

```
1 import numpy as np
2 class Graph:
3     def __init__(self,vertices):
4         self._adjMat = np.zeros((vertices, vertices))
5         self._vertices = vertices
6
7     def insert_edge(self,u,v,w=1):
8         self._adjMat[u][v] = w
9
10    def delete_edge(self,u,v):
11        self._adjMat[u][v] = 0
12
13    def get_edge(self,u,v):
14        return self._adjMat[u][v]
15
16    def vertices_count(self):
17        return self._vertices
18
19    def edge_count(self):
20        count = 0
21        for i in range(self._vertices):
22            for j in range(self._vertices):
23                if not self._adjMat[i][j] == 0:
24                    count += 1
25        return count
26
27    def indegree(self,u):
28        count = 0;
29        for i in range(self._vertices):
30            if not self._adjMat[i][u] == 0:
31                count += 1
32        return count
33
34    def outdegree(self,u):
35        count = 0;
36        for i in range(self._vertices):
37            if not self._adjMat[u][i] == 0:
38                count += 1
39        return count
40
41    def display(self):
42        print(self._adjMat)
43
44
45 G = Graph(7)
46 print('Graph Adjacency Matrix')
47 G.display()
```

```
48 G.insert_edge(0,1)
49 G.insert_edge(0,5)
50 G.insert_edge(0,6)
51 G.insert_edge(1,0)
52 G.insert_edge(1,2)
53 G.insert_edge(1,5)
54 G.insert_edge(1,6)
55 G.insert_edge(2,3)
56 G.insert_edge(2,4)
57 G.insert_edge(2,6)
58 G.insert_edge(3,4)
59 G.insert_edge(4,2)
60 G.insert_edge(4,5)
61 G.insert_edge(5,2)
62 G.insert_edge(5,3)
63 G.insert_edge(6,3)
64 print('Graph Adjacency Matrix')
65 G.display()
66 print('Number of Vertices: ', G.vertices_count())
67 print('Number of Edges: ', G.edge_count())
68 print('Outdegree of Vertex 2: ', G.outdegree(2))
69
70
71
```