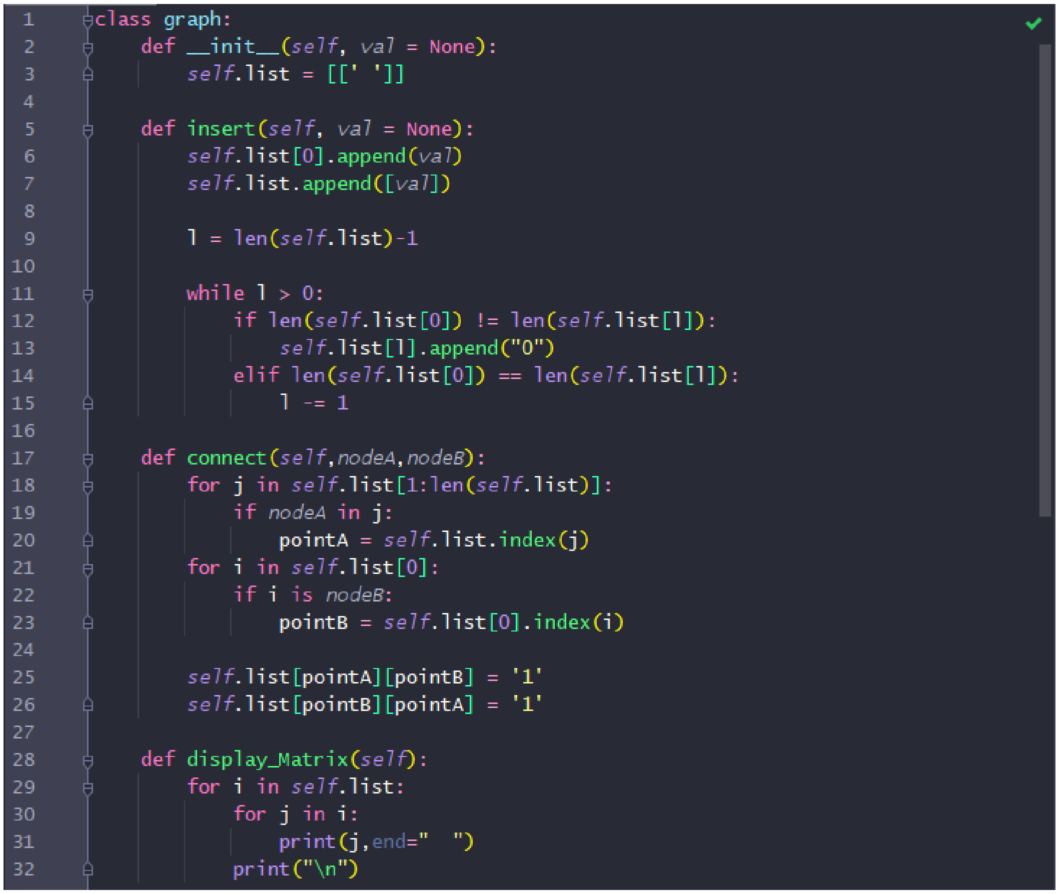
Basic graph structure

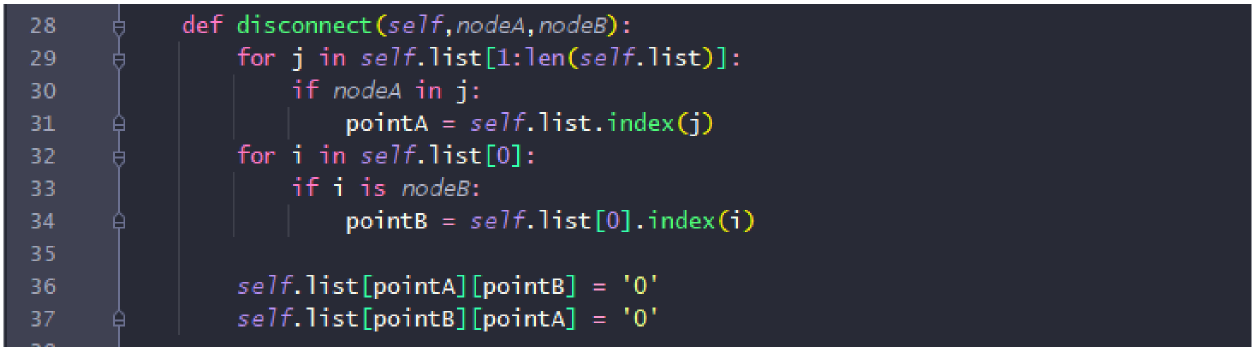
Code:

