

Create a Dictionary of lists to store the information of shipments given in the table

In [2]:

```
ship= {101 : [1,3, '14-03-2020', '25-03-2020', 'Area1', 'Area6', 'Delivered', 198],
        102 : [4,1, '18-06-2020', '09-07-2020', 'Area2', 'Area4', 'Delivered', 275],
        103 : [2,3, '01-12-2020', 'Null', 'Area5', 'Areal', 'In-Transit', 200],
        104 : [1,5, '23-06-2020', '25-06-2020', 'Area1', 'Area4', 'Delivered', 314],
        105 : [3,4, '29-08-2020', '10-09-2020', 'Area5', 'Area3', 'Delivered', 275],
        106 : [5,2, '28-09-2020', 'Null', 'Area3', 'Area1', 'In-Transit', 270]}
```

Create a Dictionary of to store the information of clients given in the table.

In [3]:

```
clients = {1: 'Phillip', 2: 'Omega III', 3: 'Ramya', 4: 'Ramesh', 5: 'John'}
clients[5]
```

Out[3]:

'John'

Write a code to replace client's id with their respective name in shipment dictionary using a loop and dictionary comprehension.

In [4]:

```
for i in ship:
    a= ship[i][0]
    b= ship[i][1]

    ship[i][0]=clients[a]
    ship[i][1]=clients[b]

ship
```

Out[4]:

```
{101: ['Phillip',
       'Ramya',
       '14-03-2020',
       '25-03-2020',
       'Area1',
       'Area6',
       'Delivered',
       198],
 102: ['Ramesh',
       'Phillip',
       '18-06-2020',
       '09-07-2020',
       'Area2',
       'Area4',
       'Delivered',
       275],
 103: ['Omega III',
       'Ramya',
       '01-12-2020',
       'Null',
       'Area5',
       'Areal',
       'In-Transit',
       200],
 104: ['Phillip',
       'John',
       '23-06-2020',
       '25-06-2020',
       'Area1',
       'Area4',
       'Delivered',
       314],
 105: ['Ramya',
       'Ramesh',
       '29-08-2020',
       '10-09-2020',
       'Area5',
       'Area3',
       'Delivered',
       275],
 106: ['John',
       'Null',
       '28-09-2020',
       'Null',
       'Area3',
       'Area1',
       'In-Transit',
       270]}
```

```

        'Area4',
        'Delivered',
        314],
105: ['Ramya',
      'Romesh',
      '29-08-2020',
      '10-09-2020',
      'Area5',
      'Area3',
      'Delivered',
      275],
106: ['John',
      'Omega III',
      '28-09-2020',
      'Null',
      'Area3',
      'Areal',
      'In-Transit',
      270]}

```

Print all shipment details that are sent by Phillip.

In [5]:

```

for i in ship:
    if (ship[i][0] == 'Phillip'): print (ship[i])

```

```

['Phillip', 'Ramya', '14-03-2020', '25-03-2020', 'Area1', 'Area6', 'Delivered', 198]
['Phillip', 'John', '23-06-2020', '25-06-2020', 'Area1', 'Area4', 'Delivered', 314]

```

Print all shipment details that are received by Ramya.

In [6]:

```

for i in ship:
    if (ship[i][1]=='Ramya'): print (ship[i])

```

```

['Phillip', 'Ramya', '14-03-2020', '25-03-2020', 'Area1', 'Area6', 'Delivered', 198]
['Omega III', 'Ramya', '01-12-2020', 'Null', 'Area5', 'Areal', 'In-Transit', 200]

```

Print all shipments which are in 'In-Transit' status.

In [7]:

```

for i in ship:
    if (ship[i][6] == 'In-Transit'):
        print (ship[i])

```

```

['Omega III', 'Ramya', '01-12-2020', 'Null', 'Area5', 'Areal', 'In-Transit', 200]
['John', 'Omega III', '28-09-2020', 'Null', 'Area3', 'Areal', 'In-Transit', 270]

```

Print all shipments which are delivered within 7 days of courier Start date.

In [8]:

```

import datetime as dt
for i in ship:
    if (ship[i][3] != 'Null'):
        start = dt.datetime.strptime (ship[i][2], "%d-%m-%Y")
        end = dt.datetime.strptime (ship[i][3], "%d-%m-%Y")
        if (end-start) <= dt.timedelta(days=7):
            print (ship[i])

```

```

['Phillip', 'John', '23-06-2020', '25-06-2020', 'Area1', 'Area4', 'Delivered', 314]

```

Print all shipments which are delivered after 15 days of courier start date or not yet been delivered.

In [9]:

```
import datetime as dt
for i in ship:
    if (ship[i][3] == 'Null'):
        print (ship[i])
    else:
        start= dt.datetime.strptime(ship[i][2], "%d-%m-%Y")
        end= dt.datetime.strptime (ship[i][3], "%d-%m-%Y")
        if (end-start) > dt.timedelta(days = 15):
            print(ship[i])
```

```
['Romesh', 'Phillip', '18-06-2020', '09-07-2020', 'Area2', 'Area4', 'Delivered', 275]
['Omega III', 'Ramya', '01-12-2020', 'Null', 'Area5', 'Areal', 'In-Transit', 200]
['Omega III', 'Ramya', '01-12-2020', 'Null', 'Area5', 'Areal', 'In-Transit', 200]
['John', 'Omega III', '28-09-2020', 'Null', 'Area3', 'Areal', 'In-Transit', 270]
```

Write a function find_all_routes to display all possible routes from senders location to receivers location given in the dictionary for each shipment.

In [12]:

```
def find_all_routes(matrix,u,d,visited, path): # matrix, u=0 to 5, False, path
    visited[u]= True #visited[0]=True

    path.append(u+1) #u+1 if u == d:
    if(u==d):
        print (path)

    else:
        for i in range(6): #0 to 5 [0][0], [0][1], [0][2]
            if matrix[u][i] == 1 and (visited[i]== False):
                find_all_routes (matrix,i, d, visited, path)

        path.pop()
        visited[u]= False

matrix = [[0,1,0,0,0,1],[1,0,1,1,0,0], [0,1,0,1,0,0], [0,1,1,0,1,0], [0,0,0,1,0,0],[1,0,0,0,0,0]]
visited =[False for i in range(6)] #

path = []
```

In [13]:

```
matrix
```

Out[13]:

```
[[0, 1, 0, 0, 0, 1],
 [1, 0, 1, 1, 0, 0],
 [0, 1, 0, 1, 0, 0],
 [0, 1, 1, 0, 1, 0],
 [0, 0, 0, 1, 0, 0],
 [1, 0, 0, 0, 0, 0]]
```

In [14]:

```
visited
```

Out[14]:

```
[False, False, False, False, False, False]
```

In [16]:

```
def show_all(ship):

    for i in ship:

        print('Shipment Id: ',i)
```

```
s = int(ship[i][4][-1])
print("Sender's Location" , S)
r = int(ship[i][5][-1])
print("Receiver's Location", r)
print('All possible routes : ')
find_all_routes (matrix, s-1,r-1, visited, path)
print>Show_all(ship))
```

In []: