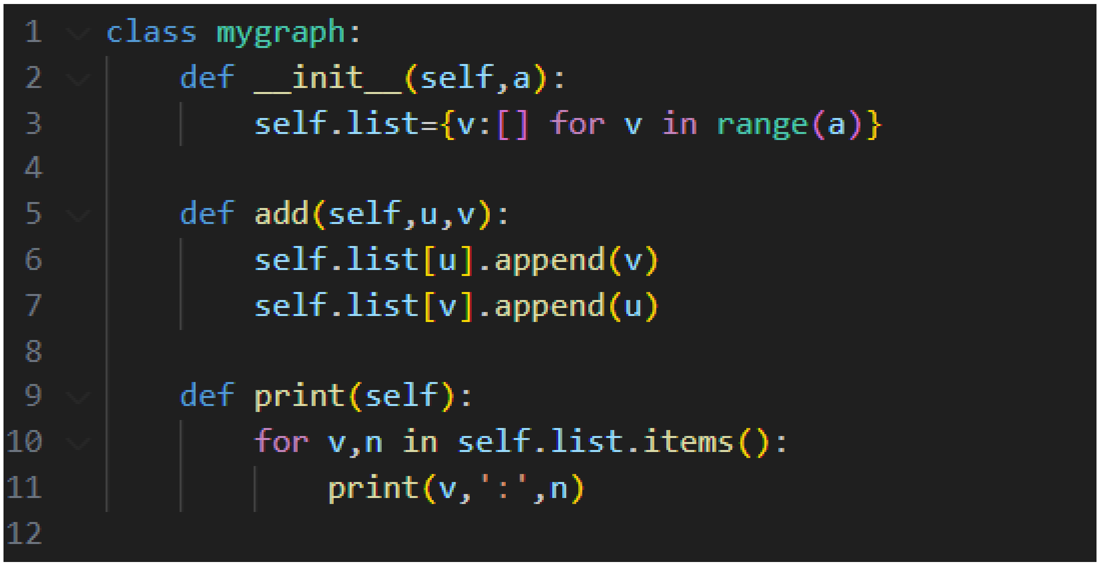
LAB#15

Example#1: Write a program to create an adjacency list for a graph. Also, create a function to add edges to the graph and another function to display the list.

Solution:

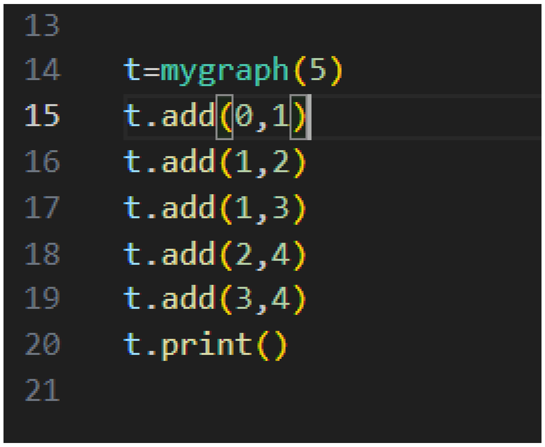


```python
class mygraph:
    def __init__(self,a):
        self.list={v:[] for v in range(a)}

    def add(self,u,v):
        self.list[u].append(v)
        self.list[v].append(u)

    def print(self):
        for v,n in self.list.items():
            print(v,':',n)

```

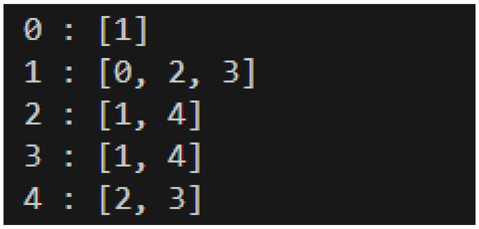


```python

t=mygraph(5)
t.add(0,1)
t.add(1,2)
t.add(1,3)
t.add(2,4)
t.add(3,4)
t.print()