Create, Connect and display Adjacency matrix function



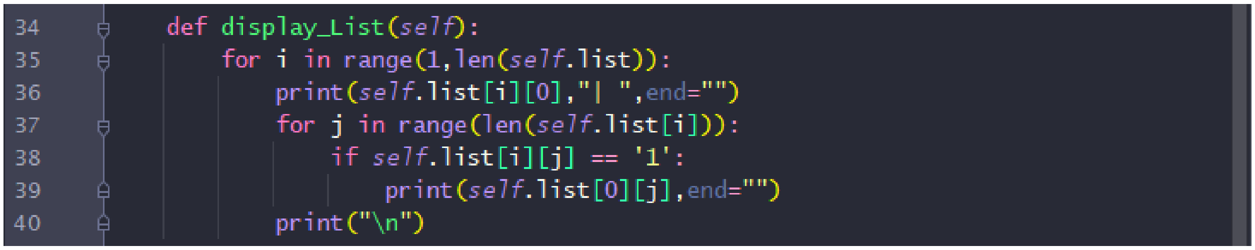
```python
class graph:
    def __init__(self, val = None):
        self.list = [[' ']]

    def insert(self, val = None):
        self.list[0].append(val)
        self.list.append([val])

        l = len(self.list)-1

        while l > 0:
            if len(self.list[0]) != len(self.list[l]):
                self.list[l].append("0")
            elif len(self.list[0]) == len(self.list[l]):
                l -= 1

    def connect(self,nodeA,nodeB):
        for j in self.list[1:len(self.list)]:
            if nodeA in j:
                pointA = self.list.index(j)
        for i in self.list[0]:
            if i is nodeB:
                pointB = self.list[0].index(i)

        self.list[pointA][pointB] = '1'
        self.list[pointB][pointA] = '1'

    def display_Matrix(self):
        for i in self.list:
            for j in i:
                print(j,end="  ")
            print("\n")
```

Disconnect function



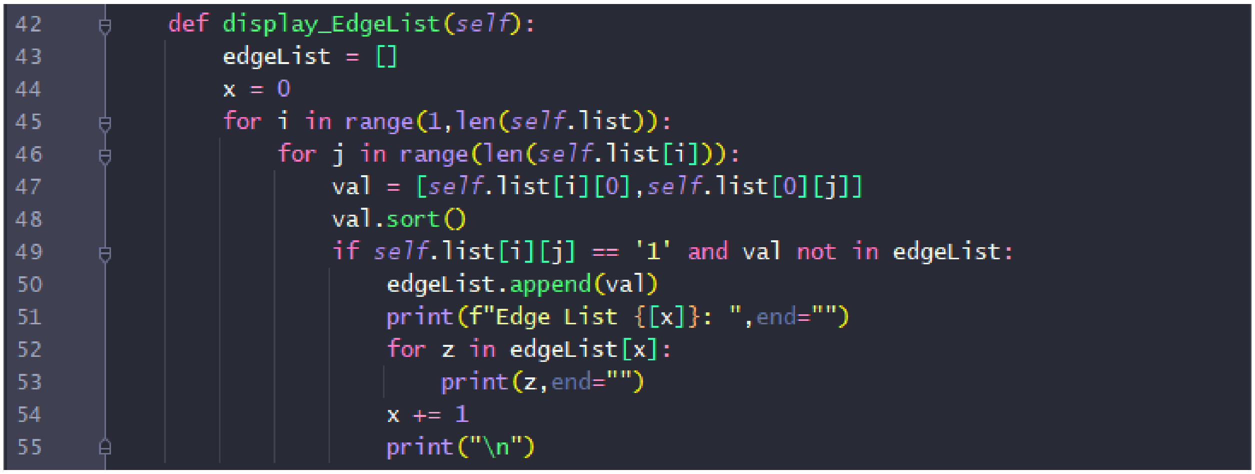
```python
    def disconnect(self,nodeA,nodeB):
        for j in self.list[1:len(self.list)]:
            if nodeA in j:
                pointA = self.list.index(j)
        for i in self.list[0]:
            if i is nodeB:
                pointB = self.list[0].index(i)

        self.list[pointA][pointB] = '0'
        self.list[pointB][pointA] = '0'
```

