In [10]:

```
import numpy as np
import math
from numpy.linalg import inv
np.set_printoptions(threshold=np.inf)
area=4e-4
E=2.1e11
nodes=int(input("Enter the nodes: "))

GLOBAL=np.zeros((nodes*2,nodes*2))
add=np.zeros((nodes*2,nodes*2))
joints=np.zeros((nodes,3))
for i in range(nodes):
    print(i+1)
    joints[i][0]=i+1    #the joint number
    joints[i][1]=input("X:")
    joints[i][2]=input("Y:")

ELEMENT=int(input("Enter the ELEMENTS: "))
element=np.zeros((ELEMENT,9))
print("Enter the nodes one after the other corresponding to the element that is displa
for j in range(ELEMENT):
    print("\n")
    print("ELEMENT NUMBER:",j+1)
    element[j][0]=j+1
    element[j][1]=input("Enter the NODE I:")
    element[j][2]=input("Enter the NODE II:")
    #element[j][3]=input("Enter the angle   :")
    #element[j][3]=math.radians(element[j][3])
    try:
        element[j][3]=np.arctan((joints[int(element[j][2]-1)][2]-joints[int(element[j]
    except:
        element[j][3]=np.tan(90)
    element[j][4]=joints[int(element[j][1]-1)][1]
    element[j][5]=joints[int(element[j][1]-1)][2]
    element[j][6]=joints[int(element[j][2]-1)][1]
    element[j][7]=joints[int(element[j][2]-1)][2]
    element[j][8]=((element[j][4]-element[j][6])**2+(element[j][5]-element[j][7])**2)*


small=np.zeros((2,2))

#print(element)

for i in range(ELEMENT):
    print("Element:",element[i][0],"node connection:",element[i][1],element[i][2],"Ang
    small[0][0]=round((np.cos(element[i][3])**2)*area*E/element[i][8],3)
    small[0][1]=round(np.cos(element[i][3])*np.sin(element[i][3])*area*E/element[i][8]
    small[1][0]=round(np.cos(element[i][3])*np.sin(element[i][3])*area*E/element[i][8]
    small[1][1]=round((np.sin(element[i][3])**2)*area*E/element[i][8],3)
    #print(small)
    #print(area*E/element[i][8])
    for x in range(nodes*2):
        for y in range(nodes*2):
            if(x==2*(element[i][1]-1) and y==2*(element[i][1]-1)):
                add[x][y]=small[0][0]
                add[x][y+1]=small[0][1]
                add[x+1][y]=small[1][0]
                add[x+1][y+1]=small[1][1]
            if(x==2*(element[i][2]-1) and y==2*(element[i][2]-1)):
```

```
                add[x][y]=small[0][0]
                add[x][y+1]=small[0][1]
                add[x+1][y]=small[1][0]
                add[x+1][y+1]=small[1][1]
            if(x==2*(element[i][1]-1) and y==2*(element[i][2]-1)):
                add[x][y]=-small[0][0]
                add[x][y+1]=-small[0][1]
                add[x+1][y]=-small[1][0]
                add[x+1][y+1]=-small[1][1]
            if(x==2*(element[i][2]-1) and y==2*(element[i][1]-1)):
                add[x][y]=-small[0][0]
                add[x][y+1]=-small[0][1]
                add[x+1][y]=-small[1][0]
                add[x+1][y+1]=-small[1][1]
    #print(add)
    GLOBAL=GLOBAL+add
    add.fill(0)
    small.fill(0)
#print(GLOBAL)
Forces=np.zeros((nodes*2,1))
vi=int(input("Enter the Total number of nodes where Point loads are applied:"))
ForceNodenumbers=np.zeros((vi))
for i in range(vi):
    ForceNodenumbers[i]=int(input("Enter the Node numbers where Load is applied one by
for i in range(vi):
    print("\n")
    print("Force Input at Node: ",ForceNodenumbers[i])
    Forces[int(2*(ForceNodenumbers[i]-1))][0]=float(input("enter the X force in the No
    Forces[int((2*(ForceNodenumbers[i]-1))+1)][0]=float(input("enter the Y force in th

GLOBAL=np.delete(GLOBAL, (7),axis=0)
GLOBAL=np.delete(GLOBAL, (6),axis=0)

GLOBAL=np.delete(GLOBAL, (5),axis=0)
GLOBAL=np.delete(GLOBAL, (4),axis=0)

GLOBAL=np.delete(GLOBAL, (3),axis=0)
GLOBAL=np.delete(GLOBAL, (2),axis=0)


GLOBAL=np.delete(GLOBAL, (7),axis=1)
GLOBAL=np.delete(GLOBAL, (6),axis=1)

GLOBAL=np.delete(GLOBAL, (5),axis=1)
GLOBAL=np.delete(GLOBAL, (4),axis=1)

GLOBAL=np.delete(GLOBAL, (3),axis=1)
GLOBAL=np.delete(GLOBAL, (2),axis=1)

Forces=np.delete(Forces, (7),axis=0)
Forces=np.delete(Forces, (6),axis=0)
Forces=np.delete(Forces, (5),axis=0)
Forces=np.delete(Forces, (4),axis=0)
Forces=np.delete(Forces, (3),axis=0)
Forces=np.delete(Forces, (2),axis=0)
print(GLOBAL)
print(Forces)
print(np.dot(inv(GLOBAL),Forces))
```

```
Enter the nodes: 4
```

```
1
X:0
Y:0
2
X:0
Y:3
3
X:2.121
Y:2.121
4
X:3
Y:0
Enter the ELEMENTS: 3
Enter the nodes one after the other corresponding to the element that is d
isplayed


ELEMENT NUMBER: 1
Enter the NODE I:1
Enter the NODE II:2
```

```
c:\users\vinay\appdata\local\programs\python\python37-32\lib\site-packages\i
pykernel_launcher.py:30: RuntimeWarning: divide by zero encountered in doubl
e_scalars
```

```


ELEMENT NUMBER: 2
Enter the NODE I:1
Enter the NODE II:3


ELEMENT NUMBER: 3
Enter the NODE I:1
Enter the NODE II:4
Element: 1.0 node connection: 1.0 2.0 Angle: 90.0 L: 3.0
Element: 2.0 node connection: 1.0 3.0 Angle: 45.0 L: 2.9995469657933347
Element: 3.0 node connection: 1.0 4.0 Angle: 0.0 L: 3.0
Enter the Total number of nodes where Point loads are applied:1
Enter the Node numbers where Load is applied one by one:1


Force Input at Node:  1.0
enter the X force in the Node:-10000
enter the Y force in the Node:-20000
[[42002114.479 14002114.479]
 [14002114.479 42002114.479]]
[[-10000.]
 [-20000.]]
[[-8.92654880e-05]
 [-4.46408345e-04]]
```