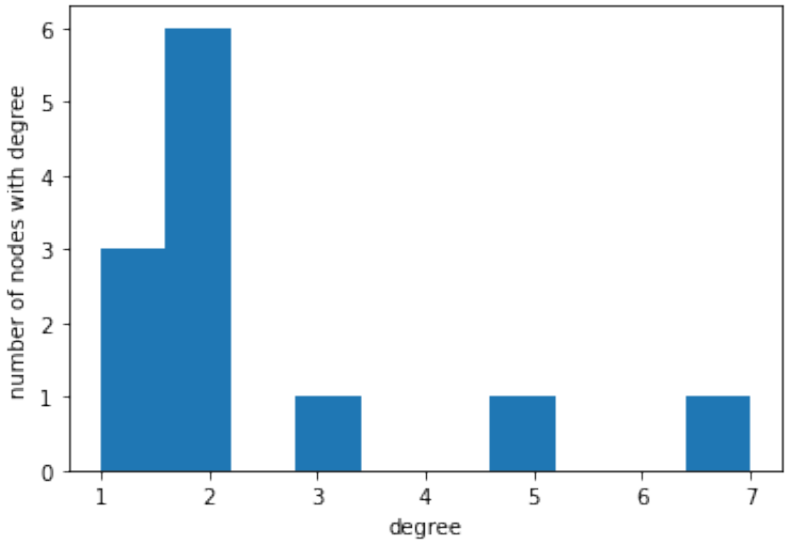
Out[5]:
```
{'1': 0.0,
 '2': 0.0,
 '3': 0.4909090909090909,
 '4': 0.0,
 '5': 0.045454545454545456,
 '12': 0.7363636363636363,
 '6': 0.0,
 '11': 0.05454545454545454,
 '10': 0.0,
 '9': 0.0,
 '8': 0.0,
 '7': 0.0}
```

In [6]:
```
#PROBLEM 6   3-> .47 12->.53
nx.eigenvector_centrality(G)
```

Out[6]:
```
{'1': 0.210864542832372,
 '2': 0.210864542832372,
 '3': 0.4652870478907011,
 '4': 0.23879994540475266,
 '5': 0.3004407253859519,
 '12': 0.5310014523926223,
 '6': 0.2406493040387847,
 '11': 0.25929496322149226,
 '10': 0.16559953035137093,
 '9': 0.16559953035137093,
 '8': 0.16559953035137093,
 '7': 0.2406493040387847}
```

In [7]:
```
#PROBLEM 7
deg_view = nx.degree(G)
deg_vals = dict(deg_view).values()
plt.hist(deg_vals)
plt.xlabel('degree')
plt.ylabel('number of nodes with degree')
```

Out[7]:
```
Text(0, 0.5, 'number of nodes with degree')
```



In [ ]:
```
```