display Adjacency list function



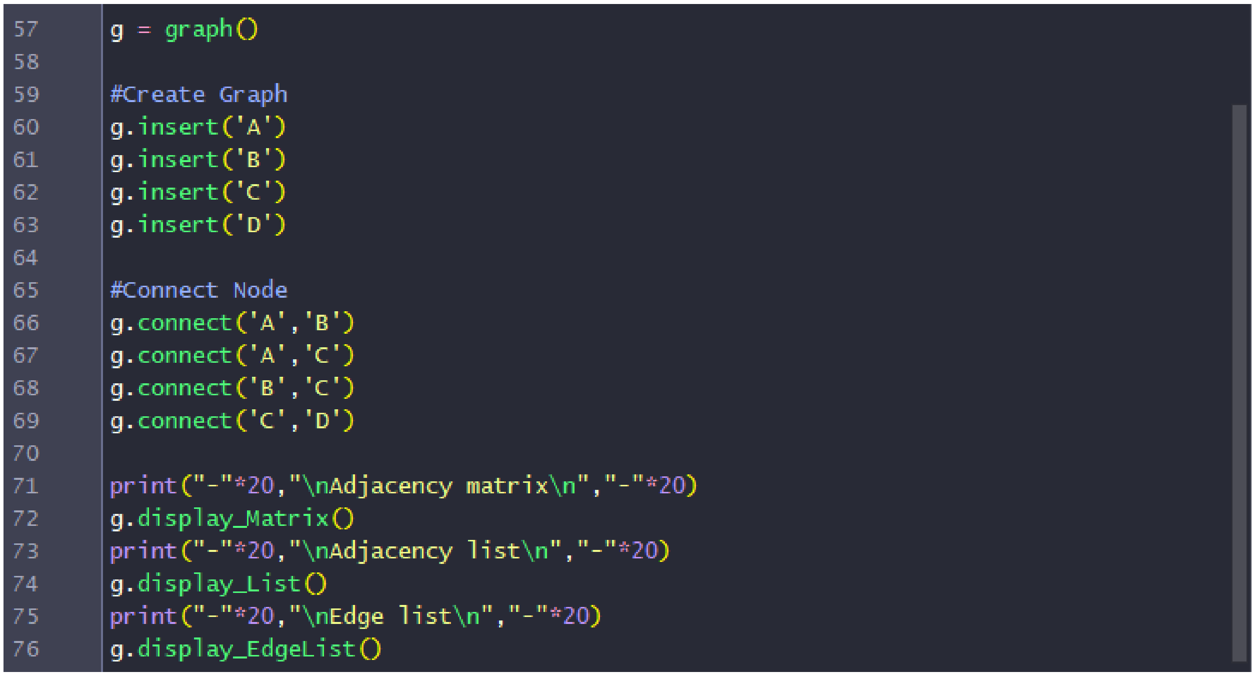
```python
    def display_List(self):
        for i in range(1,len(self.list)):
            print(self.list[i][0],"| ",end="")
            for j in range(len(self.list[i])):
                if self.list[i][j] == '1':
                    print(self.list[0][j],end="")
            print("\n")
```

display Edge list function



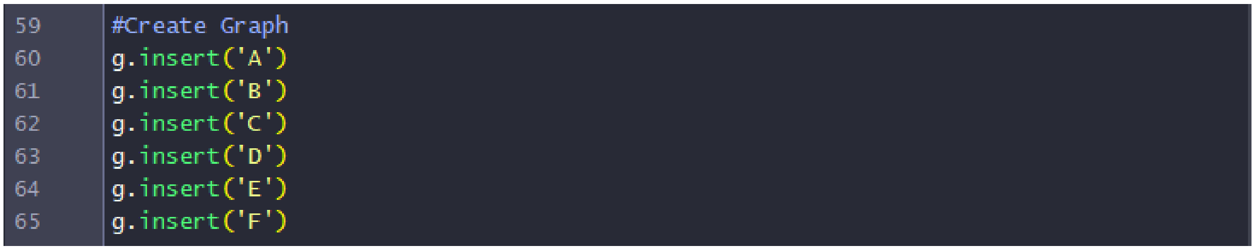
```python
    def display_EdgeList(self):
        edgeList = []
        x = 0
        for i in range(1,len(self.list)):
            for j in range(len(self.list[i])):
                val = [self.list[i][0],self.list[0][j]]
                val.sort()
                if self.list[i][j] == '1' and val not in edgeList:
                    edgeList.append(val)
                    print(f"Edge List {[x]}: ",end="")
                    for z in edgeList[x]:
                        print(z,end="")
                    x += 1
                    print("\n")
```