```

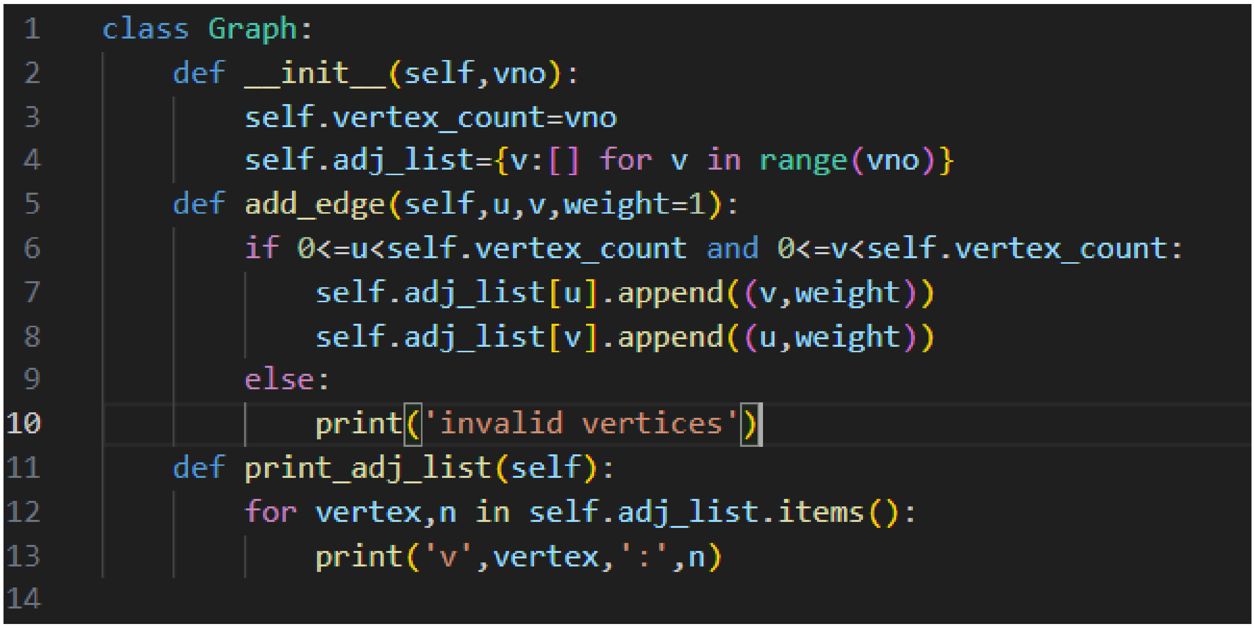
Output:



```
0 : [1]
1 : [0, 2, 3]
2 : [1, 4]
3 : [1, 4]
4 : [2, 3]
```

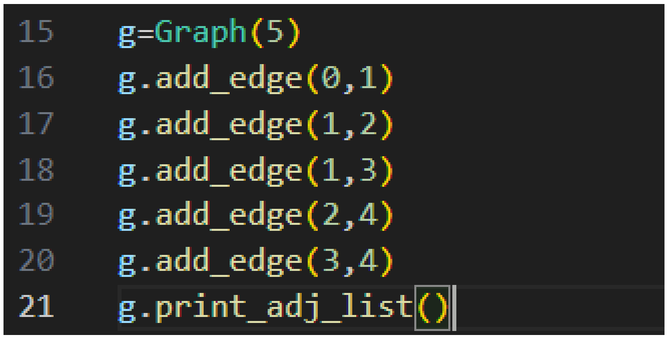
Example#2: Write a program to create an adjacency list for a graph. Also, create a function to add edges to the graph with their weights and another function to display the list.

Solution:



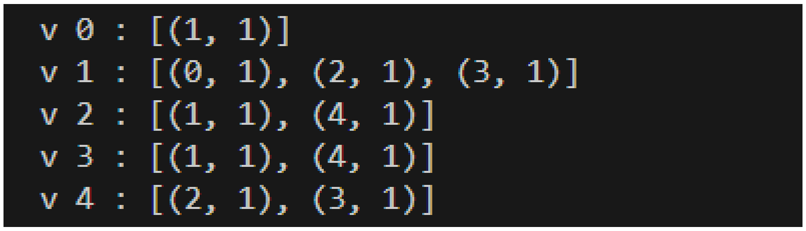
```python
class Graph:
    def __init__(self,vno):
        self.vertex_count=vno
        self.adj_list={v:[] for v in range(vno)}
    def add_edge(self,u,v,weight=1):
        if 0<=u<self.vertex_count and 0<=v<self.vertex_count:
            self.adj_list[u].append((v,weight))
            self.adj_list[v].append((u,weight))
        else:
            print('invalid vertices')
    def print_adj_list(self):
        for vertex,n in self.adj_list.items():
            print('v',vertex,':',n)

```



```python
g=Graph(5)
g.add_edge(0,1)
g.add_edge(1,2)
g.add_edge(1,3)
g.add_edge(2,4)
g.add_edge(3,4)
g.print_adj_list()
```

Result:



```
v 0 : [(1, 1)]
v 1 : [(0, 1), (2, 1), (3, 1)]
v 2 : [(1, 1), (4, 1)]
v 3 : [(1, 1), (4, 1)]
v 4 : [(2, 1), (3, 1)]
```

Class Assignment

Q: Modify the example#2 from lab#15 by adding a function remove_edge and add the weight by your own choice in the add_edge function.