EX 1



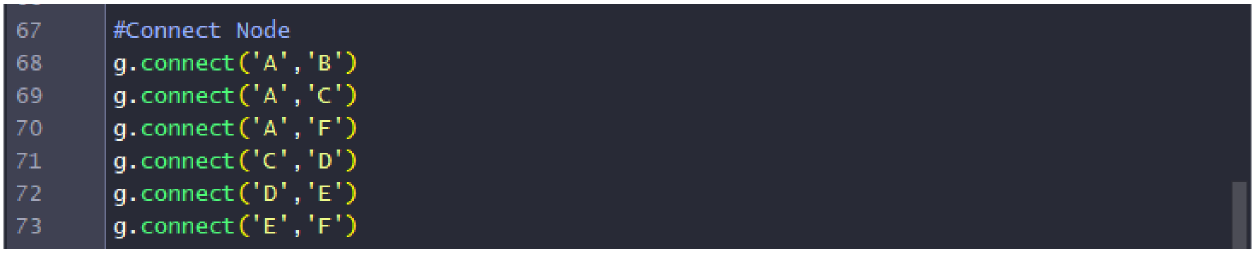
```python
g = graph()

#Create Graph
g.insert('A')
g.insert('B')
g.insert('C')
g.insert('D')

#Connect Node
g.connect('A','B')
g.connect('A','C')
g.connect('B','C')
g.connect('C','D')

print("-"*20,"\nAdjacency matrix\n","-"*20)
g.display_Matrix()
print("-"*20,"\nAdjacency list\n","-"*20)
g.display_List()
print("-"*20,"\nEdge list\n","-"*20)
g.display_EdgeList()
```

EX2 and EX3



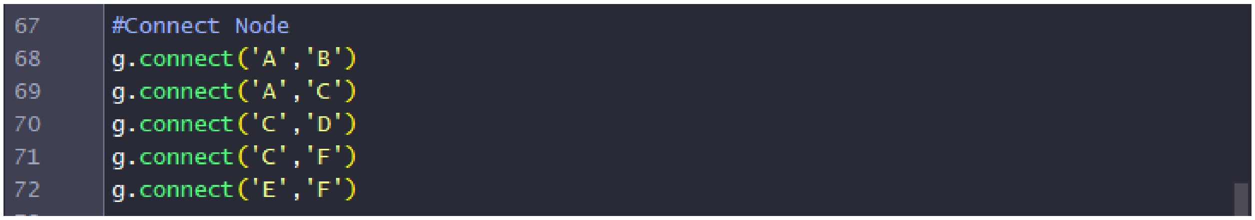
```python
#Create Graph
g.insert('A')
g.insert('B')
g.insert('C')
g.insert('D')
g.insert('E')
g.insert('F')
```

EX2



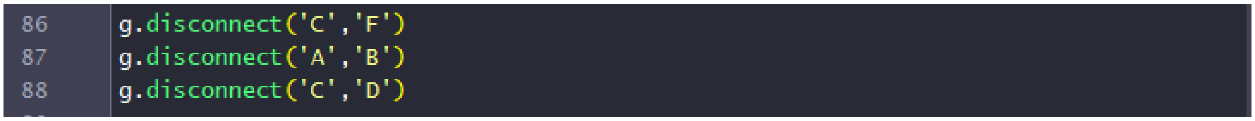
```python
#Connect Node
g.connect('A','B')
g.connect('A','C')
g.connect('A','F')
g.connect('C','D')
g.connect('D','E')
g.connect('E','F')
```

EX3-1



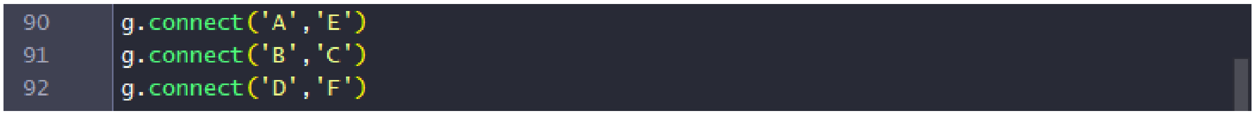
```python
#Connect Node
g.connect('A','B')
g.connect('A','C')
g.connect('C','D')
g.connect('C','F')
g.connect('E','F')
```

EX3-2



```python
g.disconnect('C','F')
g.disconnect('A','B')
g.disconnect('C','D')
```

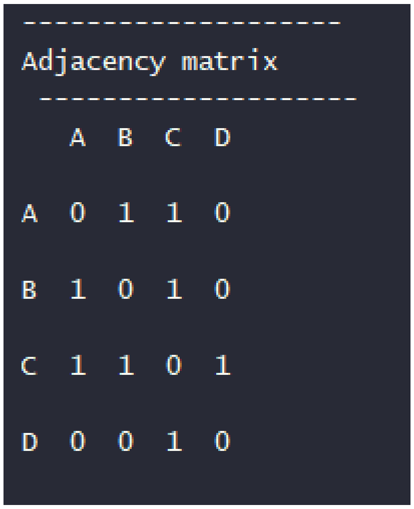
EX3-3



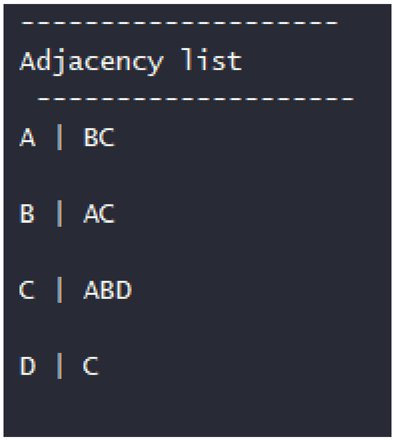
```python
g.connect('A','E')
g.connect('B','C')
g.connect('D','F')
```

Result :

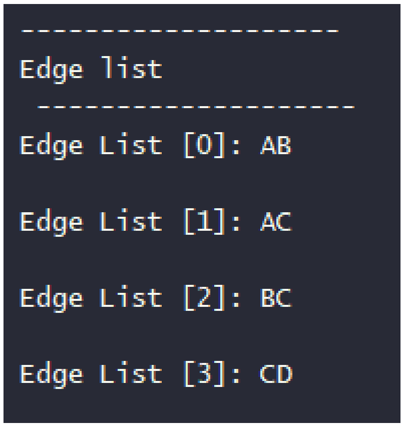
EX 1 : Adjacency matrix | EX 1 : Adjacency List | EX 1 : Edge list



```
 --------------------
Adjacency matrix
 --------------------
   A  B  C  D

A  0  1  1  0

B  1  0  1  0

C  1  1  0  1

D  0  0  1  0
```

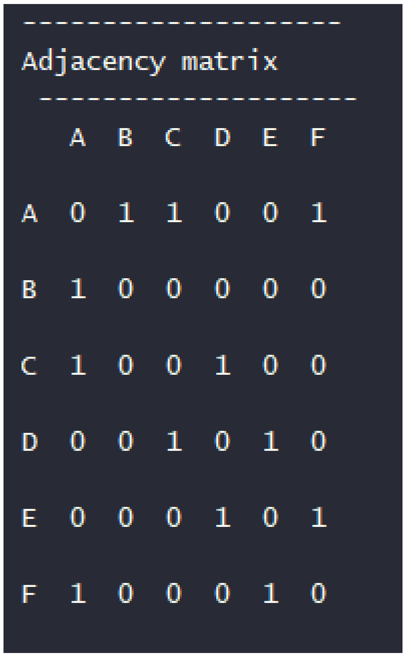


```
 --------------------
Adjacency list
 --------------------
A | BC

B | AC

C | ABD

D | C
```

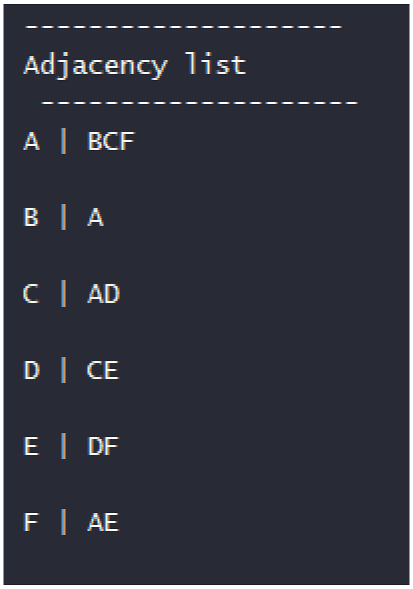


```
 --------------------
Edge list
 --------------------
Edge List [0]: AB

Edge List [1]: AC

Edge List [2]: BC

Edge List [3]: CD
```

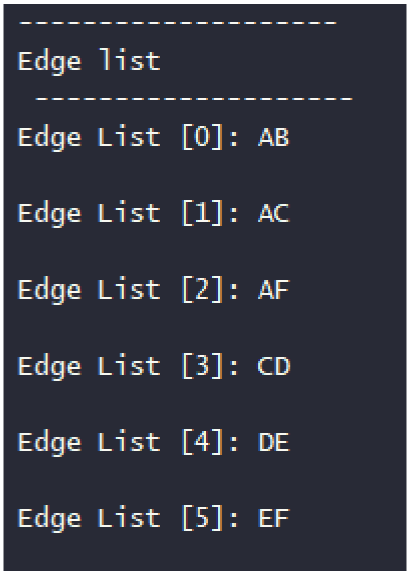
EX 2 : Adjacency matrix | EX 2 : Adjacency List | EX 2 : Edge list



```
 --------------------
Adjacency matrix
 --------------------
   A  B  C  D  E  F

A  0  1  1  0  0  1

B  1  0  0  0  0  0

C  1  0  0  1  0  0

D  0  0  1  0  1  0

E  0  0  0  1  0  1

F  1  0  0  0  1  0
```



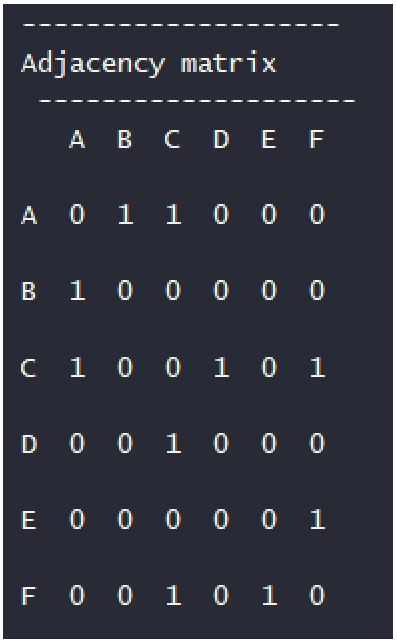
```
 --------------------
Adjacency list
 --------------------
A | BCF

B | A

C | AD

D | CE

E | DF

F | AE
```



```
 --------------------
Edge list
 --------------------
Edge List [0]: AB

Edge List [1]: AC

Edge List [2]: AF

Edge List [3]: CD

Edge List [4]: DE

Edge List [5]: EF
```

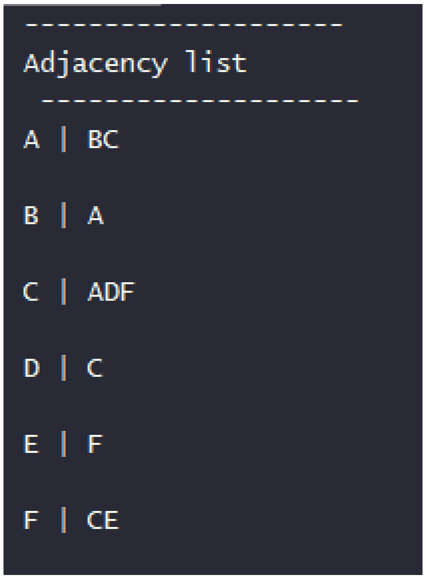
Connect AB,AC,CD,CF,EF

EX 3: Adjacency matrix



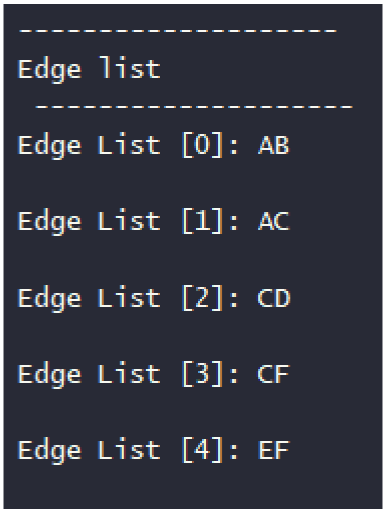
```
---------------------
Adjacency matrix
 ---------------------
   A  B  C  D  E  F

A  0  1  1  0  0  0

B  1  0  0  0  0  0

C  1  0  0  1  0  1

D  0  0  1  0  0  0

E  0  0  0  0  0  1

F  0  0  1  0  1  0
```

EX 3 : Adjacency List



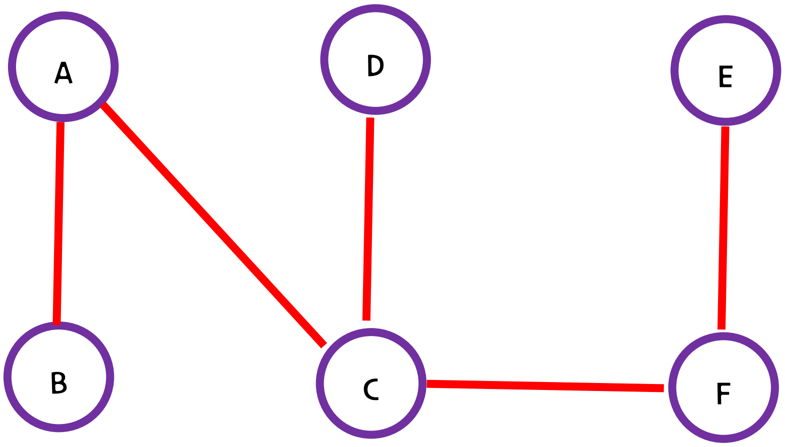
```
---------------------
Adjacency list
 ---------------------
A | BC

B | A

C | ADF

D | C

E | F

F | CE
```

EX 3 : Edge list



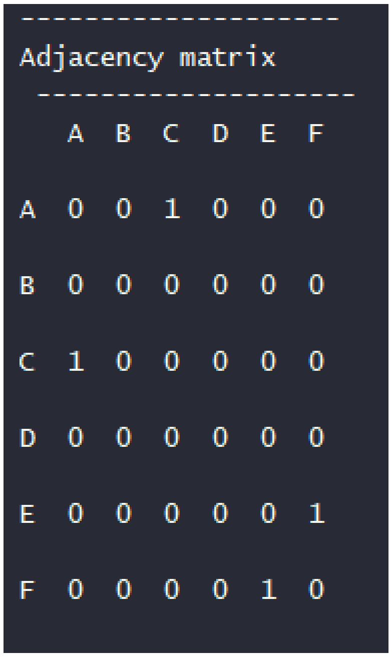
```
---------------------
Edge list
 ---------------------
Edge List [0]: AB

Edge List [1]: AC

Edge List [2]: CD

Edge List [3]: CF

Edge List [4]: EF
```

draw graph diagram

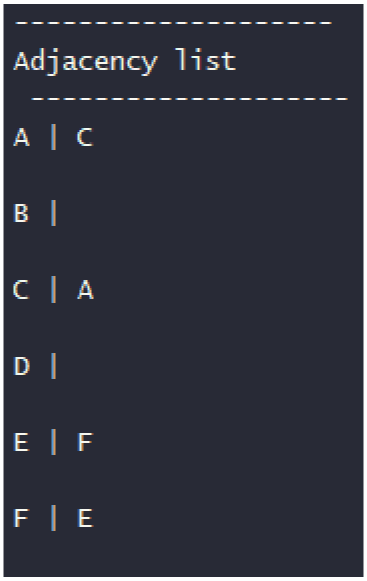


Disconnect CF,AB,CD

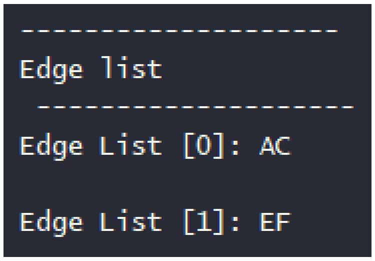
EX 3: Adjacency matrix | EX 3 : Adjacency List | EX 3 : Edge list



```
 --------------------
Adjacency matrix
  --------------------
   A  B  C  D  E  F

A  0  0  1  0  0  0

B  0  0  0  0  0  0

C  1  0  0  0  0  0

D  0  0  0  0  0  0

E  0  0  0  0  0  1

F  0  0  0  0  1  0
```

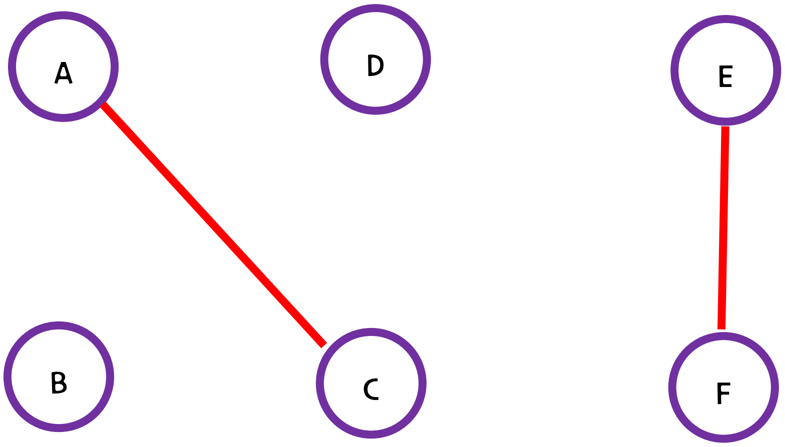


```
 --------------------
Adjacency list
  --------------------
A | C

B |

C | A

D |

E | F

F | E
```



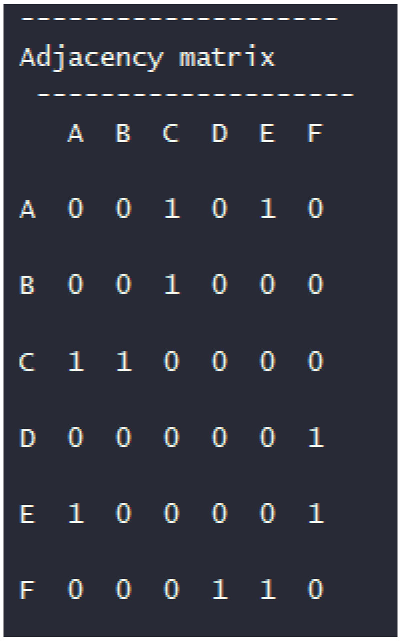
```
 --------------------
Edge list
  --------------------
Edge List [0]: AC

Edge List [1]: EF
```

draw graph diagram

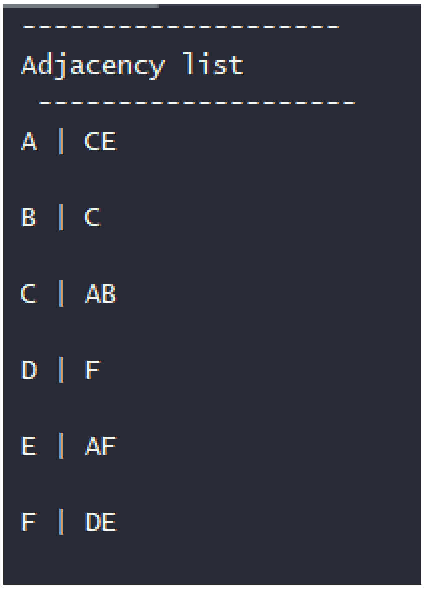


Connect AE, BC, DF

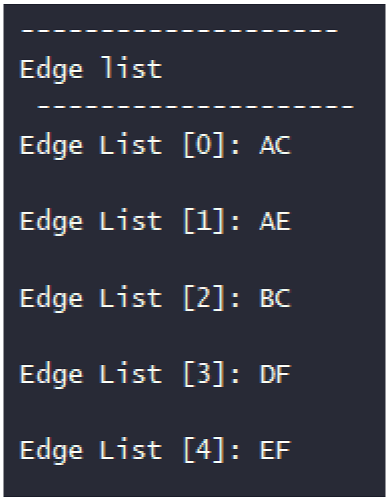
EX 3: Adjacency matrix | EX 3 : Adjacency List | EX 3 : Edge lis t



```
--------------------
Adjacency matrix
 --------------------
   A  B  C  D  E  F

A  0  0  1  0  1  0

B  0  0  1  0  0  0

C  1  1  0  0  0  0

D  0  0  0  0  0  1

E  1  0  0  0  0  1

F  0  0  0  1  1  0
```



```
--------------------
Adjacency list
 --------------------
A | CE

B | C

C | AB

D | F

E | AF

F | DE
```



```
--------------------
Edge list
 --------------------
Edge List [0]: AC

Edge List [1]: AE

Edge List [2]: BC

Edge List [3]: DF

Edge List [4]: EF
```

draw graph diagram